ExtraHop 9.9
ExtraHop REST-API-Leitfaden

Weitere Informationen finden Sie unter https://docs.extrahop.com.

Veröffentlicht: 2025-01-04

ExtraHop Networks
Seattle, WA 98101
877-333-9872 (US)
+44 (0)203 7016850 (EMEA)
+65-31585513 (APAC)
www.extrahop.com

Inhaltsübersicht

Einführung in die ExtraHop REST API

Die ExtraHop REST API ermöglicht es Ihnen, Administrations- und Konfigurationsaufgaben auf Ihrem ExtraHop-System zu automatisieren. Sie können Anfragen über eine REST-Schnittstelle (Representational State Transfer) an die ExtraHop-API senden, auf die über Ressourcen-URIs und Standard zugegriffen wird HTTP Methoden.

Wenn eine REST-API-Anfrage über HTTPS an ein ExtraHop-System gesendet wird, wird diese Anfrage authentifiziert und dann über einen API-Schlüssel autorisiert. Nach der Authentifizierung wird die Anfrage an das ExtraHop-System gesendet und der Vorgang abgeschlossen.

Video: Sehen Sie sich die entsprechende Schulung an: Überblick über die Rest-API

Jedes ExtraHop-System bietet Zugriff auf den integrierten ExtraHop REST API Explorer, mit dem Sie alle verfügbaren Systemressourcen, Methoden, Eigenschaften und Parameter anzeigen können. Der REST API Explorer ermöglicht es Ihnen auch, API-Aufrufe direkt an Ihr ExtraHop-System zu senden.

Hinweis: Dieses Handbuch richtet sich an ein Publikum, das über grundlegende Kenntnisse in der Softwareentwicklung und dem ExtraHop-System verfügt.

ExtraHop API-Anforderungen

Bevor Sie mit dem Schreiben von Skripten für die ExtraHop REST API oder dem Ausführen von Vorgängen über den REST API Explorer beginnen können, müssen Sie die folgenden Anforderungen erfüllen:

- Ihr ExtraHop-System muss konfiguriert, um die Generierung von API-Schlüsseln zu ermöglichen für den Benutzertyp, der Sie sind (remote oder lokal).
- Du musst Generieren Sie einen gültigen API-Schlüssel.
- Sie benötigen ein Benutzerkonto auf dem ExtraHop-System mit entsprechendem Privilegien für die Art der Aufgaben festlegen, die Sie ausführen möchten.

Greifen Sie auf die ExtraHop REST API zu und authentifizieren Sie sich bei ihr

Setup-Benutzer und Benutzer mit System- und Zugriffsadministrationsrechten steuern, ob Benutzer API-Schlüssel generieren können. Sie können beispielsweise verhindern, dass Remotebenutzer Schlüssel generieren, oder Sie können die API-Schlüsselgenerierung vollständig deaktivieren. Wenn diese Funktion aktiviert ist, werden API-Schlüssel von Benutzern generiert und können nur von dem Benutzer eingesehen werden, der den Schlüssel generiert hat.

Hinweis: Administratoren richten Benutzerkonten ein und weisen Berechtigungen zu, aber dann generieren Benutzer ihre eigenen API-Schlüssel. Benutzer können API-Schlüssel für ihr eigenes Konto löschen, und Benutzer mit System- und Zugriffsadministrationsrechten können API-Schlüssel für jeden Benutzer löschen. Weitere Informationen finden Sie unter Benutzer und Benutzergruppen.

Nachdem Sie einen API-Schlüssel generiert haben, müssen Sie den Schlüssel an Ihre Anforderungsheader anhängen. Das folgende Beispiel zeigt eine Anfrage, die Metadaten über die Firmware abruft, die auf dem ExtraHop-System läuft:

```
curl -i -X GET --header "Accept: application/json" \
--header "Authorization: ExtraHop apikey=2bc07e55971d4c9a88d0bb4d29ecbb29" \
"https://<hostname-or-IP-of-your-ExtraHop-system>/api/v1/extrahop"
```

Privilegienstufen

Die Benutzerberechtigungsstufen bestimmen, welche ExtraHop-System- und Verwaltungsaufgaben der Benutzer über die ExtraHop-REST-API ausführen kann.

Sie können die Berechtigungsstufen für Benutzer über das `granted_roles` und `effective_roles` Eigenschaften. Das `granted_roles` Diese Eigenschaft zeigt Ihnen, welche Rechtestufen dem Benutzer explizit gewährt werden. Das `effective_roles` Diese Eigenschaft zeigt Ihnen alle Berechtigungsstufen für einen Benutzer an, einschließlich derer, die Sie außerhalb der erteilten Rolle erhalten haben, z. B. über eine Benutzergruppe.

Das `granted_roles` und `effective_roles` Eigenschaften werden durch die folgenden Operationen zurückgegeben:

- GET /users
- GET /users/ {username}

Das `granted_roles` und `effective_roles` Eigenschaften unterstützen die folgenden Berechtigungsstufen. Beachten Sie, dass die Art der Aufgaben für jedes ExtraHop-System je nach Verfügbarkeit variiert Ressourcen sind im REST API Explorer aufgeführt und hängen von den Modulen ab, die für die System- und Benutzermodulzugriffsrechte aktiviert sind.

Privilegienstufe	Zulässige Aktionen
„system“: „voll“	• Aktiviert oder deaktiviert die API-Schlüsselgenerierung für das ExtraHop-System. • Generieren Sie einen API-Schlüssel. • Sehen Sie sich die letzten vier Ziffern und die Beschreibung für jeden API-Schlüssel auf dem System an. • Löschen Sie API-Schlüssel für jeden Benutzer. • CORS anzeigen und bearbeiten.

Privilegienstufe	Zulässige Aktionen
	• Führen Sie alle Verwaltungsaufgaben aus, die über die REST-API verfügbar sind. • Führen Sie alle ExtraHop-Systemaufgaben aus, die über die REST-API verfügbar sind.
„write“: „voll“	• Generieren Sie Ihren eigenen API-Schlüssel. • Zeigen Sie Ihren eigenen API-Schlüssel an oder löschen Sie ihn. • Ändern Sie Ihr eigenes Passwort, aber Sie können keine anderen Verwaltungsaufgaben über die REST-API ausführen. • Führen Sie alle ExtraHop-Systemaufgaben aus, die über die REST-API verfügbar sind.
„write“: „begrenzt“	• Generieren Sie einen API-Schlüssel. • Zeigen Sie ihren eigenen API-Schlüssel an oder löschen Sie ihn. • Ändern Sie Ihr eigenes Passwort, aber Sie können keine anderen Verwaltungsaufgaben über die REST-API ausführen. • Führen Sie alle GET-Operationen über die REST-API aus. • Führen Sie Metrik- und Datensatzabfragen durch.
„write“: „persönlich“	• Generieren Sie einen API-Schlüssel. • Zeigen Sie Ihren eigenen API-Schlüssel an oder löschen Sie ihn. • Ändern Sie Ihr eigenes Passwort, aber Sie können keine anderen Verwaltungsaufgaben über die REST-API ausführen. • Führen Sie alle GET-Operationen über die REST-API aus. • Führen Sie Metrik- und Datensatzabfragen durch.
„Metriken“: „voll“	• Generieren Sie einen API-Schlüssel. • Zeigen Sie Ihren eigenen API-Schlüssel an oder löschen Sie ihn. • Ändern Sie Ihr eigenes Passwort, aber Sie können keine anderen Verwaltungsaufgaben über die REST-API ausführen. • Führen Sie Metrik- und Datensatzabfragen durch.
„metrics“: „eingeschränkt“	• Generieren Sie einen API-Schlüssel. • Zeigen Sie Ihren eigenen API-Schlüssel an oder löschen Sie ihn. • Ändern Sie Ihr eigenes Passwort, aber Sie können keine anderen Verwaltungsaufgaben über die REST-API ausführen.
„ndr“: „voll“	• Sicherheitserkennungen anzeigen • Untersuchungen anzeigen und erstellen Dies ist ein Modulzugriffsrecht, das einem Benutzer zusätzlich zu einer der folgenden Systemzugriffsberechtigungsstufen gewährt werden kann: • „write“: „voll“ • „write“: „begrenzt“ • „write“: „persönlich“ • „schreiben“: null • „Metriken“: „voll“ • „metrics“: „eingeschränkt“
„ndr“: „keiner“	• Kein Zugriff auf NDR-Modulinhalte

Privilegienstufe	Zulässige Aktionen
	Dies ist ein Modulzugriffsrecht, das einem Benutzer zusätzlich zu einer der folgenden Systemzugriffsberechtigungsstufen gewährt werden kann: • „write“: „voll“ • „write“: „begrenzt“ • „write“: „persönlich“ • „schreiben“: null • „Metriken“: „voll“ • „metrics“: „eingeschränkt“
„npm“: „voll“	• Leistungserkennungen anzeigen • Dashboards anzeigen und erstellen • Benachrichtigungen anzeigen und erstellen Dies ist ein Modulzugriffsrecht, das einem Benutzer zusätzlich zu einer der folgenden Systemzugriffsberechtigungsstufen gewährt werden kann: • „write“: „voll“ • „write“: „begrenzt“ • „write“: „persönlich“ • „schreiben“: null • „Metriken“: „voll“ • „metrics“: „eingeschränkt“
„npm“: „keine“	• Kein Zugriff auf NPM-Modulinhalte Dies ist ein Modulzugriffsrecht, das einem Benutzer zusätzlich zu einer der folgenden Systemzugriffsberechtigungsstufen gewährt werden kann: • „write“: „voll“ • „write“: „begrenzt“ • „write“: „persönlich“ • „schreiben“: null • „Metriken“: „voll“ • „metrics“: „eingeschränkt“
„Pakete“: „voll“	• Pakete anzeigen und herunterladen über das `GET /packets/search` und `POST /packets/search` Operationen. Dies ist eine Zusatzberechtigung, die einem Benutzer mit einer der folgenden Berechtigungsstufen gewährt werden kann: • „write“: „voll“ • „write“: „begrenzt“ • „write“: „persönlich“ • „schreiben“: null • „Metriken“: „voll“ • „metrics“: „eingeschränkt“
„Pakete“: „voll_mit_Schlüsseln“	• Pakete und Sitzungsschlüssel anzeigen und herunterladen über das `GET /packets/search` und `POST /packets/search` Operationen.

Privilegienstufe	Zulässige Aktionen
	Dies ist eine Zusatzberechtigung, die einem Benutzer mit einer der folgenden Berechtigungsstufen gewährt werden kann: • „write“: „voll“ • „write“: „begrenzt“ • „write“: „persönlich“ • „schreiben“: null • „Metriken“: „voll“ • „metrics“: „eingeschränkt“
„Pakete“: „slices_only“	• Sehen Sie sich die ersten 64 Byte an Paketen an und laden Sie sie herunter über die `GET /packets/search` und `POST /packets/search` Operationen. Dies ist eine Zusatzberechtigung, die einem Benutzer mit einer der folgenden Berechtigungsstufen gewährt werden kann: • „write“: „voll“ • „write“: „begrenzt“ • „write“: „persönlich“ • „schreiben“: null • „Metriken“: „voll“ • „metrics“: „eingeschränkt“

API-Schlüsselzugriff verwalten

Benutzer mit System- und Zugriffsadministrationsrechten können konfigurieren, ob Benutzer API-Schlüssel für das ExtraHop-System generieren können. Sie können nur lokalen Benutzern erlauben, Schlüssel zu generieren, oder Sie können die API-Schlüsselgenerierung auch vollständig deaktivieren.

Benutzer müssen einen API-Schlüssel generieren, bevor sie Operationen über die ExtraHop REST API ausführen können. Schlüssel können nur von dem Benutzer, der den Schlüssel generiert hat, oder von Systemadministratoren mit unbegrenzten Rechten eingesehen werden. Nachdem ein Benutzer einen API-Schlüssel generiert hat, muss er den Schlüssel an seine Anforderungsheader anhängen.

1. Loggen Sie sich in die Administrationseinstellungen des ExtraHop-Systems ein über `https://<extrahop-hostname-or-IP-address>/admin`.
2. In der Auf Einstellungen zugreifen Abschnitt, klicken **API-Zugriff**.
3. In der API-Zugriff verwalten Abschnitt, wählen Sie eine der folgenden Optionen aus:
   - **Allen Benutzern erlauben, einen API-Schlüssel zu generieren**: Lokale und entfernte Benutzer können API-Schlüssel generieren.
   - **Nur lokale Benutzer können einen API-Schlüssel generieren**: Remote-Benutzer können keine API-Schlüssel generieren.
   - **Kein Benutzer kann einen API-Schlüssel generieren**: Es können keine API-Schlüssel von jedem Benutzer generiert werden.
4. klicken **Einstellungen speichern**.

Generieren Sie einen API-Schlüssel

Sie müssen einen API-Schlüssel generieren, bevor Sie Operationen über die ExtraHop REST-API ausführen können. Schlüssel können nur von dem Benutzer eingesehen werden, der den Schlüssel generiert hat, oder

von Benutzern mit System - und Zugriffsadministrationsrechten. Nachdem Sie einen API-Schlüssel generiert haben, fügen Sie den Schlüssel zu Ihren Anforderungsheadern oder dem ExtraHop REST API Explorer hinzu.

Bevor Sie beginnen
Stellen Sie sicher, dass das ExtraHop-System konfiguriert, um die Generierung von API-Schlüsseln zu ermöglichen.

1. In der Auf Einstellungen zugreifen Abschnitt, klicken Sie **API-Zugriff**.
2. In der Generieren Sie einen API-Schlüssel Abschnitt, geben Sie eine Beschreibung für den neuen Schlüssel ein, und klicken Sie dann auf **Generieren**.
3. Scrollen Sie nach unten zum API-Schlüssel Abschnitt und kopieren Sie den API-Schlüssel , der Ihrer Beschreibung entspricht.

Sie können den Schlüssel in den REST API Explorer einfügen oder den Schlüssel an einen Anforderungsheader anhängen.

Cross-Origin Resource Sharing (CORS) konfigurieren

Quellübergreifende gemeinsame Nutzung von Ressourcen (CORS) ermöglicht Ihnen den Zugriff auf die ExtraHop REST-API über Domänengrenzen und von bestimmten Webseiten aus, ohne dass die Anfrage über einen Proxyserver übertragen werden muss.

Sie können eine oder mehrere zulässige Ursprünge konfigurieren oder den Zugriff auf die ExtraHop REST-API von jedem beliebigen Ursprung aus zulassen. Nur Benutzer mit System- und Zugriffsadministrationsrechten können CORS-Einstellungen anzeigen und bearbeiten.

1. Loggen Sie sich in die Administrationseinstellungen des ExtraHop-Systems ein über `https://<extrahop-hostname-or-IP-address>/admin`.
2. In der Auf Einstellungen zugreifen Abschnitt, klicken **API-Zugriff**.
3. In der CORS-Einstellungen Abschnitt, geben Sie eine der folgenden Zugriffskonfigurationen an.
   - Um eine bestimmte URL hinzuzufügen, geben Sie eine Quell-URL in das Textfeld ein und klicken Sie dann auf das Pluszeichen (+) oder drücken Sie die EINGABETASTE.

     Die URL muss ein Schema enthalten, z. B. `HTTP` oder `HTTPS`, und der genaue Domänenname. Sie können keinen Pfad anhängen, Sie können jedoch eine Portnummer angeben.
   - Um den Zugriff von einer beliebigen URL aus zu ermöglichen, wählen Sie die **Erlaube API-Anfragen von jedem Ursprung** Ankreuzfeld.

     Hinweis: Das Zulassen des REST-API-Zugriffs von einem beliebigen Ursprung aus ist weniger sicher als das Bereitstellen einer Liste expliziter Ursprünge.
4. Klicken Sie **Einstellungen speichern** und klicken Sie dann **Erledigt**.

Richten Sie ein TLS-Zertifikat ein

Bevor Sie Anfragen an ein ExtraHop-System mit einem selbstsignierten Zertifikat stellen, müssen Sie für jeden Benutzer, der von einem bestimmten Computer aus auf das ExtraHop-System zugreift, ein TLS-Zertifikat einrichten.

Ersetzen Sie in jedem der folgenden Beispiele {HOST} durch den Hostnamen Ihres ExtraHop-Systems .

Hinweis: Das TLS-Zertifikat gilt nur für den Benutzer, der den Befehl ausführt. Jeder Benutzer muss den Befehl mit seinen Anmeldeinformationen ausführen, um das TLS-Zertifikat einzurichten.

Richten Sie TLS über Windows PowerShell ein

```
Invoke-WebRequest "http://{HOST}/public.cer" -OutFile ($env:USERPROFILE +
"\ex.cer"); Import-Certificate ($env:USERPROFILE + "\ex.cer")
```

```
-CertStoreLocation Cert:\CurrentUser\Root
```

TLS über OS X einrichten

```
curl -O http://{HOST}/public.cer; security add-trusted-cert -r trustRoot -k
~/Library/Keychains/login.keychain public.cer
```

Erfahren Sie mehr über den REST API Explorer

Der REST API Explorer ist ein webbasiertes Tool, mit dem Sie detaillierte Informationen zu den ExtraHop REST API-Ressourcen, Methoden, Parametern, Eigenschaften und Fehlercodes anzeigen können. Codebeispiele sind in Python, cURL und Ruby für jede Ressource verfügbar. Sie können Operationen auch direkt über das Tool ausführen.

Öffnen Sie den REST API Explorer

Sie können den REST API Explorer in den Administrationseinstellungen oder über die folgende URL öffnen:

```
https://<extrahop-hostname-or-ip-address>/api/v1/explore/
```

1. Loggen Sie sich in die Administrationseinstellungen des ExtraHop-Systems ein über `https://<extrahop-hostname-or-IP-address>/admin`.
2. Klicken Sie im Bereich Zugriffseinstellungen auf **API-Zugriff**.
3. Auf dem API-Zugriff Seite, klick **REST-API-Explorer**.

   Der REST API Explorer wird in Ihrem Browser geöffnet.

Betriebsinformationen anzeigen

Im REST API Explorer können Sie auf einen beliebigen Vorgang klicken, um die Konfigurationsinformationen für die Ressource anzuzeigen.

Die folgende Tabelle enthält Informationen zu den Abschnitten, die für Ressourcen im REST API Explorer verfügbar sind. Die Verfügbarkeit von Abschnitten variiert je nach HTTP-Methode. Nicht bei allen Methoden sind alle Abschnitte in der Tabelle aufgeführt.

Abschnitt	Beschreibung
Körperparameter	Stellt alle Felder für den Anforderungstext und unterstützte Werte für jedes Feld bereit.
Parameter	Stellt Informationen zu den verfügbaren Abfrageparametern bereit.
Antworten	Informiert über die möglichen HTTP Statuscodes für die Ressource. Wenn du klickst **Anfrage senden**, dieser Abschnitt enthält auch die Antwort des Server und die cURL-, Python- und Ruby-Syntax, die zum Senden der angegebenen Anfrage erforderlich ist. **Hinweis:** Klicken **Modell** um Beschreibungen der Felder anzuzeigen, die in einer Antwort zurückgegeben wurden.

Identifizieren Sie Objekte auf dem ExtraHop-System

Um API-Operationen für ein bestimmtes Objekt auszuführen, müssen Sie die Objekt-ID ermitteln. Sie können die Objekt-ID mithilfe der folgenden Methoden im REST API Explorer leicht finden.

- Die Objekt-ID wird in den Headern bereitgestellt, die von einer POST-Anfrage zurückgegeben werden. Wenn Sie beispielsweise eine POST-Anfrage senden, um eine Seite zu erstellen, zeigen die Antwortheader eine Standort-URL an.

  Die folgende Anfrage gab den Speicherort für das neu erstellte Tag als zurück `/api/v1/tags/1` und die ID für das Tag als `1`.

```
{
 "date": "Tue, 09 Nov 2021 18:21:00 GMT ",
 "via": "1.1 localhost",
 "server": "Apache",
 "content-type": "text/plain; charset=utf-8",
 "location": "/api/v1/tags/1",
 "cache-control": "private, max-age=0",
 "connection": "Keep-Alive",
 "keep-alive": "timeout=90, max=100",
 "content-length": "0"
}
```

- Die Objekt-ID wird für alle Objekte bereitgestellt, die von einer GET-Anfrage zurückgegeben werden. Wenn Sie beispielsweise eine GET-Anfrage auf allen Geräten ausführen, enthält der Antworttext Informationen für jedes Gerät, einschließlich der ID.

  Der folgende Antworttext zeigt einen Eintrag für ein einzelnes Gerät mit der ID 10212 an:

```
{
 "mod_time": 1448474346504,
 "node_id": null,
 "id": 10212,
 "extrahop_id": "test0001",
 "description": null,
 "user_mod_time": 1448474253809,
 "discover_time": 1448474250000,
 "vlanid": 0,
 "parent_id": 9352,
 "macaddr": "00:05:G3:FF:FC:28",
 "vendor": "Cisco",
 "is_l3": true,
 "ipaddr4": "10.10.10.5",
 "ipaddr6": null,
 "device_class": "node",
 "default_name": "Cisco5",
 "custom_name": null,
 "cdp_name": "",
 "dhcp_name": "",
 "netbios_name": "",
 "dns_name": "",
 "custom_type": "",
 "analysis_level": 1
},
```

ExtraHop API-Ressourcen

Sie können über die ExtraHop REST API Operationen für die folgenden Ressourcen ausführen. Sie können auch detailliertere Informationen zu diesen Ressourcen einsehen, z. B. verfügbare HTTP Methoden, Abfrageparameter und Objekteigenschaften im REST API Explorer.

Karte der Aktivitäten

Eine Aktivitätsdiagramm ist eine dynamische visuelle Darstellung der L4-L7-Protokollaktivität zwischen Geräten in Ihrem Netzwerk. Erstellen Sie in Echtzeit ein 2D- oder 3D-Layout von Geräteverbindungen, um mehr über den Verkehrsfluss und die Beziehungen zwischen Geräten zu erfahren.

Hier sind einige wichtige Überlegungen zu Activity Maps:

- In Standard Analysis und Erweiterte Analyse Analysis können Sie nur Aktivitätskarten für Geräte erstellen. Geräte im Entdeckungsmodus sind nicht in Activity Maps enthalten. Weitere Informationen finden Sie unter Analysestufen.
- Wenn Sie eine Aktivitätsdiagramm für ein Gerät, eine Aktivitätsgruppe oder eine Gerätegruppe ohne Protokollaktivität im ausgewählten Zeitintervall erstellen, wird die Map ohne Daten angezeigt. Ändern Sie das Zeitintervall oder Ihre Herkunftsauswahl und versuchen Sie es erneut.
- Sie können eine Aktivitätsdiagramm in einem erstellen Konsole um die Geräteverbindungen all Ihrer Sensoren zu sehen.

Weitere Informationen zum Konfigurieren und Navigieren in Activity Maps finden Sie unter Karten der Aktivitäten.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /activitymaps	Ruft alle Aktivitätskarten ab.
POST /activitymaps	Erstellen Sie eine neue Aktivitätsdiagramm.
POST /activitymaps/query	Führen Sie eine Netzwerktopologieabfrage durch, die Aktivitätsdiagramm Map-Daten als Flatfile-Inhalt zurückgibt.
/activitymaps/ {id} LÖSCHEN	Löscht eine bestimmte Aktivitätsdiagramm.
GET /activitymaps/ {id}	Rufen Sie eine bestimmte Aktivitätsdiagramm ab.
PATCH /activitymaps/ {id}	Aktualisieren Sie eine bestimmte Aktivitätsdiagramm.
POST /activitymaps/ {id} /query	Führen Sie eine Topologieabfrage für eine bestimmte Aktivitätsdiagramm durch, die Aktivitätsdiagramm Map-Daten als Flatfile-Inhalt zurückgibt.
GET /activitymaps/ {id} /sharing	Rufen Sie die Benutzer und ihre Freigabeberechtigungen für eine bestimmte Aktivitätsdiagramm ab.
PATCH /activitymaps/ {id} /sharing	Aktualisieren Sie die Benutzer und ihre Freigabeberechtigungen für eine bestimmte Aktivitätsdiagramm.

Betrieb	Beschreibung
PUT /activitymaps/ {id} /sharing	Ersetzen Sie die Benutzer und ihre Freigabeberechtigungen für eine bestimmte Aktivitätsdiagramm.

Einzelheiten der Operation

`POST /activitymaps`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Eigenschaften der Aktivitätsdiagramm.

`name:` ***Schnur***

Der freundliche Name für die Aktivitätsdiagramm.

`short_code:` ***Schnur***

(Optional) Der eindeutige Kurzcode, der global für alle Activity Maps gilt.

`description:` ***Schnur***

Die Beschreibung für die Aktivitätsdiagramm.

`weighting:` ***Schnur***

(Optional) Der Metrikwert, der bestimmt, wie Aktivitäten zwischen Geräten gewichtet werden. Unterstützte Elementwerte sind „Bytes“, „Verbindungen“ und „Turns“.

`mode:` ***Schnur***

(Optional) Das Layout der Aktivitätsdiagramm. Unterstützte Werte sind „2dforce“ und „3dforce“.

`show_alert_status:` ***Boolescher Wert***

(Optional) Gibt an, ob der Alarmstatus für Geräte auf der Aktivitätsdiagramm werden soll. Wenn diese Option aktiviert ist, steht die Farbe jedes Geräts auf der Karte für die schwerwiegendste Warnstufe, die dem Gerät zugeordnet ist.

`walks:` ***Reihe von Objekten***

Die Liste von einem oder mehreren Wanderobjekten. Ein Spaziergang ist ein Verkehrsweg, der aus einer oder mehreren Stufen besteht. Jeder Walk beginnt mit einem oder mehreren Ursprungsgeräten und erweitert sich auf Verbindungen zu Peer-Geräten, die auf Protokollaktivitäten basieren. Jede Erweiterung vom Ursprung aus ist ein Schritt. Der Inhalt des Objekts wird im Abschnitt „Gehen“ unten definiert.

`origins:` ***Reihe von Objekten***

Die Liste eines oder mehrerer Ursprungsgeräte des ersten Schritts innerhalb des Spaziergangs. Der Objektinhalt wird im Abschnitt „source_object“ unten definiert.

`object_type:` ***Schnur***

Der Quelltyp der Metrik.

Die folgenden Werte sind gültig:

- `device`
- `device_group`

`object_id:` ***Zahl***

Der eindeutige Bezeichner für das Quellobjekt.

`steps:` ***Reihe von Objekten***

Die Liste von einem oder mehreren Schritten innerhalb des Spaziergangs. Jeder Schritt wird durch die Protokollaktivität zwischen Geräten des vorherigen Schritts und einer

neuen Gruppe von Peer-Geräten definiert. Der Objektinhalt wird im Abschnitt „Schritt“ unten definiert.

`relationships:` ***Reihe von Objekten***

(Optional) Die Liste mit einem oder mehreren Filtern, die die Beziehung zwischen zwei Geräten definieren. Die Filter geben an, nach welchen Rollen und Protokollen gesucht werden soll, wenn Peer-Geräte in diesem Schritt gefunden werden. Beziehungen werden in der Aktivitätsdiagramm als Rand dargestellt. Objektinhalte werden im Abschnitt „Beziehung“ weiter unten definiert. Wenn kein Wert angegeben ist, sucht der Vorgang nach allen Peers.

`protocol:` ***Schnur***

(Optional) Das mit der Beziehung verknüpfte Metrikprotokoll, z. B. „HTTP“ oder „DNS“. Der Vorgang sucht nur nach Verbindungen zwischen Geräten über das angegebene Protokoll.

`role:` ***Schnur***

(Optional) Die Geräterolle, die dem Metrik Protokoll der Beziehung zugeordnet ist. Der Vorgang sucht nur nach Verbindungen zwischen Geräten über das zugehörige Protokoll in der angegebenen Rolle. Unterstützte Rollenwerte sind „Client“, „Server“ oder „Any“. Auf „any“ setzen, um alle Client-, Server- und Peer-Gerätebeziehungen zu finden, die dem angegebenen Protokoll zugeordnet sind.

`peer_in:` ***Reihe von Objekten***

(Optional) Die Liste von einem oder mehreren Peer-Geräteobjekten, die in die Activity Map aufgenommen werden sollen. Nur Beziehungen zu Peers des angegebenen Quellobjekts sind enthalten. Der Objektinhalt wird im Abschnitt „source_object“ unten definiert.

`object_type:` ***Schnur***

Der Quelltyp der Metrik.

Die folgenden Werte sind gültig:

- `device`
- `device_group`

`object_id:` ***Zahl***

Der eindeutige Bezeichner für das Quellobjekt.

`peer_not_in:` ***Reihe von Objekten***

(Optional) Die Liste von einem oder mehreren Peer-Geräteobjekten, die von der Aktivitätsdiagramm ausgeschlossen werden sollen. Beziehungen zu Peers des angegebenen Quellobjekts sind ausgeschlossen. Der Objektinhalt wird im Abschnitt „source_object“ unten definiert.

`object_type:` ***Schnur***

Der Quelltyp der Metrik.

Die folgenden Werte sind gültig:

- `device`
- `device_group`

`object_id:` ***Zahl***

Der eindeutige Bezeichner für das Quellobjekt.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "mode": "string",
```

```
    "name": "string",
    "short_code": "string",
    "show_alert_status": true,
    "walks": {
        "origins": {
            "object_type": "string",
            "object_id": 0
        },
        "steps": {
            "relationships": {
                "protocol": "string",
                "role": "string"
            },
            "peer_in": {
                "object_type": "string",
                "object_id": 0
            },
            "peer_not_in": {
                "object_type": "string",
                "object_id": 0
            }
        }
    },
    "weighting": "string"
}
```

`POST /activitymaps/query`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Eigenschaften der Topologieabfrage.

`from:` ***Zahl***

Der Anfangszeitstempel des Zeitbereichs, den die Abfrage durchsucht, ausgedrückt in Millisekunden seit der Epoche.

`until:` ***Zahl***

(Optional) Der letzte Zeitstempel des Zeitbereichs, den die Abfrage durchsucht, ausgedrückt in Millisekunden seit der Epoche. Wenn kein Wert gesetzt ist, wird das Abfrageende standardmäßig auf „now“ gesetzt.

`weighting:` ***Schnur***

(Optional) Der Metrikwert, der bestimmt, wie Aktivitäten zwischen Geräten gewichtet werden.

Die folgenden Werte sind gültig:

- `bytes`
- `connections`
- `turns`

`edge_annotations:` ***Reihe von Zeichenketten***

(Optional) Die Liste mit einer oder mehreren Kantenanmerkungen, die in die Topologieabfrage aufgenommen werden sollen.

Die folgenden Werte sind gültig:

- `protocols`
- `appearances`

`walks:` ***Reihe von Objekten***

Die Liste von einem oder mehreren Walk-Objekten, die in die Topologieabfrage aufgenommen werden sollen. Ein Spaziergang ist ein Verkehrsweg, der aus einer oder mehreren Stufen besteht. Jeder Walk beginnt mit einem oder mehreren Ursprungsgeräten und erweitert sich auf Verbindungen zu Peer-Geräten, die auf Protokollaktivitäten basieren. Jede Erweiterung vom Ursprung aus ist ein Schritt. Der Objektinhalt wird im Abschnitt „topology_walk“ unten definiert.

`origins:` ***Reihe von Objekten***

Die Liste eines oder mehrerer Ursprungsgeräte des ersten Schritts innerhalb des Spaziergangs. Der Objektinhalt wird im Abschnitt „topology_source“ unten definiert.

`object_type:` ***Schnur***

Der Typ des Quellobjekts.

Die folgenden Werte sind gültig:

- `all_devices`
- `device_group`
- `device`

`object_id:` ***Zahl***

Der eindeutige Bezeichner für das Quellobjekt. Auf 0 setzen, wenn der Wert des Parameter „object_type“ „all_devices“ ist.

`steps:` ***Reihe von Objekten***

Die Liste von einem oder mehreren Schritten innerhalb des Spaziergangs. Jeder Schritt wird durch die Protokollaktivität zwischen Geräten des vorherigen Schritts und einer neuen Gruppe von Peer-Geräten definiert. Objektinhalte werden im Abschnitt „topology_step“ unten definiert.

`relationships:` ***Reihe von Objekten***

(Optional) Die Liste mit einem oder mehreren Filtern, die die Beziehung zwischen zwei Geräten definieren. Die Filter geben an, nach welchen Rollen und Protokollen gesucht werden soll, wenn Peer-Geräte in diesem Schritt gefunden werden. Beziehungen werden in der Aktivitätsdiagramm als Rand dargestellt. Wenn kein Wert festgelegt ist, umfasst die Operation alle Peers. Der Objektinhalt wird im Abschnitt „topology_relationship“ weiter unten definiert.

`role:` ***Schnur***

(Optional) Die Rolle des Peer-Geräts im Verhältnis zum Ursprungsgerät.

Die folgenden Werte sind gültig:

- `client`
- `server`
- `any`

`protocol:` ***Schnur***

(Optional) Das Protokoll, über das das Ursprungsgerät kommuniziert, z. B. „HTTP“. Wenn kein Wert festgelegt ist, enthält das Objekt ein beliebiges Protokoll.

`peer_in:` ***Reihe von Objekten***

(Optional) Die Liste von einem oder mehreren Peer-Geräten, die in das Topologiediagramm aufgenommen werden sollen. Nur Beziehungen zu Peers des angegebenen Quellobjekts sind enthalten. Der Objektinhalt wird im Abschnitt „topology_source“ unten definiert.

`object_type:` ***Schnur***

Der Typ des Quellobjekts.

Die folgenden Werte sind gültig:

- `all_devices`
- `device_group`
- `device`

`object_id:` ***Zahl***

Der eindeutige Bezeichner für das Quellobjekt. Auf 0 setzen, wenn der Wert des Parameter „object_type“ „all_devices“ ist.

`peer_not_in:` ***Reihe von Objekten***

(Optional) Die Liste von einem oder mehreren Peer-Geräten, die aus dem Topologiediagramm ausgeschlossen werden sollen. Beziehungen zu Peer-Geräten des angegebenen Quellobjekts sind ausgeschlossen. Der Objektinhalt wird im Abschnitt „topology_source“ unten definiert.

`object_type:` ***Schnur***

Der Typ des Quellobjekts.

Die folgenden Werte sind gültig:

- `all_devices`
- `device_group`
- `device`

`object_id:` ***Zahl***

Der eindeutige Bezeichner für das Quellobjekt. Auf 0 setzen, wenn der Wert des Parameter „object_type“ „all_devices“ ist.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "edge_annotations": [],
    "from": 0,
    "until": 0,
    "walks": {
        "origins": {
            "object_type": "string",
            "object_id": 0
        },
        "steps": {
            "relationships": {
                "role": "string",
                "protocol": "string"
            },
            "peer_in": {
                "object_type": "string",
                "object_id": 0
            },
            "peer_not_in": {
                "object_type": "string",
                "object_id": 0
            }
        }
    },
    "weighting": "string"
}
```

`GET /activitymaps`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": 0,
    "mod_time": 0,
    "mode": "string",
    "name": "string",
    "owner": "string",
    "rights": [
        "string"
    ],
    "short_code": "string",
    "show_alert_status": true,
    "walks": [],
    "weighting": "string"
}
```

`GET /activitymaps/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Aktivitätsdiagramm.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": 0,
    "mod_time": 0,
    "mode": "string",
    "name": "string",
    "owner": "string",
    "rights": [
        "string"
    ],
    "short_code": "string",
    "show_alert_status": true,
    "walks": [],
    "weighting": "string"
}
```

`POST /activitymaps/{id}/query`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Aktivitätsdiagramm.

`body:` ***Objekt***

Die Eigenschaften der Topologieabfrage.

`from:` ***Zahl***

Der Anfangszeitstempel des Zeitbereichs, den die Abfrage durchsucht, ausgedrückt in Millisekunden seit der Epoche.

`until:` ***Zahl***

(Optional) Der letzte Zeitstempel des Zeitbereichs, den die Abfrage durchsucht, ausgedrückt in Millisekunden seit der Epoche. Wenn kein Wert gesetzt ist, wird das Abfrageende standardmäßig auf „now“ gesetzt.

`edge_annotations:` ***Reihe von Zeichenketten***

(Optional) Die Liste mit einer oder mehreren Kantenanmerkungen, die in die Topologieabfrage aufgenommen werden sollen.

Die folgenden Werte sind gültig:

- `protocols`
- `appearances`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "edge_annotations": [],
    "from": 0,
    "until": 0
}
```

`DELETE /activitymaps/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Aktivitätsdiagramm.

`PATCH /activitymaps/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Aktivitätsdiagramm.

`body:` ***Objekt***

Die Eigenschaften der Aktivitätsdiagramm, die aktualisiert werden sollen.

`GET /activitymaps/{id}/sharing`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Aktivitätsdiagramm.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "anyone": "string",
    "groups": {},
    "users": {}
}
```

`PUT /activitymaps/{id}/sharing`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Benutzer und ihre Berechtigungsstufen.

`id:` ***Zahl***

Die eindeutige Kennung für die Aktivitätsdiagramm.

```
PATCH /activitymaps/{id}/sharing
```

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Benutzer und ihre Berechtigungsstufen.

`id:` ***Zahl***
Die eindeutige Kennung für die Aktivitätsdiagramm.

Warnung

Alerts sind Systembenachrichtigungen, die nach bestimmten Warnungskriterien generiert werden. Standardwarnungen sind im System verfügbar, oder Sie können eine benutzerdefinierte Alarm erstellen.

Erkennungen und Schwellenwertwarnungen können so eingestellt werden, dass Sie Alarm werden, wenn eine Metrik den in der Warnungskonfiguration definierten Wert überschreitet. Trendwarnungen können nicht über die REST-API konfiguriert werden. Weitere Informationen finden Sie unter Warnmeldungen.

Hinweis: Erkennungen durch maschinelles Lernen erfordern eine Verbindung zu ExtraHop Cloud Services.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /alerts	Rufen Sie alle Benachrichtigungen ab.
POST/Benachrichtigungen	Erstellen Sie eine neue Alarm mit bestimmten Werten.
/alerts/ {id} LÖSCHEN	Löschen Sie eine bestimmte Alarm.
GET /alerts/ {id}	Rufen Sie eine bestimmte Alarm ab.
PATCH /alerts/ {id}	Wenden Sie Aktualisierungen auf eine bestimmte Alarm an.
GET /alerts/ {id} /applications	Rufen Sie alle Anwendungen ab, denen eine bestimmte Alarm zugewiesen wurde.
POST /alerts/ {id} /Anwendungen	Weisen Sie Anwendungen eine bestimmte Alarm zu und heben Sie deren Zuweisung auf.
LÖSCHEN Sie /alerts/ {id} /applications/ {child-id}	Heben Sie die Zuweisung einer Anwendung zu einer bestimmten Alarm auf.
POST /alerts/ {id} /Anwendungen/ {Child-ID}	Weisen Sie eine Anwendung einer bestimmten Alarm zu.
GET /alerts/ {id} /devicegroups	Alles abrufen Gerätegruppen denen eine bestimmte Alarm zugewiesen wurde.
POST /alerts/ {id} /devicegroups	Weisen Sie Gerätegruppen eine bestimmte Alarm zu und heben Sie deren Zuweisung auf.
LÖSCHEN Sie /alerts/ {id} /devicegroups/ {child-id}	Heben Sie die Zuweisung einer Gerätegruppe zu einer bestimmten Alarm auf.
POST /alerts/ {id} /devicegroups/ {child-id}	Weisen Sie einer bestimmten Alarm eine Gerätegruppe zu.

Betrieb	Beschreibung
GET /alerts/ {id} /devices	Ruft alle Geräte ab, denen eine bestimmte Alarm zugewiesen wurde.
POST /alerts/ {id} /Geräte	Weisen Sie Geräten eine bestimmte Alarm zu und heben Sie deren Zuweisung auf.
LÖSCHEN /alerts/ {id} /devices/ {child-id}	Heben Sie die Zuweisung eines Gerät zu einer bestimmten Alarm auf.
POST /alerts/ {id} /devices/ {child-id}	Weisen Sie einem bestimmten Alarm ein Gerät zu.
GET /alerts/ {id} /emailgroups	Ruft alle E-Mail-Gruppen ab, denen eine bestimmte Alarm zugewiesen wurde.
POST /alerts/ {id} /emailgroups	Weisen Sie E-Mail-Gruppen eine bestimmte Alarm zu und heben Sie deren Zuweisung auf.
LÖSCHEN Sie /alerts/ {id} /emailgroups/ {child-id}	Heben Sie die Zuweisung einer E-Mail-Gruppe zu einer bestimmten Alarm auf.
POST /alerts/ {id} /emailgroups/ {child-id}	Weisen Sie einer bestimmten Alarm eine E-Mail-Gruppe zu.
GET /alerts/ {id} /exclusionintervals	Ruft alle Ausschlussintervalle ab, denen eine bestimmte Alarm zugewiesen wurde.
POST /alerts/ {id} /exclusionintervals	Weisen Sie Ausschlussintervallen eine bestimmte Alarm zu und heben Sie deren Zuweisung auf.
LÖSCHEN /alerts/ {id} /exclusionintervals/ {child-id}	Heben Sie die Zuweisung eines Ausschlussintervalls zu einer bestimmten Alarm auf.
POST /alerts/ {id} /exclusionintervals/ {child-id}	Weisen Sie einer bestimmten Alarm ein Ausschlussintervall zu.
GET /alerts/ {id} /networks	Ruft alle Netzwerke ab, denen eine bestimmte Alarm zugewiesen wurde.
POST /alerts/ {id} /Netzwerke	Weisen Sie Netzwerken eine bestimmte Alarm zu und heben Sie deren Zuweisung auf.
LÖSCHEN /alerts/ {id} /networks/ {child-id}	Heben Sie die Zuweisung eines Netzwerk zu einer bestimmten Alarm auf.
POST /alerts/ {id} /networks/ {child-id}	Weisen Sie einer bestimmten Alarm ein Netzwerk zu.
GET /alerts/ {id} /stats	Rufen Sie alle zusätzlichen Statistiken für eine bestimmte Alarm ab.

Einzelheiten der Operation

```
GET /alerts
```

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "apply_all": true,
    "author": "string",
    "categories": [
        "string"
```

```
    ],
    "cc": [],
    "description": "string",
    "disabled": true,
    "field_name": "string",
    "field_name2": "string",
    "field_op": "string",
    "id": 0,
    "interval_length": 0,
    "mod_time": 0,
    "name": "string",
    "notify_snmp": true,
    "object_type": "string",
    "operand": "string",
    "operator": "string",
    "param": {},
    "param2": {},
    "protocols": [
        "string"
    ],
    "refire_interval": 0,
    "severity": 0,
    "stat_name": "string",
    "type": "string",
    "units": "string"
}
```

`POST /alerts`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die angegebenen Eigenschaftswerte auf die neue Alarm an.

`description:` ***Schnur***

Eine optionale Beschreibung für die Alarm.

`notify_snmp:` ***Boolesch***

(Optional) Gibt an, ob ein SNMP-Trap gesendet werden soll, wenn eine Alarm generiert wird.

`field_op:` ***Schnur***

Die Art des Vergleichs zwischen den Feldern field_name und field_name2 beim Anwenden eines Verhältnisses. Gilt nur für Schwellenwert-Alarme.

Die folgenden Werte sind gültig:

- `/`
- `null`

`stat_name:` ***Schnur***

Der Statistikname für die Alarm. Gilt nur für Schwellenwert-Alarme.

`disabled:` ***Boolesch***

(Optional) Gibt an, ob die Alarm deaktiviert ist.

`operator:` ***Schnur***

Der logische Operator, der angewendet wird, wenn der Wert des Operandenfeldes mit den Warnbedingungen verglichen wird. Gilt nur für Schwellenwert-Alarme.

Die folgenden Werte sind gültig:

- `==`
- `>`

- `<`
- `>=`
- `<=`

`operand:` ***Schnur***

Der Wert, der mit den Alarmbedingungen verglichen werden soll. Die Vergleichsmethode wird durch den Wert des Operatorfeldes spezifiziert. Gilt nur für Schwellenwert-Alarme.

`field_name:` ***Schnur***

Der Name der überwachten Metrik. Gilt nur für Schwellenwert-Alarme.

`name:` ***Schnur***

Der eindeutige, freundliche Name für die Alarm.

`cc:` ***Reihe von Zeichenketten***

Die Liste der E-Mail-Adressen, die nicht in einer E-Mail-Gruppe enthalten sind und die Benachrichtigungen erhalten sollen.

`apply_all:` ***Boolesch***

Gibt an, ob die Alarm allen verfügbaren Datenquellen zugewiesen ist.

`severity:` ***Zahl***

(Optional) Der Schweregrad der Alarm, der im Warnungsverlauf, in E-Mail-Benachrichtigungen und SNMP-Traps angezeigt wird. Die Schweregrade 0-2 erfordern sofortige Aufmerksamkeit. Die Schweregrade sind beschrieben in der REST-API-Leitfaden.

Die folgenden Werte sind gültig:

- `0`
- `1`
- `2`
- `3`
- `4`
- `5`
- `6`
- `7`

`author:` ***Schnur***

Der Name des Benutzers, der die Alarm erstellt hat.

`param:` ***Objekt***

Der erste Warnparameter, der entweder ein Schlüsselmuster oder ein Datenpunkt ist. Gilt nur für Schwellenwert-Alarme.

`interval_length:` ***Zahl***

Die Länge des Warnintervalls, ausgedrückt in Sekunden. Gilt nur für Schwellenwert-Alarme.

Die folgenden Werte sind gültig:

- `30`
- `60`
- `120`
- `300`
- `600`
- `900`
- `1200`
- `1800`

`param2:` ***Objekt***

Der zweite Warnparameter, der entweder ein Schlüsselmuster oder ein Datenpunkt ist. Gilt nur für Schwellenwert-Alarme.

`units:` ***Schnur***

Das Intervall, in dem der Alarmzustand bewertet werden soll. Gilt nur für Schwellenwert-Alarme.

Die folgenden Werte sind gültig:

- `none`
- `period`
- `1 sec`
- `1 min`
- `1 hr`

`field_name2:` ***Schnur***

Die zweite überwachte Metrik bei der Anwendung eines Verhältnisses. Gilt nur für Schwellenwert-Alarme.

`refire_interval:` ***Zahl***

(Optional) Das Zeitintervall, in dem die Alarmbedingungen überwacht werden, ausgedrückt in Sekunden.

Die folgenden Werte sind gültig:

- `300`
- `600`
- `900`
- `1800`
- `3600`
- `7200`
- `14400`

`type:` ***Schnur***

Die Art der Alarm.

Die folgenden Werte sind gültig:

- `threshold`

`object_type:` ***Schnur***

Der Typ der Metrikquelle, die von der Alert-Konfiguration überwacht wird. Gilt nur für Erkennungswarnungen.

Die folgenden Werte sind gültig:

- `application`
- `device`

`protocols:` ***Reihe von Zeichenketten***

(Optional) Die Liste der überwachten Protokolle. Gilt nur für Erkennungswarnungen.

`categories:` ***Reihe von Zeichenketten***

(Optional) Die Liste einer oder mehrerer Erkennungskategorien. Eine Alarm wird nur generiert, wenn eine Erkennung in den angegebenen Kategorien identifiziert wird. Gilt nur für Erkennungswarnungen.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "apply_all": true,
    "author": "string",
    "categories": [
        "string"
    ],
```

```
    "cc": [],
    "description": "string",
    "disabled": true,
    "field_name": "string",
    "field_name2": "string",
    "field_op": "string",
    "interval_length": 0,
    "name": "string",
    "notify_snmp": true,
    "object_type": "string",
    "operand": "string",
    "operator": "string",
    "param": {},
    "param2": {},
    "protocols": [
        "string"
    ],
    "refire_interval": 0,
    "severity": 0,
    "stat_name": "string",
    "type": "string",
    "units": "string"
}
```

`GET /alerts/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "apply_all": true,
    "author": "string",
    "categories": [
        "string"
    ],
    "cc": [],
    "description": "string",
    "disabled": true,
    "field_name": "string",
    "field_name2": "string",
    "field_op": "string",
    "id": 0,
    "interval_length": 0,
    "mod_time": 0,
    "name": "string",
    "notify_snmp": true,
    "object_type": "string",
    "operand": "string",
    "operator": "string",
    "param": {},
    "param2": {},
    "protocols": [
        "string"
    ],
    "refire_interval": 0,
    "severity": 0,
    "stat_name": "string",
    "type": "string",
```

```
    "units": "string"
}
```

`DELETE /alerts/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`PATCH /alerts/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Wenden Sie die angegebenen Eigenschaftswertaktualisierungen auf die Alarm an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`GET /alerts/{id}/stats`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "alert_id": 0,
    "field_name": "string",
    "id": 0,
    "param": "string",
    "stat_name": "string"
}
```

`GET /alerts/{id}/devicegroups`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/devicegroups`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Liste der eindeutigen Identifikatoren für Gerätegruppen, die der Alarm zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/devicegroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`DELETE /alerts/{id}/devicegroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`GET /alerts/{id}/emailgroups`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/emailgroups`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Liste der eindeutigen Identifikatoren für E-Mail-Gruppen, die der Warnung zugewiesen sind oder nicht zugewiesen wurden.

`assign:` ***Reihe von Zahlen***

IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***

IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/emailgroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die E-Mail-Gruppe.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`DELETE /alerts/{id}/emailgroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die E-Mail-Gruppe.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`GET /alerts/{id}/exclusionintervals`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/exclusionintervals`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Liste der eindeutigen Identifikatoren für Ausschlussintervalle, die der Alarm zugewiesen ist oder nicht.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/exclusionintervals/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Ausschlussintervall.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`DELETE /alerts/{id}/exclusionintervals/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Ausschlussintervall.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`GET /alerts/{id}/devices`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/devices`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Liste der eindeutigen Identifikatoren für Geräte, die der Alarm zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Gerät.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`DELETE /alerts/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Gerät.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`GET /alerts/{id}/networks`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/networks`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Liste der eindeutigen Identifikatoren für Netzwerke, die der Alarm zugewiesen sind oder nicht.
`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen
`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/networks/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Netzwerk.
`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`DELETE /alerts/{id}/networks/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Netzwerk.
`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`GET /alerts/{id}/applications`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/applications`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Liste der eindeutigen Identifikatoren für Anwendungen, die der Alarm zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***

IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***

IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`POST /alerts/{id}/applications/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Anwendung.

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`DELETE /alerts/{id}/applications/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Anwendung.

`id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

Schweregrad Warnung

Der Schweregrad, den Sie für eine Alarm angeben, wird auf der Seite Benachrichtigungen, E-Mail-Benachrichtigungen und SNMP-Traps angezeigt.

Die folgenden Schweregrad werden unterstützt. Schweregrade 0-2 erfordern sofortige Aufmerksamkeit.

Wert	Name	Beschreibung
0	Notfall	Die Systemfunktionalität ist nicht verfügbar.
1	Warnung	Sofortiges Handeln ist erforderlich.
2	Kritisch	Es treten kritische Bedingungen auf.
3	Fehler	Es treten Fehlerbedingungen auf.
4	Warnung	Es treten Warnbedingungen auf.

Wert	Name	Beschreibung
5	Hinweis	Normaler Betrieb findet unter erheblichen Bedingungen statt, z. B. bei einem Neustart.
6	Informationen	Bei Prozessaktualisierungen laufen normale Abläufe ab.
7	Debuggen	Meldungen auf Debug-Ebene sind verfügbar.

Priorität der Analyse

Das ExtraHop-System analysiert und klassifiziert den Traffic für jedes Gerät, das es entdeckt. Ihre Lizenz reserviert Kapazität für das ExtraHop-System, um Metriken für hoher Wert Geräte zu sammeln. Diese Kapazität ist mit zwei Analyseebenen verknüpft: Fortgeschrittene Analyse und Standardanalyse.

Sie können angeben, welche Geräte die Stufen Erweiterte Analyse und Standard Analysis erhalten, indem Sie Folgendes konfigurieren Regeln für die Analysepriorität. Analyseprioritäten helfen dabei, das ExtraHop-System darüber zu informieren, welche Geräte in Ihrer Umgebung wichtig sind. Eine dritte Analyseebene, der Entdeckungsmodus, ist für Geräte verfügbar, die sich nicht in Advanced oder Standard Analysis befinden.

Hinweis: Standardmäßig verwaltet jeder Sensor seine eigenen Analyseprioritäten. Wenn der Sensor an eine Konsole angeschlossen ist, können Sie diese zentral verwalten gemeinsame Systemeinstellungen von der Konsole aus.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
HOLEN SIE SICH /analysispriority/config/ {sensor_id}	Rufen Sie die Analyseprioritätsregeln für eine bestimmte Sensor.
PUT /analysispriority/config/ {sensor_id}	Ersetzen Sie die Analyseprioritätsregeln für eine bestimmte Sensor.
GET /analysispriority/ {sensor_id} /manager	Rufen Sie das System ab, das für die Verwaltung der Analyseprioritätsregeln für das konfiguriert ist Sensor.
PATCH /analysispriority/ {sensor_id} /manager	Aktualisieren Sie das System, das die Analyseprioritätsregeln für eine bestimmte Gruppe verwaltet Sensor.

Einzelheiten der Operation

```
GET /analysispriority/{appliance_id}/manager
```

Geben Sie die folgenden Parameter an.

`appliance_id`: ***Zahl***
Die Kennung für den lokalen Sensor. Dieser Wert muss auf 0 gesetzt werden.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "manager": {}
```

```
}
```

`GET /analysispriority/config/{appliance_id}`

Geben Sie die folgenden Parameter an.

`appliance_id:` ***Zahl***

Die Kennung für einen Sensor. Setzen Sie diesen Wert auf 0, wenn Sie einen Sensor aufrufen.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "advanced_rules": [],
    "autofill_advanced": true,
    "autofill_standard": true,
    "is_in_effect": true,
    "standard_rules": []
}
```

`PUT /analysispriority/config/{appliance_id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Eigenschaften der Regeln für die Prioritätsanalyse.

`autofill_advanced:` ***Boolescher Wert***

Gibt an, ob Geräte automatisch in Erweiterte Analyse platziert werden sollen, bis die Kapazität erreicht ist. Geräte in der Liste advanced_rules werden priorisiert, gefolgt von Geräten in der Liste standard_rules und dann nach der Erkennungszeit für das Gerät. Die Kapazität für Erweiterte Analyse wird durch die ExtraHop-Systemlizenz bestimmt.

`advanced_rules:` ***Reihe von Objekten***

(Optional) Die Erweiterte Analyse Analysis-Prioritätsregeln für eine Gerätegruppe.

`type:` ***Schnur***

Der Gruppentyp, für den die Prioritätsregeln für die Analyse gelten.

Die folgenden Werte sind gültig:

- `device_group`

`object_id:` ***Zahl***

Die eindeutige Kennung für die Gruppe.

`description:` ***Schnur***

(Optional) Die Beschreibung der Prioritätsregeln für Analysen.

`autofill_standard:` ***Boolescher Wert***

Gibt an, ob Geräte automatisch in die Standardanalyse aufgenommen werden sollen, bis die Gesamtkapazität erreicht ist. Geräte in der Liste standard_rules werden priorisiert, gefolgt von der Erkennungszeit für das Gerät. Die Gesamtkapazität wird durch die ExtraHop-Systemlizenz bestimmt.

`standard_rules:` ***Reihe von Objekten***

(Optional) Die Standardanalyse-Prioritätsregeln für eine Gerätegruppe.

`type:` ***Schnur***

Der Gruppentyp, für den die Prioritätsregeln für die Analyse gelten.

Die folgenden Werte sind gültig:

- `device_group`

`object_id:` ***Zahl***

Die eindeutige Kennung für die Gruppe.

`description:` ***Schnur***

(Optional) Die Beschreibung der Prioritätsregeln für Analysen.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "advanced_rules": {
        "type": "string",
        "object_id": 0,
        "description": "string"
    },
    "autofill_advanced": true,
    "autofill_standard": true,
    "standard_rules": {
        "type": "string",
        "object_id": 0,
        "description": "string"
    }
}
```

`appliance_id:` ***Zahl***

Die Kennung für einen Sensor. Setzen Sie diesen Wert auf 0, wenn Sie einen Sensor aufrufen.

`PATCH /analysispriority/{appliance_id}/manager`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die ID des Sensor oder der Konsole, die die Prioritätsregeln für die Analyse des lokalen Sensor verwaltet. Setzen Sie diesen Wert auf 0, um die Verwaltung des lokalen Sensor wiederherzustellen.

`manager:` ***Zahl***

Die eindeutige Kennung für den verwaltenden Sensor oder die verwaltende Konsole.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "manager": 0
}
```

`appliance_id:` ***Zahl***

Die Kennung für den lokalen Sensor. Dieser Wert muss auf 0 gesetzt werden.

API-Schlüssel

Ein API-Schlüssel ermöglicht es einem Benutzer, Operationen über die ExtraHop REST API durchzuführen.

Sie können den ersten API-Schlüssel für das Setup-Benutzerkonto über die REST-API generieren. Alle anderen API-Schlüssel werden über die API-Zugriffsseite in den Administrationseinstellungen generiert.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
HOLEN SIE SICH /apikeys	Ruft alle API-Schlüssel ab.
POST /apikeys	Erstellen Sie den ersten API-Schlüssel für das Setup-Benutzerkonto.

Betrieb	Beschreibung
GET /apikeys/ {keyid}	Rufen Sie Informationen zu einem bestimmten API-Schlüssel ab.

Einzelheiten der Operation

`GET /apikeys`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": 0,
    "key": "string",
    "time_added": 0,
    "user_id": 0,
    "username": "string"
}
```

`GET /apikeys/{keyid}`

Geben Sie die folgenden Parameter an.

`keyid:` ***Zahl***

Die eindeutige Kennung für den API-Schlüssel.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": 0,
    "key": "string",
    "time_added": 0,
    "user_id": 0,
    "username": "string"
}
```

`POST /apikeys`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Das Passwort des Setup-Benutzers.

`password:` ***Schnur***

Das Passwort für den Setup-Benutzer.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "password": "string"
}
```

Gerät

Das ExtraHop-System besteht aus einem Netzwerk verbundener ExtraHop-Appliances, wie Sensoren, Konsolen, Recordstores und Packetstores, die Aufgaben wie das Überwachen des Datenverkehrs, das Analysieren von Daten, das Speichern von Daten und das Identifizieren von Erkennungen ausführen.

Sie können Informationen abrufen und Verbindungen für lokale und entfernte ExtraHop-Appliances herstellen.

Hinweis: Sie können nur eine Verbindung zwischen ähnlichen ExtraHop-Appliances wie RevealX Enterprise oder Performance herstellen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /appliances	Ruft alle Remote-ExtraHop-Appliances ab, die mit der lokalen Appliance verbunden sind.
POST /Geräte	Stellen Sie eine neue Verbindung zu einer Remote-ExtraHop-Appliance her.
LÖSCHE /Appliances/ {id}	Trennen Sie eine bestimmte ExtraHop-Appliance von dieser Konsole.
GET /Appliances/ {id}	Ruft eine bestimmte Remote-ExtraHop-Appliance ab, die mit der lokalen Appliance verbunden ist (nur gültig auf Konsolen).
HOLEN SIE SICH /Appliances/ {id} /cloudservices	Rufen Sie den Status der ExtraHop Cloud Services auf dieser Appliance ab.
POST /Appliances/ {id} /cloudservices	Ändern Sie die ExtraHop Cloud Services-Einstellungen auf dieser Appliance.
HOLEN SIE SICH /Appliances/ {id} /productkey	Ruft den Produktschlüssel für eine angegebene Appliance ab (nur gültig auf Konsolen).
GET /Appliances/ {ids_id} /association	Ruft die ID des Paketsensor ab, mit dem der IDS-Sensor verbunden ist.
POST /Appliances/ {ids_id} /association	Verbinden Sie einen IDS-Sensor mit einem Paketsensor.
GET /Appliances/firmware/next	Rufen Sie Firmware-Versionen ab, auf die Remote-ExtraHop-Systeme aktualisiert werden können (nur gültig auf Konsolen).
POST /Appliances/Firmware/Upgrade	Aktualisieren Sie die Firmware auf externen ExtraHop-Systemen, die mit dem lokalen System verbunden sind. Firmware-Images werden von ExtraHop Cloud Services heruntergeladen (nur gültig auf Konsolen).
`GET /appliances/{ids_id}/association`	Ruft die ID des Paketsensor ab, mit dem der IDS-Sensor verbunden ist (nur gültig auf Konsolen).
`POST /appliances/{ids_id}/association`	Verbindet einen IDS-Sensor mit einem Paketsensor (nur gültig auf Konsolen).

Einzelheiten der Operation

`GET /appliances`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "add_time": 0,
    "advanced_analysis_capacity": 0,
    "analysis_levels_managed": true,
    "connection_type": "string",
    "data_access": true,
    "display_name": "string",
    "fingerprint": "string",
    "firmware_version": "string",
    "hostname": "string",
    "id": 0,
    "license_platform": "string",
    "license_status": "string",
    "licensed_features": {},
    "licensed_modules": [
        "string"
    ],
    "managed_by_local": true,
    "manages_local": true,
    "nickname": "string",
    "platform": "string",
    "status_message": "string",
    "sync_time": 0,
    "total_capacity": 0,
    "uuid": "string"
}
```

`GET /appliances/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Geben Sie den eindeutigen Bezeichner für die Appliance an. Geben Sie 0 an, um die lokale Appliance auszuwählen.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "add_time": 0,
    "advanced_analysis_capacity": 0,
    "analysis_levels_managed": true,
    "connection_type": "string",
    "data_access": true,
    "display_name": "string",
    "fingerprint": "string",
    "firmware_version": "string",
    "hostname": "string",
    "id": 0,
    "license_platform": "string",
    "license_status": "string",
    "licensed_features": {},
    "licensed_modules": [
        "string"
    ],
```

```
    "managed_by_local": true,
    "manages_local": true,
    "nickname": "string",
    "platform": "string",
    "status_message": "string",
    "sync_time": 0,
    "total_capacity": 0,
    "uuid": "string"
}
```

`GET /appliances/{ids_id}/association`

Geben Sie die folgenden Parameter an.

`ids_id:` ***Zahl***
Geben Sie die ID des IDS-Sensors an.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "associated_sensor_id": 0
}
```

`POST /appliances/{ids_id}/association`

Geben Sie die folgenden Parameter an.

`ids_id:` ***Zahl***
Geben Sie die ID des IDS-Sensors an.

`body:` ***Objekt***
Geben Sie die ID des Paketsensor an.

`associated_sensor_id:` ***Zahl***
Die ID des Paketsensor.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "associated_sensor_id": 0
}
```

`GET /appliances/{id}/productkey`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Geben Sie den eindeutigen Bezeichner für die Appliance an.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "product_key": "string"
}
```

`GET /appliances/firmware/next`

Geben Sie die folgenden Parameter an.

`ids:` ***Schnur***

(Optional) Eine CSV-Liste mit eindeutigen Identifikatoren für die Remote-Appliances. Wenn dieser Parameter angegeben ist, gibt der Vorgang Firmware-Versionen zurück, auf die jedes der angegebenen Remote-Geräte aktualisiert werden kann. Wenn dieser Parameter nicht angegeben ist, gibt der Vorgang Firmware-Versionen zurück, auf die jedes Remote-Gerät aktualisiert werden kann.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "release": "string",
    "versions": []
}
```

`GET /appliances/{id}/cloudservices`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Geben Sie den eindeutigen Bezeichner für die Appliance an. Dieser Wert muss auf 0 gesetzt werden, wodurch die lokale Appliance ausgewählt wird.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "connection_status": "string",
    "connection_status_color": "string",
    "enabled_services": [],
    "last_active_time": 0,
    "last_analyzed_time": 0
}
```

`POST /appliances/{id}/cloudservices`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Geben Sie den eindeutigen Bezeichner für die Appliance an. Dieser Wert muss auf 0 gesetzt werden, wodurch die lokale Appliance ausgewählt wird.

`body:` ***Objekt***

Geben Sie die Aktion an, um die Einstellungen der ExtraHop Cloud Services zu ändern.

`action:` ***Schnur***

Geben Sie die Aktion an, um die Einstellungen der ExtraHop Cloud Services zu ändern.

Die folgenden Werte sind gültig:

- `unenroll`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "action": "string"
}
```

`POST /appliances/firmware/upgrade`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Firmware-Upgrade-Optionen.

`version:` ***Schnur***

Die Firmware-Version, auf die Appliances aktualisiert werden sollen. Sie können eine Liste der gültigen Versionen mit der Operation GET /api/v1/appliances/firmware/next abrufen.

`system_ids:` ***Reihe von Zahlen***

Eine Liste eindeutiger Identifikatoren für die Remote-Appliances. Sie können Appliance-IDs mit der Operation GET /api/v1/appliances abrufen; Appliance-IDs werden in den ID-Feldern der Antwort zurückgegeben.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "system_ids": [],
    "version": "string"
}
```

`POST /appliances`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Geben Sie die Eigenschaften der neuen Verbindung an.

`host:` ***Schnur***

Der Hostname der Remote-Appliance.

`remote_setup_password:` ***Schnur***

(Optional) Das Passwort für das Einrichtungsbenutzerkonto im Ziel-EXA- oder ExtraHop-Packetstore. Dieser Parameter ist nicht erforderlich, wenn die Remote-Appliance ein Knoten in einem Explore-Cluster ist, der bereits mit der Konsole verbunden ist. Dieser Parameter ist nicht gültig, wenn es sich bei dem Remote-Gerät um einen Sensor handelt.

`remote_pairing_token:` ***Schnur***

(Optional) Das auf dem Zielsensor generierte Token. Sie müssen diesen Parameter angeben, um sich beim Zielsensor zu authentifizieren. Dieser Parameter ist nicht gültig, wenn Sie eine Verbindung zu einem EXA- oder ExtraHop-Paketstore herstellen.

`fingerprint:` ***Schnur***

(Optional) Der Fingerabdruck der Remote-Appliance. Wenn Sie eine Konsole mit einem EXA- oder ExtraHop-Paketspeicher verbinden, ist dieses Feld erforderlich. Andernfalls geben Sie 'insecure_skip_verification' an, um die Fingerabdrucküberprüfung zu umgehen. Beachten Sie, dass das Umgehen der Überprüfung Man-in-the-Middle-Angriffe ermöglichen kann.

`reset_configuration:` ***Boolesch***

(Optional) Gibt an, ob die Konfiguration der Remote-Appliance zurückgesetzt werden soll.

`remote_nickname_for_local:` ***Schnur***

(Optional) Der Spitzname für die Remote-Appliance, auf den die lokale Appliance verweist. Wenn Sie einen Sensor an ein anderes Gerät anschließen, ist dieses Feld erforderlich.

`local_nickname_for_remote:` ***Schnur***

(Optional) Der Spitzname für die lokale Appliance, auf den die Remote-Appliance verweist.

`remote_appliance_type:` ***Schnur***

Der Appliance-Typ für die neue Verbindung.

Die folgenden Werte sind gültig:

- `command`

- explore
- discover
- trace

`manages_local:` ***Boolesch***

(Optional) Gibt an, ob die Remote-Appliance die lokale Appliance verwaltet.

`managed_by_local:` ***Boolesch***

(Optional) Gibt an, ob die Remote-Appliance von der lokalen Appliance verwaltet wird. Wenn Sie eine Konsole mit einem Sensor verbinden, ist dieses Feld nicht erforderlich, da die Konsole immer die angeschlossenen Sensoren verwaltet.

`data_access:` ***Boolesch***

Gibt an, ob Daten zwischen der lokalen Appliance und der Remote-Appliance gemeinsam genutzt werden können.

`product_key:` ***Schnur***

(Optional) Der Produktschlüssel für die Remote-Appliance. Wenn dieser Parameter angegeben ist, wird die Remote-Appliance mit dem Product Key lizenziert. Dieser Parameter ist ungültig, wenn der Parameter remote_pairing_token angegeben ist.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "data_access": true,
    "fingerprint": "string",
    "host": "string",
    "local_nickname_for_remote": "string",
    "managed_by_local": true,
    "manages_local": true,
    "product_key": "string",
    "remote_appliance_type": "string",
    "remote_nickname_for_local": "string",
    "remote_pairing_token": "string",
    "remote_setup_password": "string",
    "reset_configuration": true
}
```

`DELETE /appliances/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Geben Sie den eindeutigen Bezeichner für die Remote-Appliance an.

Bewerbung

Anwendungen sind benutzerdefinierte Gruppen, die Metriken sammeln, die durch Trigger für verschiedene Arten von Traffic identifiziert wurden. Die Standardanwendung All Activity enthält alle gesammelten Metriken.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit der Anwendungsressource ausführen können:

Betrieb	Beschreibung
GET /anwendungen	Rufen Sie alle Anwendungen ab, die innerhalb eines bestimmten Zeitraums aktiv waren.
POST /Anwendungen	Erstellen Sie eine neue Anwendung.

Betrieb	Beschreibung
GET /Anwendungen/ {id}	Rufen Sie eine bestimmte Anwendung ab.
PATCH /Anwendungen/ {id}	Aktualisieren Sie eine bestimmte Anwendung.
GET /Anwendungen/ {id} /Aktivität	Ruft alle Aktivitäten für eine bestimmte Anwendung ab.
GET /applications/ {id} /alerts	Alles abrufen Warnungen die einer bestimmten Anwendung zugewiesen sind.
POST /Anwendungen/ {id} /Benachrichtigungen	Weisen Sie einer bestimmten Anwendung Warnmeldungen zu und heben Sie deren Zuweisung auf.
LÖSCHEN Sie /applications/ {id} /alerts/ {child-id}	Heben Sie die Zuweisung einer Alarm zu einer bestimmten Anwendung auf.
POST /Anwendungen/ {id} /alerts/ {Child-ID}	Weisen Sie einer bestimmten Anwendung eine Alarm zu.
GET /Anwendungen/ {id} /dashboards	Rufen Sie alle Dashboards ab, die sich auf eine bestimmte Anwendung beziehen.

Einzelheiten der Operation

`GET /applications/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
: Die eindeutige Kennung für die Anwendung.

`include_criteria:` ***Boolescher Wert***
: (Optional) Gibt an, ob die mit der Anwendung verknüpften Kriterien in die Antwort aufgenommen werden sollen.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "criteria": [],
    "description": "string",
    "discovery_id": "string",
    "display_name": "string",
    "extrahop_id": "string",
    "id": 0,
    "mod_time": 0,
    "node_id": 0,
    "user_mod_time": 0
}
```

`POST /applications`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
: Die Eigenschaften der Anwendung.

`node_id:` ***Zahl***

(Optional) Die eindeutige Kennung für den Sensor, dem diese Anwendung zugeordnet ist. Der Bezeichner kann über den Vorgang GET /appliances abgerufen werden. Dieses Feld ist nur auf einer Konsole gültig.

`discovery_id:` ***Schnur***

Die eindeutige Kennung für die Anwendung, die auf der Anwendungsseite im ExtraHop-System angezeigt wird.

`display_name:` ***Schnur***

Der benutzerfreundliche Name für die Anwendung.

`description:` ***Schnur***

(Optional) Eine optionale Beschreibung der Anwendung.

`criteria:` ***Reihe von Objekten***

(Optional) Eine Reihe von Protokoll- und Quellkriterien, die mit der Anwendung verknüpft sind. Der Inhalt dieses Arrays wird im Abschnitt „Kriterien“ unten definiert.

`protocol_default:` ***Schnur***

Die von der Anwendung überwachten Standardprotokolle. Unterstützte Werte sind „any“ und „none“.

`sources:` ***Reihe von Objekten***

Ein Array, das eine oder mehrere der Anwendung zugeordnete Metrik Quellen enthält. Die Anwendung sammelt nur Metriken aus den angegebenen Quellen. Der Inhalt dieses Arrays ist im Abschnitt „Quelle“ unten definiert.

`type:` ***Schnur***

Der Typ der Metrikquelle, die der Anwendung zugeordnet ist. Unterstützte Quelltypwerte sind 'Gerät' und 'device_group'.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät oder die Gerätegruppe, die der Anwendung zugeordnet ist.

`protocols:` ***Objekt***

(Optional) Die Liste mit einer oder mehreren Protokoll- und Rollenzuordnungen, die der Anwendung zugeordnet sind. Die Anwendung sammelt nur Metriken aus den angegebenen Protokollen. Das Format jedes Protokoll ist {'Protokoll': 'role'}. Beispiel: {'http': 'server'}. Unterstützte Rollenwerte sind „Client“, „Server“, „any“ oder „none“.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "criteria": {
        "protocol_default": "string",
        "sources": {
            "type": "string",
            "id": 0
        },
        "protocols": {}
    },
    "description": "string",
    "discovery_id": "string",
    "display_name": "string",
    "node_id": 0
}
```

`PATCH /applications/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die angegebenen Eigenschaftenupdates auf die Anwendung an.

`id:` ***Zahl***

Die eindeutige Kennung für die Anwendung.

`GET /applications`

Geben Sie die folgenden Parameter an.

`active_from:` ***Zahl***

(Optional) Gibt nur Anwendungen zurück, die nach der angegebenen Zeit aktiv sind. Positive Werte geben die Zeit in Millisekunden seit der Epoche an. Negative Werte geben die Zeit in Millisekunden vor der aktuellen Uhrzeit an.

`active_until:` ***Zahl***

(Optional) Gibt nur Anwendungen zurück, die vor dem angegebenen Zeitpunkt aktiv waren. Positive Werte geben die Zeit in Millisekunden seit der Epoche an. Negative Werte geben die Zeit in Millisekunden vor der aktuellen Uhrzeit an.

`limit:` ***Zahl***

(Optional) Beschränken Sie die Anzahl der Anwendungen, die zurückgegeben werden, auf die angegebene Höchstzahl.

`offset:` ***Zahl***

(Optional) Überspringen Sie die ersten n Anwendungsergebnisse. Dieser Parameter wird häufig mit dem Grenzparameter kombiniert.

`search_type:` ***Schnur***

Der Objekttyp, nach dem gesucht werden soll.

Die folgenden Werte sind gültig:

- `any`
- `name`
- `node`
- `discovery_id`
- `extrahop-id`

`value:` ***Schnur***

(Optional) Die Suchkriterien. Fügen Sie vor und nach den Kriterien einen Schrägstrich hinzu, um den RegEx-Abgleich anzuwenden.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "criteria": [],
    "description": "string",
    "discovery_id": "string",
    "display_name": "string",
    "extrahop_id": "string",
    "id": 0,
    "mod_time": 0,
    "node_id": 0,
    "user_mod_time": 0
}
```

`GET /applications/{id}/activity`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Anwendung.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "application_id": 0,
    "from_time": 0,
    "id": 0,
    "mod_time": 0,
    "stat_name": "string",
    "until_time": 0
}
```

`GET /applications/{id}/alerts`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Rufen Sie den eindeutigen Bezeichner für die Anwendung ab.

`direct_assignments_only:` ***Boolescher Wert***

(Optional) Gibt an, ob die Ergebnisse auf Warnungen beschränkt sind, die der Anwendung direkt zugewiesen sind.

`POST /applications/{id}/alerts`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Weist die angegebene Liste eindeutiger Kennungen für Warnmeldungen zu oder hebt deren Zuweisung auf.

`assign:` ***Reihe von Zahlen***

IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***

IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***

Geben Sie eine eindeutige Kennung für die Anwendung ein.

`POST /applications/{id}/alerts/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`id:` ***Zahl***

Die eindeutige Kennung für die Anwendung.

```
DELETE /applications/{id}/alerts/{child-id}
```

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`id:` ***Zahl***
Die eindeutige Kennung für die Anwendung.

```
GET /applications/{id}/dashboards
```

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Anwendung.

Audit-Protokoll

Das Audit-Log zeigt eine Datensatz aller aufgezeichneten Systemadministrations- und Konfigurationsaktivitäten an, z. B. die Uhrzeit der Aktivität, den Benutzer, der die Aktivität ausgeführt hat, den Vorgang, die Betriebsdetails und die Systemkomponente.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /auditlog	Ruft alle Audit-Log-Meldungen ab.

Einzelheiten der Operation

```
GET /auditlog
```

Geben Sie die folgenden Parameter an.

`limit:` ***Zahl***
(Optional) Die maximale Anzahl von Protokollnachrichten, die zurückgegeben werden sollen.

`offset:` ***Zahl***
(Optional) Die Anzahl der Protokollnachrichten, die in den Ergebnissen übersprungen werden sollen. Gibt Logmeldungen ab dem Offset-Wert zurück.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "body": {},
    "id": 0,
    "occur_time": 0,
    "time": 0
}
```

Auth

Sie können eine sichere SSO-Authentifizierung (Single Sign-On) für das ExtraHop-System über einen oder mehrere SAML-Identitätsanbieter (Security Assertion Markup Language) konfigurieren.

Wenn sich ein Benutzer bei einem ExtraHop-System anmeldet, das als Service Provider (SP) für die SAML-SSO-Authentifizierung konfiguriert ist, fordert das ExtraHop-System die Autorisierung vom entsprechenden

Identity Provider (IdP) an. Der Identitätsanbieter authentifiziert die Anmeldedaten des Benutzers und gibt dann die Autorisierung für den Benutzer an das ExtraHop-System zurück. Der Benutzer kann dann auf das ExtraHop-System zugreifen.

Betrieb	Beschreibung
GET /auth/identityproviders	Rufen Sie alle Identitätsanbieter ab.
POST /auth/identityproviders	Fügen Sie einen Identitätsanbieter für die Remoteauthentifizierung hinzu.
LÖSCHEN Sie /auth/identityproviders/ {id}	Löschen Sie einen bestimmten Identitätsanbieter.
GET /auth/identityproviders/ {id}	Rufen Sie einen bestimmten Identitätsanbieter ab.
PATCH /auth/identityproviders/ {id}	Aktualisieren Sie einen vorhandenen Identitätsanbieter.
GET /auth/identityproviders/ {id} /privileges	Rufen Sie die Berechtigungseinstellungen für einen bestimmten Identitätsanbieter ab.
PATCH /auth/identityproviders/ {id} /privileges	Aktualisieren Sie die Berechtigungseinstellungen für einen bestimmten Identitätsanbieter.
GET /auth/samlsp	Rufen Sie die Metadaten des SAML-Sicherheitsanbieters (SP) für dieses ExtraHop-System ab.

Einzelheiten der Operation

`POST /auth/identityproviders`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Parameter für den Identitätsanbieter.

`name:` ***Schnur***

Der Name des Identitätsanbieters.

`enabled:` ***Boolescher Wert***

Gibt an, ob die Authentifizierung über den Identity Provider auf dem ExtraHop-System aktiviert ist.

`entity_id:` ***Schnur***

(Optional) Die SAML 2.0-EntityID.

`sso_url:` ***Schnur***

(Optional) Die SAML 2.0-Single-Sign-On-URL (SSO).

`signing_certificate:` ***Schnur***

(Optional) Das SAML 2.0-X.509-Signaturzertifikat im PEM-Format.

`type:` ***Schnur***

Der Typ des Identitätsanbieters.

Die folgenden Werte sind gültig:

- `saml`

`auto_provision_users:` ***Boolescher Wert***

Gibt an, ob ein Benutzer über den Identity Provider auf dem ExtraHop-System erstellt werden kann.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "auto_provision_users": true,
    "enabled": true,
    "entity_id": "string",
    "name": "string",
    "signing_certificate": "string",
    "sso_url": "string",
    "type": "string"
}
```

`GET /auth/identityproviders`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "auto_provision_users": true,
    "enabled": true,
    "entity_id": "string",
    "id": 0,
    "name": "string",
    "signing_certificate": "string",
    "sso_url": "string",
    "type": "string"
}
```

`GET /auth/identityproviders/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für den Identitätsanbieter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "auto_provision_users": true,
    "enabled": true,
    "entity_id": "string",
    "id": 0,
    "name": "string",
    "signing_certificate": "string",
    "sso_url": "string",
    "type": "string"
}
```

`PATCH /auth/identityproviders/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für den Identitätsanbieter.

`body:` ***Objekt***

Die Parameter für den Identitätsanbieter.

`DELETE /auth/identityproviders/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Identitätsanbieter.

`GET /auth/identityproviders/{id}/privileges`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Identitätsanbieter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "detectionsaccesslevel": {},
    "ndrlevel": {},
    "npmlevel": {},
    "packetslevel": {},
    "writelevel": {}
}
```

`PATCH /auth/identityproviders/{id}/privileges`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Identitätsanbieter.

`body:` ***Objekt***
Ein Objekt, das die Berechtigungseinstellungen enthält.

`GET /auth/samlsp`

Geben Sie die folgenden Parameter an.

`xml:` ***Boolescher Wert***
(Optional) Gibt an, ob die SAML 2.0-XML-Metadaten abgerufen werden sollen.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "acs_url": "string",
    "entity_id": "string",
    "xml": "string"
}
```

Bündel

Bundles sind Dokumente im JSON-Format, die Informationen zur ausgewählten Systemkonfiguration enthalten, z. B. Trigger, Dashboards, Anwendungen oder Warnungen.

Sie können ein Paket erstellen und diese Konfigurationen dann auf ein anderes ExtraHop-System übertragen oder das Paket als Backup speichern. Bundles können auch heruntergeladen werden von ExtraHop Lösungspakete und über die REST-API angewendet. Weitere Informationen finden Sie unter Bündel.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
HOLEN SIE SICH /bundles	Rufen Sie Metadaten zu allen Bundles auf dem ExtraHop-System ab.
POST /Bundles	Laden Sie ein neues Paket in das ExtraHop-System hoch.
/bundles/ {id} LÖSCHEN	Löscht ein bestimmtes Paket.
ERHALTE /bundles/ {id}	Rufen Sie einen bestimmten Bundle-Export ab.
POST /bundles/ {id} /apply	Wenden Sie ein gespeichertes Paket auf das ExtraHop-System an.

Einzelheiten der Operation

`GET /bundles`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "built_in": true,
    "created_time": 0,
    "description": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string"
}
```

`POST /bundles`

Geben Sie die folgenden Parameter an.

`body:` ***Schnur***

Ein JSON-formatierter Bundle-Export.

`name:` ***Schnur***

Der freundliche Name für das Paket.

`description:` ***Schnur***

(Optional) Eine optionale Beschreibung für das Paket.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "name": "string"
}
```

`GET /bundles/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das Paket.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "built_in": true,
    "created_time": 0,
    "description": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string"
}
```

`DELETE /bundles/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Paket.

`POST /bundles/{id}/apply`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Paket.

`body:` ***Objekt***
Die Konfigurationsoptionen für die Anwendung des Paket.

`policy:` ***Schnur***
Gibt an, ob widersprüchliche Objekte überschrieben oder übersprungen werden sollen.

Die folgenden Werte sind gültig:

- `overwrite`
- `skip`

`include_assignments:` ***Boolescher Wert***
Gibt an, ob Objektzuweisungen mit dem Paket wiederhergestellt werden sollen.

`node_ids:` ***Reihe von Zahlen***
Eine Liste mit eindeutigen Kennungen für die Sensoren, auf die das Paket angewendet werden soll. Dieses Feld ist nur auf einer Konsole gültig.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "include_assignments": true,
    "node_ids": [],
    "policy": "string"
}
```

Wolke

Mit dieser Ressource können Sie Ihre lokalen Geräte verbinden Sensoren zu RevealX 360 Weitere Informationen finden Sie unter Stellen Sie über selbstverwaltete Sensoren eine Verbindung zu RevealX 360 her.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
POST /cloud/connect	Verbinden Sie das ExtraHop-System mit RevealX 360.

Einzelheiten der Operation

`POST /cloud/connect`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Das Token, das Sie mit RevealX 360 generiert haben.

`cloud_token:` ***Schnur***

Das Token, das Sie mit RevealX 360 generiert haben.

`nickname:` ***Schnur***

Ein Spitzname zur einfachen Identifizierung des Sensor.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "cloud_token": "string",
    "nickname": "string"
}
```

Benutzerdefiniertes Gerät

Sie können ein benutzerdefiniertes Gerät erstellen, indem Sie eine Reihe von Regeln definieren.

Sie können beispielsweise ein benutzerdefiniertes Gerät mit einer IP-Adresse in einem bestimmten VLAN erstellen. Standardmäßig werden alle IP-Adressen außerhalb der lokal überwachten Broadcast-Domänen hinter einem Router zusammengefasst. Um Geräte zu identifizieren, die sich hinter diesem Router befinden, können Sie ein benutzerdefiniertes Gerät erstellen und dann Messwerte von dem Gerät sammeln. Weitere Informationen finden Sie unter Erstellen Sie benutzerdefinierte Geräte über die REST-API.

Hinweis: Die benutzerdefinierte Geräteressource ist auf Konsolen nicht verfügbar.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /customdevices	Rufen Sie alle benutzerdefinierten Geräte ab.
POST /customdevices	Erstellen Sie ein benutzerdefiniertes Gerät.
/customdevices/ {id} LÖSCHEN	Löscht ein bestimmtes benutzerdefiniertes Gerät.
GET /customdevices/ {id}	Rufen Sie ein bestimmtes benutzerdefiniertes Gerät ab.
PATCH /customdevices/ {id}	Aktualisieren Sie ein bestimmtes benutzerdefiniertes Gerät.

Einzelheiten der Operation

`GET /customdevices`

Geben Sie die folgenden Parameter an.

`include_criteria:` ***Boolesch***
(Optional) Gibt an, ob die benutzerdefinierten Gerätekriterien enthalten sein sollen.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "author": "string",
    "criteria": [],
    "description": "string",
    "disabled": true,
    "extrahop_id": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string"
}
```

`GET /customdevices/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das benutzerdefinierte Gerät.

`include_criteria:` ***Boolesch***
(Optional) Gibt an, ob die benutzerdefinierten Gerätekriterien enthalten sein sollen.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "author": "string",
    "criteria": [],
    "description": "string",
    "disabled": true,
    "extrahop_id": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string"
}
```

`DELETE /customdevices/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das benutzerdefinierte Gerät.

`POST /customdevices`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Wenden Sie die angegebenen Eigenschaftswerte auf das neue benutzerdefinierte Gerät an.

`author:` ***Schnur***

Der Name des Erstellers des benutzerdefinierten Gerät.

`extrahop_id:` ***Schnur***

(Optional) Eine eindeutige Kennung für das benutzerdefinierte Gerät. Wenn dieses Feld nicht angegeben ist, wird eine ID aus dem benutzerdefinierten Gerätenamen generiert. Die ID darf keine Leerzeichen enthalten und kann nach dem Speichern des benutzerdefinierten Gerät nicht geändert werden.

`name:` ***Schnur***

Der benutzerfreundliche Name für das benutzerdefinierte Gerät.

`description:` ***Schnur***

(Optional) Eine optionale Beschreibung des benutzerdefinierten Gerät.

`disabled:` ***Boolesch***

Zeigt an, ob das benutzerdefinierte Gerät inaktiv ist.

`criteria:` ***Reihe von Objekten***

(Optional) Eine Reihe von benutzerdefinierten Gerätekriterien für dieses Gerät. Wenn dieses Feld mit der PATCH-Methode angegeben wird, werden alle zuvor angegebenen Kriterien gelöscht.

`ipaddr:` ***Schnur***

Die IP-Adresse, der das benutzerdefinierte Gerät zugeordnet werden soll.

`ipaddr_direction:` ***Schnur***

Die Verkehrsrichtung, der die iPaddr-Adresse zugeordnet werden soll. Die Kriterien bestimmen, welche Richtung des Datenverkehrs zur oder von der IPaddr-Adresse entspricht.

Die folgenden Werte sind gültig:

- `any`
- `dst`
- `src`

`ipaddr_peer:` ***Schnur***

Die IP-Adresse, mit der die Ipadder-Adresse kommuniziert, um das benutzerdefinierte Gerät zuzuordnen. Falls angegeben, begrenzt dieser Parameter den Datenverkehr, dem das benutzerdefinierte Gerät entspricht. Wenn ipaddr_direction beispielsweise „src“ ist, passt das benutzerdefinierte Gerät nur den Datenverkehr an die ipaddr_peer-Adresse von der ipaddr-Adresse an. Dieser Parameter ist nur gültig, wenn ipaddr angegeben ist und ipaddr_direction nicht „any“ ist.

`src_port_min:` ***Zahl***

Die untere Quellportgrenze, an die das benutzerdefinierte Gerät angepasst werden soll. Unterstützte Werte: 1-65535.

`src_port_max:` ***Zahl***

Die maximale Quellportgrenze, an die das benutzerdefinierte Gerät angepasst werden kann. Unterstützte Werte: 1-65535.

`dst_port_min:` ***Zahl***

Die untere Zielportgrenze, an die das benutzerdefinierte Gerät angepasst werden soll. Unterstützte Werte: 1-65535.

`dst_port_max:` ***Zahl***

Die maximale Zielportgrenze, an die das benutzerdefinierte Gerät angepasst werden kann. Unterstützte Werte: 1-65535.

`vlan_min:` ***Zahl***

Die untere VLAN-Grenze, an die das benutzerdefinierte Gerät angepasst werden soll.

`vlan_max:` ***Zahl***

Die maximale VLAN-Grenze, an die das benutzerdefinierte Gerät angepasst werden soll.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "author": "string",
    "criteria": {
        "ipaddr": "string",
        "ipaddr_direction": "string",
        "ipaddr_peer": "string",
        "src_port_min": 0,
        "src_port_max": 0,
        "dst_port_min": 0,
        "dst_port_max": 0,
        "vlan_min": 0,
        "vlan_max": 0
    },
    "description": "string",
    "disabled": true,
    "extrahop_id": "string",
    "name": "string"
}
```

`PATCH /customdevices/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wenden Sie die angegebenen Eigenschaftswertaktualisierungen auf das benutzerdefinierte Gerät an.

`id:` ***Zahl***

Die eindeutige Kennung für das benutzerdefinierte Gerät.

Personalisierung

Mit der Anpassungsressource können Sie Backup-Dateien auf dem ExtraHop-System verwalten. Sie müssen über System- und Zugriffsadministrationsrechte verfügen, um Operationen mit dieser Ressource ausführen zu können.

Sicherungsdateien enthalten sowohl Anpassungen als auch Systemressourcen. Anpassungen sind benutzerdefinierte Objekte wie Warnungen, Dashboards, Trigger und benutzerdefinierte Metriken. Systemressourcen sind Elemente wie Bundles, lokale Benutzer und Gruppen sowie das TLS-Zertifikat. Weitere Informationen finden Sie unter Einen Sensor oder eine Konsole sichern und wiederherstellen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /Anpassungen	Rufen Sie alle Sicherungsdateien ab.
POST /Anpassungen	Erstellen Sie eine Sicherungsdatei.
GET /anpassungen/status	Ruft Statusdetails für den letzten Sicherungsversuch ab.
LÖSCHE /customizations/ {id}	Löscht eine bestimmte Sicherungsdatei.
GET /customizations/ {id}	Ruft eine bestimmte Sicherungsdatei ab.

Bedienung	Beschreibung
POST /anpassungen/ {id} /apply	Stellen Sie nur Anpassungen aus einer bestimmten Sicherungsdatei wieder her.
POST /anpassungen/ {id} /herunterladen	Laden Sie eine bestimmte Sicherungsdatei herunter.

Einzelheiten der Operation

`GET /customizations`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "auto": true,
    "create_time": 0,
    "id": 0,
    "name": "string",
    "recovered": true
}
```

`POST /customizations`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Ein eindeutiger Name für die Backup-Datei.

`name:` ***Schnur***

Ein eindeutiger Name für die Backup-Datei.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "name": "string"
}
```

`GET /customizations/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Sicherungsdatei.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "auto": true,
    "create_time": 0,
    "id": 0,
    "name": "string",
    "recovered": true
}
```

`DELETE /customizations/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Sicherungsdatei.

`POST /customizations/{id}/apply`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Sicherungsdatei.

`POST /customizations/{id}/download`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Sicherungsdatei.

`GET /customizations/status`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "did_last_succeed": true,
    "last_attempt_time": 0,
    "last_success_time": 0
}
```

Armaturenbretter

Dashboards sind integrierte oder benutzerdefinierte Ansichten Ihrer ExtraHop-Metrikinformationen. Weitere Informationen finden Sie unter Dashboards.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
HOLEN SIE SICH /dashboards	Rufen Sie alle Dashboards ab.
/dashboards/ {id} LÖSCHEN	Löschen Sie ein bestimmtes Dashboard.
GET /dashboards/ {id}	Rufen Sie ein bestimmtes Dashboard ab.
PATCH /dashboards/ {id}	Aktualisieren Sie den Besitz eines bestimmten Dashboard.
GET /dashboards/ {id} /reports	Rufen Sie Dashboard-Berichte ab, die ein bestimmtes Dashboard enthalten. **Hinweis:** Dieser Vorgang ist nur von einer Konsole aus verfügbar.
GET /dashboards/ {id} /sharing	Rufen Sie die Benutzer und ihre Freigabeberechtigungen für ein bestimmtes Dashboard ab.
PATCH /dashboards/ {id} /sharing	Aktualisieren Sie die Benutzer und ihre Freigabeberechtigungen für ein bestimmtes Dashboard.

Betrieb	Beschreibung
PUT /dashboards/ {id} /sharing	Ersetzen Sie die Benutzer und ihre Freigabeberechtigungen für ein bestimmtes Dashboard.

Einzelheiten der Operation

`GET /dashboards`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "author": "string",
    "comment": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "owner": "string",
    "rights": [
        "string"
    ],
    "short_code": "string",
    "type": "string"
}
```

`GET /dashboards/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Dashboard.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "author": "string",
    "comment": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "owner": "string",
    "rights": [
        "string"
    ],
    "short_code": "string",
    "type": "string"
}
```

`DELETE /dashboards/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Dashboard.

`PATCH /dashboards/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Der Benutzername des Dashboard-Besitzers.
`id:` ***Zahl***
Die eindeutige Kennung für das Dashboard.

`GET /dashboards/{id}/sharing`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Dashboard.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "anyone": "string",
    "groups": {},
    "users": {}
}
```

`PUT /dashboards/{id}/sharing`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Benutzer und ihre Berechtigungsstufen.
`id:` ***Zahl***
Die eindeutige Kennung für das Dashboard.

`PATCH /dashboards/{id}/sharing`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Benutzer und ihre Berechtigungsstufen.
`id:` ***Zahl***
Die eindeutige Kennung für das Dashboard.

`GET /dashboards/{id}/reports`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Dashboard.

Gerät

Geräte sind Objekte in Ihrem Netzwerk, die von Ihrem ExtraHop-System identifiziert und klassifiziert wurden. Weitere Informationen finden Sie unter Geräte.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /Geräte	Rufen Sie alle Geräte ab, die innerhalb eines bestimmten Zeitraums aktiv waren. Weitere

Betrieb	Beschreibung
	Informationen finden Sie unter Extrahieren Sie die Geräteliste über die REST-API. **Hinweis:** Ein Gerät gilt als inaktiv, wenn fünf Minuten lang keine Pakete gesendet oder empfangen wurden. Wenn ein Gerät jedoch nach einem Zeitraum der Inaktivität von weniger als fünf Tagen wieder Pakete sendet oder empfängt, wird davon ausgegangen, dass das Gerät kontinuierlich aktiv war, auch während des Zeitraums der Inaktivität.
POST /Geräte/Suche	Rufen Sie alle Geräte ab, die bestimmten Kriterien entsprechen. Weitere Informationen finden Sie unter Suchen Sie über die REST-API nach einem Gerät. **Hinweis:** Ein Gerät gilt als inaktiv, wenn fünf Minuten lang keine Pakete gesendet oder empfangen wurden. Wenn ein Gerät jedoch nach einem Zeitraum der Inaktivität von weniger als fünf Tagen wieder Pakete sendet oder empfängt, wird davon ausgegangen, dass das Gerät kontinuierlich aktiv war, auch während des Zeitraums der Inaktivität.
GET /devices/ {id}	Rufen Sie ein bestimmtes Gerät ab.
PATCH /Geräte/ {id}	Aktualisieren Sie ein bestimmtes Gerät.
GET /devices/ {id} /activity	Ruft alle Aktivitäten für ein Gerät ab.
GET /devices/ {id} /alerts	Alles abrufen Warnungen die einem bestimmten Gerät zugewiesen sind.
POST /devices/ {id} /alerts	Weisen Sie Warnmeldungen ein bestimmtes Gerät zu und heben Sie die Zuweisung auf.
LÖSCHEN Sie /devices/ {id} /alerts/ {child-id}	Heben Sie die Zuweisung einer Alarm zu einem bestimmten Gerät auf.
POST /devices/ {id} /alerts/ {child-id}	Weisen Sie einem bestimmten Gerät eine Alarm zu.
GET /devices/ {id} /dashboards	Rufen Sie alle Dashboards ab, die sich auf ein bestimmtes Gerät beziehen.
GET /devices/ {id} /devicegroups	Alles abrufen Gerätegruppen die einem bestimmten Gerät zugewiesen sind.
POST /devices/ {id} /devicegroups	Weisen Sie Gerätegruppen ein bestimmtes Gerät zu und heben Sie die Zuweisung auf.
LÖSCHEN Sie /devices/ {id} /devicegroups/ {child-id}	Heben Sie die Zuweisung einer Gerätegruppe zu einem bestimmten Gerät auf.
POST /devices/ {id} /devicegroups/ {child-id}	Weisen Sie einem bestimmten Gerät eine Gerätegruppe zu.

Betrieb	Beschreibung
GET /devices/ {id} /dnsnames	Ruft alle DNS-Namen ab, die einem bestimmten Gerät zugeordnet sind.
GET /devices/ {id} /ipaddrs	Ruft alle IP-Adressen ab, die innerhalb eines bestimmten Zeitraums mit einem bestimmten Gerät verknüpft waren.
GET /devices/ {id} /software	Ruft eine Liste der Software ab, die auf dem angegebenen Gerät ausgeführt wird.
GET /devices/ {id} /tags	Ruft alle Tags ab, die einem bestimmten Gerät zugewiesen sind.
POST /Geräte/ {id} /tags	Weisen Sie Tags ein bestimmtes Gerät zu und heben Sie die Zuweisung auf.
LÖSCHEN Sie /devices/ {id} /tags/ {child-id}	Heben Sie die Zuweisung eines Tags zu einem bestimmten Gerät auf.
POST /Geräte/ {id} /tags/ {Kinder-ID}	Weisen Sie einem bestimmten Gerät ein Tag zu.
GET /devices/ {id} /triggers	Ruft alle Trigger ab, die einem bestimmten Gerät zugewiesen sind.
POST /Geräte/ {id} /Trigger	Weisen Sie Triggern ein bestimmtes Gerät zu und heben Sie die Zuweisung auf.
LÖSCHEN /devices/ {id} /triggers/ {child-id}	Heben Sie die Zuweisung eines Auslöser zu einem bestimmten Gerät auf.
POST /Geräte/ {id} /triggers/ {Kinder-ID}	Weisen Sie einem bestimmten Gerät einen Auslöser zu.

Einzelheiten der Operation

`GET /devices`

Geben Sie die folgenden Parameter an.

`active_from:` ***Zahl***

(Optional) Der Anfangszeitstempel für die Anfrage. Gibt nur Geräte zurück, die nach dieser Zeit aktiv sind. Die Zeit wird in Millisekunden seit der Epoche ausgedrückt. 0 gibt den Zeitpunkt der Anfrage an. Ein negativer Wert wird relativ zur aktuellen Uhrzeit ausgewertet. Die Standardeinheit für einen negativen Wert ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`active_until:` ***Zahl***

(Optional) Der Endzeitstempel für die Anfrage. Nur Gerät zurückgeben, die vor diesem Zeitpunkt aktiv waren. Folgt den gleichen Zeitwertrichtlinien wie der active_from Parameter.

`limit:` ***Zahl***

(Optional) Beschränken Sie die Anzahl der zurückgegebenen Geräte auf die angegebene Höchstzahl.

`offset:` ***Zahl***

(Optional) Überspringen Sie die ersten n Geräteergebnisse. Dieser Parameter wird häufig mit dem Grenzwertparameter kombiniert.

`search_type:` ***Schnur***

Gibt das zu durchsuchende Feld an.

Die folgenden Werte sind gültig:

- any
- name
- discovery_id
- ip address
- mac address
- vendor
- type
- tag
- activity
- node
- vlan
- discover time

`value:` ***Schnur***

(Optional) Gibt die Suchkriterien an.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "activity": [],
    "analysis": "string",
    "analysis_level": 0,
    "auto_role": "string",
    "cdp_name": "string",
    "cloud_account": "string",
    "cloud_instance_description": "string",
    "cloud_instance_id": "string",
    "cloud_instance_name": "string",
    "cloud_instance_type": "string",
    "critical": true,
    "custom_criticality": "string",
    "custom_make": "string",
    "custom_model": "string",
    "custom_name": "string",
    "custom_type": "string",
    "default_name": "string",
    "description": "string",
    "device_class": "string",
    "dhcp_name": "string",
    "discover_time": 0,
    "discovery_id": "string",
    "display_name": "string",
    "dns_name": "string",
    "extrahop_id": "string",
    "id": 0,
    "ipaddr4": "string",
    "ipaddr6": "string",
    "is_l3": true,
    "last_seen_time": 0,
    "macaddr": "string",
    "mod_time": 0,
    "model": "string",
    "model_override": "string",
    "netbios_name": "string",
    "node_id": 0,
    "on_watchlist": true,
    "parent_id": 0,
    "role": "string",
    "subnet_id": "string",
    "user_mod_time": 0,
    "vendor": "string",
```

```
    "vlanid": 0,
    "vpc_id": "string"
}
```

`POST /devices/search`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Gerätekriterien.

`active_from:` ***Zahl***

(Optional) Der Anfangszeitstempel für die Anfrage. Gibt nur Geräte zurück, die nach dieser Zeit aktiv sind. Die Zeit wird in Millisekunden seit der Epoche ausgedrückt. 0 gibt den Zeitpunkt der Anfrage an. Ein negativer Wert wird relativ zur aktuellen Uhrzeit ausgewertet. Die Standardeinheit für einen negativen Wert ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`active_until:` ***Zahl***

(Optional) Der Endzeitstempel für die Anfrage. Gibt nur Geräte zurück, die vor diesem Zeitpunkt aktiv waren. Folgt den gleichen Zeitwertrichtlinien wie der active_from Parameter.

`limit:` ***Zahl***

(Optional) Beschränken Sie die Anzahl der zurückgegebenen Geräte auf die angegebene Höchstzahl.

`offset:` ***Zahl***

(Optional) Überspringen Sie die angegebene Anzahl von Geräten. Dieser Parameter wird häufig mit dem Grenzwertparameter kombiniert, um Ergebnismengen zu paginieren.

`filter:` ***Objekt***

(Optional) Geben Sie die Filterkriterien für Suchergebnisse an.

`field:` ***Schnur***

Der Name des Feldes, nach dem die Ergebnisse gefiltert werden sollen. Die Suche vergleicht den Inhalt des Feldparameters mit dem Wert des Operandenparameters.

Die folgenden Werte sind gültig:

- `name`
- `discovery_id`
- `ipaddr`
- `macaddr`
- `vendor`
- `tag`
- `activity`
- `node`
- `vlan`
- `discover_time`
- `role`
- `dns_name`
- `dhcp_name`
- `netbios_name`
- `cdp_name`
- `custom_name`
- `software`
- `model`

- `is_critical`
- `instance_id`
- `instance_name`
- `instance_type`
- `cloud_account`
- `vpc_id`
- `subnet_id`
- `is_active`
- `analysis`
- `network_locality_type`
- `network_locality_id`
- `id`

`operator:` ***Schnur***

Die Vergleichsmethode, die angewendet wird, wenn der Operandenwert mit dem Feldinhalt verglichen wird. Alle Filterobjekte benötigen einen Operator.

Die folgenden Werte sind gültig:

- `>`
- `<`
- `<=`
- `>=`
- `=`
- `!=`
- `startswith`
- `and`
- `or`
- `not`
- `exists`
- `not_exists`
- `~`
- `!~`
- `in`
- `not_in`

`operand:` ***Zeichenfolge oder Zahl oder Objekt oder Array***

Der Wert, den die Abfrage abzugleichen versucht. Die Abfrage vergleicht den Wert des Operanden mit dem Inhalt des Feldparameters und wendet die durch den Operatorparameter angegebene Vergleichsmethode an. Sie können den Operanden als Zeichenfolge, Ganzzahl oder Objekt angeben. Informationen zu Objektwerten finden Sie in der REST-API-Leitfaden.

`rules:` ***Reihe von Objekten***

Ein Array aus einem oder mehreren Filterobjekten, die rekursiv eingebettet werden können. Für diesen Parameter sind nur die Operatoren „und“, „oder“ oder „nicht“ zulässig.

`result_fields:` ***Reihe von Zeichenketten***

(Optional) Gibt die angegebenen Felder und die Geräte-ID zurück. Wenn diese Option nicht angegeben ist, werden alle Felder zurückgegeben.

Die folgenden Werte sind gültig:

- `mod_time`
- `node_id`
- `id`

- extrahop_id
- discovery_id
- display_name
- description
- user_mod_time
- discover_time
- vlanid
- parent_id
- macaddr
- vendor
- is_l3
- ipaddr4
- ipaddr6
- device_class
- default_name
- custom_name
- cdp_name
- dhcp_name
- netbios_name
- dns_name
- custom_type
- auto_role
- analysis_level
- analysis
- role
- on_watchlist
- last_seen_time
- activity
- model
- model_override
- custom_make
- custom_model
- critical
- custom_criticality
- cloud_instance_id
- cloud_instance_type
- cloud_instance_description
- cloud_instance_name
- cloud_account
- vpc_id
- subnet_id

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "active_from": 0,
    "active_until": 0,
    "filter": {
        "field": "string",
        "operator": "string",
        "operand": "string",
        "rules": []
    },
```

```
    "limit": 0,
    "offset": 0,
    "result_fields": []
}
```

`GET /devices/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "activity": [],
    "analysis": "string",
    "analysis_level": 0,
    "auto_role": "string",
    "cdp_name": "string",
    "cloud_account": "string",
    "cloud_instance_description": "string",
    "cloud_instance_id": "string",
    "cloud_instance_name": "string",
    "cloud_instance_type": "string",
    "critical": true,
    "custom_criticality": "string",
    "custom_make": "string",
    "custom_model": "string",
    "custom_name": "string",
    "custom_type": "string",
    "default_name": "string",
    "description": "string",
    "device_class": "string",
    "dhcp_name": "string",
    "discover_time": 0,
    "discovery_id": "string",
    "display_name": "string",
    "dns_name": "string",
    "extrahop_id": "string",
    "id": 0,
    "ipaddr4": "string",
    "ipaddr6": "string",
    "is_l3": true,
    "last_seen_time": 0,
    "macaddr": "string",
    "mod_time": 0,
    "model": "string",
    "model_override": "string",
    "netbios_name": "string",
    "node_id": 0,
    "on_watchlist": true,
    "parent_id": 0,
    "role": "string",
    "subnet_id": "string",
    "user_mod_time": 0,
    "vendor": "string",
    "vlanid": 0,
    "vpc_id": "string"
}
```

`PATCH /devices/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Wenden Sie die angegebenen Eigenschaftswertaktualisierungen auf das Gerät an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`GET /devices/{id}/activity`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "device_id": 0,
    "from_time": 0,
    "id": 0,
    "mod_time": 0,
    "stat_name": "string",
    "until_time": 0
}
```

`GET /devices/{id}/ipaddrs`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`from:` ***Zahl***
(Optional) Ruft IP-Adressen ab, die dem Gerät nach dem angegebenen Datum zugeordnet wurden, ausgedrückt in Millisekunden seit der Epoche.

`until:` ***Zahl***
(Optional) Ruft IP-Adressen ab, die dem Gerät vor dem angegebenen Datum zugeordnet wurden, ausgedrückt in Millisekunden seit der Epoche.

`GET /devices/{id}/dnsnames`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`from:` ***Zahl***
(Optional) Ruft DNS-Namen ab, die dem Gerät nach dem angegebenen Datum zugeordnet wurden, ausgedrückt in Millisekunden seit der Epoche.

`until:` ***Zahl***
(Optional) Ruft DNS-Namen ab, die dem Gerät vor dem angegebenen Datum zugeordnet wurden, ausgedrückt in Millisekunden seit der Epoche.

`GET /devices/{id}/triggers`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`direct_assignments_only:` ***Boolesch***
(Optional) Beschränken Sie die Ergebnisse auf Trigger, die dem Gerät direkt zugewiesen sind.

`POST /devices/{id}/triggers`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Eine Liste eindeutiger Identifikatoren für Trigger, die dem Gerät zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`POST /devices/{id}/triggers/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`DELETE /devices/{id}/triggers/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`GET /devices/{id}/dashboards`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`GET /devices/{id}/devicegroups`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät.

`active_from:` ***Zahl***

(Optional) Der Anfangszeitstempel für die Anfrage. Gibt nur dynamische Gerätegruppen zurück, zu denen das Gerät nach dieser Zeit gehörte. Die Zeit wird in Millisekunden seit der Epoche ausgedrückt. 0 gibt den Zeitpunkt der Anfrage an. Ein negativer Wert wird relativ zur aktuellen Uhrzeit ausgewertet. Die Standardeinheit für einen negativen Wert ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`active_until:` ***Zahl***

(Optional) Der Endzeitstempel für die Anfrage. Gibt nur dynamische Gerätegruppen zurück, zu denen das Gerät vor diesem Zeitpunkt gehörte. Folgt den gleichen Zeitwertrichtlinien wie der active_from Parameter.

`POST /devices/{id}/devicegroups`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Liste der eindeutigen Identifikatoren für Gerätegruppen, die dem Gerät zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***

IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***

IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`POST /devices/{id}/devicegroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`DELETE /devices/{id}/devicegroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`GET /devices/{id}/tags`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`POST /devices/{id}/tags`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Eine Liste eindeutiger Identifikatoren für Tags, die dem Gerät zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`POST /devices/{id}/tags/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Tag.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`DELETE /devices/{id}/tags/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Tag.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`GET /devices/{id}/alerts`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`direct_assignments_only:` ***Boolesch***

(Optional) Beschränken Sie die Ergebnisse auf Warnungen, die dem Gerät direkt zugewiesen sind.

`POST /devices/{id}/alerts`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Liste der eindeutigen Identifikatoren für Warnmeldungen, die dem Gerät zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***

IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***

IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`POST /devices/{id}/alerts/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`DELETE /devices/{id}/alerts/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für die Alarm.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`GET /devices/{id}/software`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das Gerät, die als API-ID auf der Geräteseite im ExtraHop-System angezeigt wird.

`from:` ***Zahl***

(Optional) Gibt Software zurück, die nach dem angegebenen Datum auf dem Gerät beobachtet wurde, ausgedrückt in Millisekunden seit der Epoche.

`until:` ***Zahl***

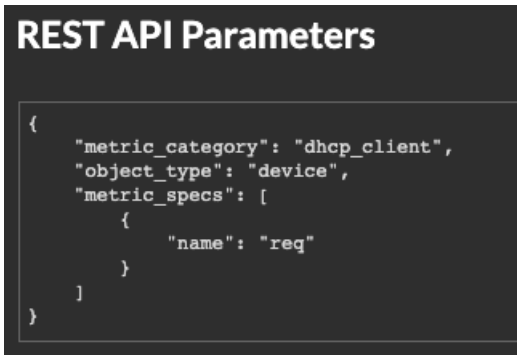
(Optional) Gibt Software zurück, die vor dem angegebenen Datum auf dem Gerät beobachtet wurde, ausgedrückt in Millisekunden seit der Epoche.

Operandenwerte für die Gerätesuche

Mit dem POST /devices/search-Vorgang können Sie anhand der in Filterobjekten angegebenen Kriterien nach Geräten suchen. Jedes Objekt sollte einen eindeutigen Wert für den enthalten `operand` Feld, das für das angegebene Feld gültig ist `field` Wert.

`activity`

Um nach Metrik Aktivität zu suchen, geben Sie den `field` Wert als `activity` und die `operand` Wert als `metric_category`. Du kannst finden `metric_category` Werte im Abschnitt REST-API-Parameter des Metrikkatalogs.



Das folgende Beispiel gibt Ergebnisse für Geräte zurück, die allen für einen DHCP-Client klassifizierten Metrikaktivitäten entsprechen, z. B. der Anzahl der gesendeten DHCP-Anfragen .

```
{
    "filter":  {
        "field": "activity",
        "operand": "dhcp_client",
        "operator": "="
    }
}
```

Hinweis: Rufen Sie programmgesteuert eine Liste aller Metrik Aktivitäten für ein Gerät über die `GET /devices/{id}/activity` Betrieb. Das `stat_name` Wert entspricht dem `metric_category` Wert in der `metric_catalog`, nach dem letzten Punkt.

In der folgenden Beispielantwort ist die `stat_name` Wert ist `extrahop.device.dhcp_client`. Entfernen Sie den Text vor dem letzten Punkt, um den zu identifizieren `metric_catalog` Wert von `dhcp_client`.

```
{
    "id": 198606,
    "from_time": 1581537120000,
    "until_time": 1581542520000,
    "mod_time": 1581542533963,
    "device_id": 30096,
    "stat_name": "extrahop.device.dhcp_client"
}
```

Analyse

Um nach Geräteanalyseebene zu suchen, geben Sie den `field` Wert als `analysis` und die `operand` Wert als eine der folgenden Zeichenketten:

Standard
Geräte in Standardanalyse.

fortgeschrittene
Geräte in Erweiterte Analyse.

Entdeckung
Geräte in Entdeckungsmodus.

l2_exempt
Geräte in L2 Parent Analysis.

flow_log
Geräte in der Strömungsanalyse.

`discover_time`

Um nach einem Zeitraum zu suchen, geben Sie den `field` Wert als `discover_time` und ein `operand` Wert mit `from` und `until` Parameter, wobei es sich bei den Werten um Daten handelt, ausgedrückt in Millisekunden seit der Epoche.

Das folgende Beispiel gibt Ergebnisse für alle Geräteaktivitäten zurück, die am 21. August 2019 zwischen 13:00 Uhr und 15:00 Uhr stattfanden.

```
{
    "filter":  {
        "field": "discover_time",
        "operand": {
            "from": "1566392400000",
            "until": "1566399600000"
        },
        "operator": "="
    }
}
```

`discovery_id`

Um nach der eindeutigen ID für das Gerät zu suchen, geben Sie die `field` Wert als `discovery_id` und die `operand` Wert als Discovery-ID.

```
{
  "filter": {
    "field": "discovery_id",
```

```
    "operand": "c12vf90qpg290000",
    "operator": "="
  }
}
```

id

Um mehrere Geräte abzurufen, geben Sie den Feldwert als an `id`, das `operator` Wert als `in`, und die `operand` Wert als Array von IDs.

```
{
  "filter": {
    "field": "id",
    "operand": [5388,5387],
    "operator": "in"
  }
}
```

Um Geräte aus den Suchergebnissen auszuschließen, geben Sie einen Filter mit mehreren Regeln an und geben Sie eine Regel mit dem Feldwert als an `id`, das `operator` Wert als `not_in`, und die `operand` Wert als Array von IDs.

```
{
  "filter": {
    "operator": "and",
    "rules": [
      {
        "field": "id",
        "operand": [5388,5387],
        "operator": "not_in"
      },
      {
        "field": "discover_time",
        "operand": {
          "from": "1692984750000",
          "until": "1693416750000"
        },
        "operator": "="
      }
    ]
  }
}
```

ist_aktiv

Um nach Geräten zu suchen, die in den letzten 30 Minuten aktiv waren, geben Sie den Feldwert als an `is_active` und die `operand` Wert als boolescher Wert.

```
{
  "filter": {
    "field": "is_active",
    "operand": true,
    "operator": "="
  }
}
```

`ipaddr`

Um nach der IP-Adresse zu suchen, geben Sie den `field` Wert als `ipaddr` und die `operand` Wert als IP-Adresse oder CIDR-Block.

```
{
  "filter": {
    "field": "ipaddr",
    "operand": "192.168.12.0/28",
    "operator": "="
  }
}
```

`node`

Um nach der eindeutigen ID eines zu suchen Sensor, spezifizieren Sie die `field` Wert als `node` und die `operand` Wert als Sensor UUID.

```
{
  "filter": {
    "field": "node",
    "operand": "qqvsplfa-zxsk-32l0-19g1-076vfr42pw31",
    "operator": "="
  }
}
```

`macaddr`

Um nach der MAC-Adresse eines Gerät zu suchen, geben Sie den Feldwert als an `macaddr` und der Operandenwert als MAC-Adresse des Gerät. Das folgende Beispiel gibt Ergebnisse für Geräte mit einer MAC-Adresse von `C1:1C:N2:0Q:PJ:10` oder `C1:1C:N2:0Q:PJ:11`.

```
{
  "filter": {
    "operator": "or",
    "rules": [
      {
        "field": "macaddr",
        "operand": "C1:1C:N2:0Q:PJ:10",
        "operator": "="
      },
      {
        "field": "macaddr",
        "operand": "C1:1C:N2:0Q:PJ:11",
        "operator": "="
      }
    ]
  }
}
```

`model`

Um nach dem Gerätemodell zu suchen, geben Sie den `field` Wert als `model`. Wenn der Betreiber `=`, `!=`, `exists`, oder `not_exists`, geben Sie den Operanden als Modell-ID an, die Sie in der `model` Feld von `POST /device/search` Antworten.

```
{
  "filter": {
    "field": "model",
    "operand": "apple_ipad_pro_12_9_inch_wifi_cellular_5th_gen",
```

```
    "operator": "="
  }
}
```

Wenn der Betreiber ~ oder !~, geben Sie den Operanden als Namen der Marke und des Modells an, die Sie bei der Suche nach einem Gerät im ExtraHop-System einsehen können.

```
{
  "filter": {
    "field": "model",
    "operand": "Apple iPad Pro",
    "operator": "~"
  }
}
```

`name`

Um nach dem Anzeigenamen des Gerät zu suchen, geben Sie den `field` Wert als Name und der `operand` Wert als Gerätename oder als Regex-Zeichenfolge.

```
{
  "filter": {
    "field": "name",
    "operand": "VMware B2CEB6",
    "operator": "="
  }
}
```

Netzwerk_Lokalitäts-ID

Um nach Netzwerklokalität zu suchen, geben Sie den `field` Wert als `network_locality_id` und der Operandenwert als Netzwerklokalitäts-ID.

```
{
  "filter": {
    "field": "network_locality_id",
    "operand": 123,
    "operator": "="
  }
}
```

`role`

Um nach der Geräterolle zu suchen, geben Sie die `field` Wert als `role` und die `operand` Wert als Geräterolle.

```
{
  "filter": {
    "field": "role",
    "operand": "voip_phone",
    "operator": "="
  }
}
```

`software`

Um nach der auf dem Gerät ausgeführten Software zu suchen, geben Sie die `field` Wert als `software` und die `operand` Wert als die ID, die dieser Software auf dem ExtraHop-System zugeordnet ist.

```
{
  "filter": {
    "field": "software",
    "operand": "windows_10",
    "operator": "="
  }
}
```

Hinweis: Rufen Sie programmgesteuert eine Liste aller Software-IDs ab, die einem Gerät zugeordnet sind, über den `GET /devices/{id}/software` Betrieb.

In der folgenden Beispielantwort ist die `id` Wert für die Software ist `windows_10`.

```
[
  {
    "software_type": "OS",
    "name": "Windows",
    "version": "10",
    "description": null,
    "id": "windows_10"
  }
]
```

`software_type`

Um nach der Art der auf dem Gerät ausgeführten Software zu suchen, geben Sie die `field` Wert als `software_type` und die `operand` Wert als Softwaretyp-ID.

```
{
  "filter": {
    "field": "software_type",
    "operand": "OS",
    "operator": "="
  }
}
```

Hinweis: Rufen Sie programmgesteuert eine Liste aller Softwaretyp-IDs ab, die einem Gerät zugeordnet sind, über die `GET /devices/{id}/software` Betrieb.

In der folgenden Beispielantwort lautet der ID-Wert für den Softwaretyp `OS`.

```
[
  {
    "software_type": "OS",
    "name": "Windows",
    "version": "10",
    "description": null,
    "id": "windows_10"
  }
]
```

`tag`

Um nach einem Geräte-Tag zu suchen, geben Sie das `field` Wert als `tag` und die `operand` Wert als Tag-Name oder als Regex-Zeichenfolge.

```
{
  "filter": {
    "field": "tag",
    "operand": "Custom Tag",
    "operator": "="
  }
}
```

Hinweis: Rufen Sie programmgesteuert eine Liste aller Geräte-Tags über die `GET /devices/{id}/tags` Betrieb.

In der folgenden Beispielantwort ist die `name` Der Wert für das Tag ist `Custom Tag`.

```
[
  {
    "mod_time": 1521577040934,
    "id": 19,
    "name": "Custom Tag"
  }
]
```

vlan

Um nach der ID eines VLANs zu suchen, geben Sie den `field` Wert als `vlan` und die `operand` Wert als ID des VLAN.

```
{
  "filter": {
    "field": "vlan",
    "operand": "0",
    "operator": "="
  }
}
```

Suche mit regulären Ausdrücken (Regex)

Mit Sicherheit `field` Werte, die Zeichenfolge kann in Regex-Syntax vorliegen. Spezifizieren Sie die `operand` Wert als Objekt, das einen `value` Parameter mit der Regex-Syntax, die Sie abgleichen möchten, und einem `is_regex` Parameter, der auf gesetzt ist `true`. Das folgende Beispiel gibt Ergebnisse für alle DNS-Namen zurück, die mit enden `com`.

```
{
    "filter":  {
        "field": "dns_name",
        "operand": {
            "value": ".*?com",
            "is_regex": true
        },
        "operator": "="
    }
}
```

Ein `operand` Feld mit Regex-Syntax ist gültig für Folgendes `field` Werte:

- CDP_Name
- benutzerdefinierter_Name

- DNS-Name
- dhcp_name
- Modell
- Name
- netbios_name
- Software
- Tag
- Lieferant

Unterstützte Zeiteinheiten

Für die meisten Parameter ist die Standardeinheit für die Zeitmessung Millisekunden. Die folgenden Parameter geben jedoch alternative Zeiteinheiten wie Minuten und Stunden zurück oder akzeptieren diese:

- Gerät
  - aktive_von
  - aktiv_bis
- Gerätegruppe
  - aktive_von
  - aktiv_bis
- Metriken
  - von
  - bis
- Protokoll aufzeichnen
  - von
  - bis
  - kontext_ttl

Die folgende Tabelle zeigt die unterstützten Zeiteinheiten:

Zeiteinheit	Einheitensuffix
Jahr	`y`
Monat	`M`
Woche	`w`
Tag	`d`
Stunde	`h`
Minute	`m`
Zweiter	`s`
Millisekunde	`ms`

Um für einen Parameter eine andere Zeiteinheit als Millisekunden anzugeben, hängen Sie das Einheitensuffix an den Wert an. Um beispielsweise Geräte anzufordern, die in den letzten 30 Minuten aktiv waren, geben Sie den folgenden Parameterwert an:

```
GET /api/v1/devices?active_from=-30m
```

Das folgende Beispiel spezifiziert eine Suche nach HTTP Datensätze, die vor 1 bis 2 Stunden erstellt wurden:

```
{
    "from": "-2h",
    "until": "-1h",
    "types": ["~http"]
}
```

Gerätegruppe

Gerätegruppen kann entweder statisch oder dynamisch sein.

Eine statische Gerätegruppe ist benutzerdefiniert. Sie erstellen eine Gerätegruppe und identifizieren dann jedes Gerät manuell und weisen es dieser Gruppe zu. Eine dynamische Gerätegruppe wird durch eine Reihe von konfigurierten Regeln definiert und automatisch verwaltet.

Sie können beispielsweise eine Gerätegruppe erstellen und dann eine Regel festlegen, um alle Geräte innerhalb eines bestimmten IP-Adressbereichs zu klassifizieren, sodass sie dieser Gruppe automatisch hinzugefügt werden. Weitere Informationen finden Sie unter Gerätegruppen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /devicegroups	Ruft alle Gerätegruppen ab, die innerhalb eines bestimmten Zeitraums aktiv waren.
POST /Gerätegruppen	Erstellen Sie eine neue Gerätegruppe.
/devicegroups/ {id} LÖSCHEN	Löscht eine Gerätegruppe.
GET /devicegroups/ {id}	Rufen Sie eine bestimmte Gerätegruppe ab.
PATCH /Gerätegruppen/ {id}	Aktualisieren Sie eine bestimmte Gerätegruppe.
GET /devicegroups/ {id} /alerts	Alles abrufen Warnungen die einer bestimmten Gerätegruppe zugewiesen sind.
POST /devicegroups/ {id} /alerts	Weisen Sie Benachrichtigungen eine bestimmte Gerätegruppe zu und heben Sie deren Zuweisung auf.
LÖSCHEN Sie /devicegroups/ {id} /alerts/ {child-id}	Heben Sie die Zuweisung einer Alarm zu einer bestimmten Gerätegruppe auf.
POST /devicegroups/ {id} /alerts/ {child-id}	Weisen Sie einer bestimmten Gerätegruppe eine Alarm zu.
GET /devicegroups/ {id} /dashboards	Rufen Sie alle Dashboards ab, die sich auf eine bestimmte Gerätegruppe beziehen.
GET /devicegroups/ {id} /devices	Ruft alle Geräte in der Gerätegruppe ab, die innerhalb eines bestimmten Zeitfensters aktiv sind. **Hinweis:** Ein Gerät gilt als inaktiv, wenn fünf Minuten lang keine Pakete gesendet oder empfangen wurden. Wenn ein Gerät jedoch nach einem Zeitraum der Inaktivität von weniger als fünf Tagen wieder Pakete sendet oder empfängt,

Betrieb	Beschreibung
	wird davon ausgegangen, dass das Gerät kontinuierlich aktiv war, auch während des Zeitraums der Inaktivität.
POST /devicegroups/ {id} /devices	Weisen Sie Geräte einer bestimmten statischen Gerätegruppe zu und heben Sie deren Zuweisung auf.
LÖSCHEN /devicegroups/ {id} /devices/ {child-id}	Heben Sie die Zuweisung eines Gerät zu einer bestimmten statischen Gerätegruppe auf.
POST /devicegroups/ {id} /devices/ {child-id}	Weisen Sie ein Gerät einer bestimmten statischen Gerätegruppe zu.
GET /devicegroups/ {id} /triggers	Ruft alle Trigger ab, die einer bestimmten Gerätegruppe zugewiesen sind.
POST /devicegroups/ {id} /triggers	Weisen Sie Triggern eine bestimmte Gerätegruppe zu und heben Sie deren Zuweisung auf.
LÖSCHEN /devicegroups/ {id} /triggers/ {child-id}	Heben Sie die Zuweisung eines Auslöser zu einer bestimmten Gerätegruppe auf.
POST /devicegroups/ {id} /triggers/ {Child-ID}	Weisen Sie einer bestimmten Gerätegruppe einen Auslöser zu.

Einzelheiten der Operation

`GET /devicegroups`

Geben Sie die folgenden Parameter an.

`since:` ***Zahl***

(Optional) Gibt nur Gerätegruppen zurück, die nach dieser Zeit geändert wurden, ausgedrückt in Millisekunden seit der Epoche.

`all:` ***Boolesch***

(Optional) Veraltet. Ersetzt durch den Typparameter.

`name:` ***Schnur***

(Optional) Der Regex-Suchwert zum Filtern der Gerätegruppen nach Namen.

`type:` ***Schnur***

(Optional) Gibt nur Gerätegruppen des angegebenen Typs zurück.

Die folgenden Werte sind gültig:

- `user_created`
- `built_in`
- `all`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "built_in": true,
    "description": "string",
    "dynamic": true,
    "editors": [],
    "field": "string",
    "filter": {},
    "id": 0,
```

```
    "include_custom_devices": true,
    "mod_time": 0,
    "name": "string",
    "value": "string"
}
```

`GET /devicegroups/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "built_in": true,
    "description": "string",
    "dynamic": true,
    "editors": [],
    "field": "string",
    "filter": {},
    "id": 0,
    "include_custom_devices": true,
    "mod_time": 0,
    "name": "string",
    "value": "string"
}
```

`POST /devicegroups`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Wendet die angegebenen Eigenschaftswerte auf die neue Gerätegruppe an.

`description:` ***Schnur***
Eine optionale Beschreibung der Gerätegruppe.

`name:` ***Schnur***
Der benutzerfreundliche Name für die Gerätegruppe.

`include_custom_devices:` ***Boolesch***
(Optional) Veraltet. Ersetzt durch den Filterparameter.

`dynamic:` ***Boolesch***
(Optional) Gibt an, ob die Gerätegruppe dynamisch ist.

`field:` ***Schnur***
Veraltet. Ersetzt durch den Filterparameter.

Die folgenden Werte sind gültig:

- `any`
- `name`
- `ip address`
- `mac address`
- `vendor`
- `type`
- `tag`
- `vlan`

- `activity`
- `node`
- `discover time`

`value:` ***Objekt***

(Optional) Veraltet. Ersetzt durch den Filterparameter.

`filter:` ***Objekt***

(Optional) Geben Sie die Filterkriterien für Suchergebnisse an.

`field:` ***Schnur***

Der Name des Feldes, nach dem die Ergebnisse gefiltert werden sollen. Die Suche vergleicht den Inhalt des Feldparameters mit dem Wert des Operandenparameters.

Die folgenden Werte sind gültig:

- `name`
- `ipaddr`
- `macaddr`
- `vendor`
- `tag`
- `activity`
- `node`
- `vlan`
- `discover_time`
- `role`
- `dns_name`
- `dhcp_name`
- `netbios_name`
- `cdp_name`
- `custom_name`
- `software`
- `model`
- `is_critical`
- `instance_id`
- `instance_name`
- `instance_type`
- `cloud_account`
- `vpc_id`
- `subnet_id`
- `is_active`
- `network_locality_type`
- `network_locality_id`
- `id`

`operator:` ***Schnur***

Die Vergleichsmethode, die angewendet wird, wenn der Operandenwert mit dem Feldinhalt verglichen wird. Alle Filterobjekte benötigen einen Operator.

Die folgenden Werte sind gültig:

- `>`
- `<`
- `<=`
- `>=`
- `=`

- `!=`
- `startswith`
- `and`
- `or`
- `not`
- `exists`
- `not_exists`
- `~`
- `!~`

`operand:` ***Zeichenfolge oder Zahl oder Objekt***

Der Wert, den die Abfrage abzugleichen versucht. Die Abfrage vergleicht den Wert des Operanden mit dem Inhalt des Feldparameters und wendet die durch den Operatorparameter angegebene Vergleichsmethode an. Sie können den Operanden als Zeichenfolge, Ganzzahl oder Objekt angeben. Informationen zu Objektwerten finden Sie in der REST-API-Leitfaden.

`rules:` ***Reihe von Objekten***

Ein Array aus einem oder mehreren Filterobjekten, die rekursiv eingebettet werden können. Für diesen Parameter sind nur die Operatoren „und“, „oder“ oder „nicht“ zulässig.

`editors:` ***Reihe von Zeichenketten***

(Optional) Die Liste der Benutzer, die die Gerätegruppe bearbeiten können.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "dynamic": true,
    "editors": [],
    "field": "string",
    "filter": {
        "field": "string",
        "operator": "string",
        "operand": "string",
        "rules": []
    },
    "include_custom_devices": true,
    "name": "string",
    "value": "string"
}
```

`DELETE /devicegroups/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`PATCH /devicegroups/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die Aktualisierungen der angegebenen Eigenschaftswerte auf eine bestimmte Gerätegruppe an.

`description:` ***Schnur***

Eine optionale Beschreibung der Gerätegruppe.

`name:` ***Schnur***

Der benutzerfreundliche Name für die Gerätegruppe.

`include_custom_devices:` ***Boolesch***

(Optional) Veraltet. Ersetzt durch den Filterparameter.

`field:` ***Schnur***

Veraltet. Ersetzt durch den Filterparameter.

Die folgenden Werte sind gültig:

- `any`
- `name`
- `ip address`
- `mac address`
- `vendor`
- `type`
- `tag`
- `vlan`
- `activity`
- `node`
- `discover time`

`value:` ***Objekt***

(Optional) Veraltet. Ersetzt durch den Filterparameter.

`filter:` ***Objekt***

(Optional) Geben Sie die Filterkriterien für Suchergebnisse an.

`editors:` ***Reihe von Zeichenketten***

(Optional) Die Liste der Benutzer, die die Gerätegruppe bearbeiten können.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "editors": [],
    "field": "string",
    "filter": {},
    "include_custom_devices": true,
    "name": "string",
    "value": "string"
}
```

`id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`GET /devicegroups/{id}/alerts`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`direct_assignments_only:` ***Boolesch***

(Optional) Beschränken Sie die Ergebnisse auf Warnungen, die direkt der Gerätegruppe zugewiesen sind.

`POST /devicegroups/{id}/alerts/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`DELETE /devicegroups/{id}/alerts/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die Alarm.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`POST /devicegroups/{id}/alerts`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Liste der eindeutigen Identifikatoren für Warnmeldungen, die der Gerätegruppe zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`GET /devicegroups/{id}/triggers`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`direct_assignments_only:` ***Boolesch***
(Optional) Beschränken Sie die Ergebnisse auf Trigger, die direkt der Gerätegruppe zugewiesen sind.

`POST /devicegroups/{id}/triggers/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`DELETE /devicegroups/{id}/triggers/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`POST /devicegroups/{id}/triggers`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Liste der eindeutigen Identifikatoren für Trigger, die der Gerätegruppe zugewiesen oder nicht zugewiesen sind.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`POST /devicegroups/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für ein Gerät.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`DELETE /devicegroups/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für ein Gerät.

`id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`POST /devicegroups/{id}/devices`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Liste der eindeutigen Identifikatoren für Geräte, die der Gerätegruppe zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***

IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***

IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`GET /devicegroups/{id}/devices`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

`active_from:` ***Zahl***

(Optional) Der Anfangszeitstempel für die Anfrage. Gibt nur Geräte zurück, die nach dieser Zeit aktiv sind. Die Zeit wird in Millisekunden seit der Epoche ausgedrückt. 0 gibt den Zeitpunkt der Anfrage an. Ein negativer Wert wird relativ zur aktuellen Uhrzeit ausgewertet. Die Standardeinheit für einen negativen Wert ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`active_until:` ***Zahl***

(Optional) Der Endzeitstempel für die Anfrage. Nur Gerät zurückgeben, die vor diesem Zeitpunkt aktiv waren. Folgt den gleichen Zeitwertrichtlinien wie der active_from−Parameter.

`limit:` ***Zahl***

(Optional) Begrenzen Sie die Anzahl der zurückgegebenen Geräte.

`offset:` ***Zahl***

(Optional) Überspringen Sie die ersten n Geräteergebnisse. Dieser Parameter wird häufig mit dem Grenzwertparameter kombiniert.

`GET /devicegroups/{id}/dashboards`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Gerätegruppe.

Unterstützte Zeiteinheiten

Für die meisten Parameter ist die Standardeinheit für die Zeitmessung Millisekunden. Die folgenden Parameter geben jedoch alternative Zeiteinheiten wie Minuten und Stunden zurück oder akzeptieren diese:

- Gerät
  - aktive_von
  - aktiv_bis

- Gerätegruppe
  - aktive_von
  - aktiv_bis
- Metriken
  - von
  - bis
- Protokoll aufzeichnen
  - von
  - bis
  - kontext_ttl

Die folgende Tabelle zeigt die unterstützten Zeiteinheiten:

Zeiteinheit	Einheitensuffix
Jahr	`y`
Monat	`M`
Woche	`w`
Tag	`d`
Stunde	`h`
Minute	`m`
Zweiter	`s`
Millisekunde	`ms`

Um für einen Parameter eine andere Zeiteinheit als Millisekunden anzugeben, hängen Sie das Einheitensuffix an den Wert an. Um beispielsweise Geräte anzufordern, die in den letzten 30 Minuten aktiv waren, geben Sie den folgenden Parameterwert an:

```
GET /api/v1/devices?active_from=-30m
```

Das folgende Beispiel spezifiziert eine Suche nach HTTP Datensätze, die vor 1 bis 2 Stunden erstellt wurden:

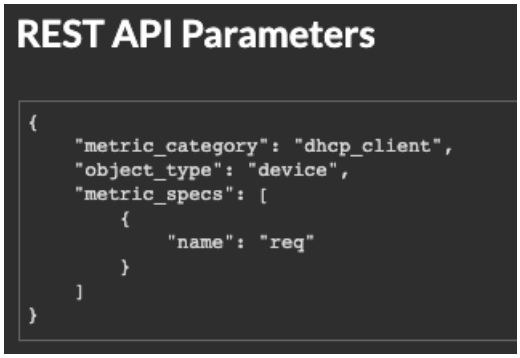
```
{
    "from": "-2h",
    "until": "-1h",
    "types": ["~http"]
}
```

Operandenwerte für Gerätegruppen

Der Vorgang POST /devicegroups ermöglicht es Ihnen, Gerätegruppen gemäß den in Filterobjekten angegebenen Kriterien zu erstellen. Jedes Objekt sollte einen eindeutigen Wert für die enthalten `operand` Feld, das für das angegebene Feld gültig ist `field` Wert.

`activity`

Um Geräte nach Metrik Aktivität auszuwählen, geben Sie die `field` Wert als `activity` und der `operand` Wert als `metric_category`. Du kannst finden `metric_category` Werte im Abschnitt REST-API-Parameter des Metrikkatalogs.



Im folgenden Beispiel werden Geräte ausgewählt, deren Metrik Aktivität für einen DHCP-Client klassifiziert ist, z. B. die Anzahl der gesendeten DHCP-Anfragen .

```
{
    "filter":  {
        "field": "activity",
        "operand": "dhcp_client",
        "operator": "="
    }
}
```

Hinweis: Rufen Sie programmgesteuert eine Liste aller Metrik Aktivitäten für ein Gerät ab über `GET /devices/{id}/activity` Betrieb. Die `stat_name` Wert entspricht dem `metric_category` Wert in der `metric_catalog`, nach dem letzten Punkt.

In der folgenden Beispielantwort ist der `stat_name` Wert ist `extrahop.device.dhcp_client`. Entfernen Sie den Text vor dem letzten Punkt, um das zu identifizieren `metric_catalog` Wert von `dhcp_client`.

```
{
    "id": 198606,
    "from_time": 1581537120000,
    "until_time": 1581542520000,
    "mod_time": 1581542533963,
    "device_id": 30096,
    "stat_name": "extrahop.device.dhcp_client"
}
```

`discover_time`

Um Geräte nach einem Zeitraum auszuwählen, geben Sie den `field` Wert als `discover_time` und ein `operand` Wert mit `from` und `until` Parameter, wobei die Werte Datumsangaben sind, ausgedrückt in Millisekunden seit der Epoche.

Im folgenden Beispiel werden Geräte ausgewählt, deren Aktivität am 21. August 2019 zwischen 13:00 Uhr und 15:00 Uhr stattfand.

```
{
    "filter":  {
        "field": "discover_time",
        "operand": {
            "from": "1566392400000",
            "until": "1566399600000"
        },
        "operator": "="
    }
}
```

`discovery_id`

Um Geräte nach eindeutiger Geräte-ID auszuwählen, geben Sie die `field` Wert als `discovery_id` und der `operand` Wert als Discovery-ID.

```
{
  "filter": {
    "field": "discovery_id",
    "operand": "c12vf90qpg290000",
    "operator": "="
  }
}
```

`ipaddr`

Um Geräte nach IP-Adresse auszuwählen, geben Sie die `field` Wert als `ipaddr` und der `operand` Wert als IP-Adresse oder CIDR-Block.

```
{
  "filter": {
    "field": "ipaddr",
    "operand": "192.168.12.0/28",
    "operator": "="
  }
}
```

`node`

Um Geräte anhand der eindeutigen ID eines auszuwählen Sensor, spezifizieren Sie die `field` Wert als `node` und der `operand` Wert als Appliance-UUID.

```
{
  "filter": {
    "field": "node",
    "operand": "qqvsplfa-zxsk-32l0-19g1-076vfr42pw31",
    "operator": "="
  }
}
```

`macaddr`

Um Geräte nach MAC-Adresse auszuwählen, geben Sie den Feldwert als `macaddr` und der Operandenwert als MAC-Adresse des Gerät. Das folgende Beispiel gibt Ergebnisse für Geräte mit einer MAC-Adresse von `C1:1C:N2:0Q:PJ:10` oder `C1:1C:N2:0Q:PJ:11`.

```
{
  "filter": {
    "operator": "or",
    "rules": [
      {
        "field": "macaddr",
        "operand": "C1:1C:N2:0Q:PJ:10",
        "operator": "="
      },
      {
        "field": "macaddr",
        "operand": "C1:1C:N2:0Q:PJ:11",
        "operator": "="
      }
    ]
```

```
  }
}
```

`name`

Um Geräte nach Anzeigenamen auszuwählen, geben Sie den `field` Wert als Name und `operand` Wert als Gerätename oder als Regex-Zeichenfolge.

```
{
  "filter": {
    "field": "name",
    "operand": "VMware B2CEB6",
    "operator": "="
  }
}
```

Netzwerk-Lokalitäts-ID

Um Geräte nach Netzwerkstandort auszuwählen, geben Sie den `field` Wert als `network_locality_id` und der Operandenwert als Netzwerk-Lokalitäts-ID.

```
{
  "filter": {
    "field": "network_locality_id",
    "operand": 123,
    "operator": "="
  }
}
```

`role`

Um Geräte nach Rolle auszuwählen, geben Sie die `field` Wert als `role` und der `operand` Wert als Geräterolle.

```
{
  "filter": {
    "field": "role",
    "operand": "voip_phone",
    "operator": "="
  }
}
```

`software`

Um Geräte anhand der auf dem Gerät ausgeführten Software auszuwählen, geben Sie die `field` Wert als `software` und der `operand` Wert als ID, die dieser Software auf dem ExtraHop-System zugeordnet ist, oder als Regex-Zeichenfolge.

```
{
  "filter": {
    "field": "software",
    "operand": "windows_10",
    "operator": "="
  }
}
```

Hinweis: Rufen Sie programmgesteuert eine Liste aller Software-IDs ab, die einem Gerät zugeordnet sind, über `GET /devices/{id}/software` Betrieb.

In der folgenden Beispielantwort ist der `id` Wert für die Software ist `windows_10`.

```
[
  {
    "software_type": "OS",
    "name": "Windows",
    "version": "10",
    "description": null,
    "id": "windows_10"
  }
]
```

`tag`

Um Geräte nach Tag auszuwählen, geben Sie den `field` Wert als `tag` und der `operand` Wert als Tagname oder als Regex-Zeichenfolge.

```
{
  "filter": {
    "field": "tag",
    "operand": "Custom Tag",
    "operator": "="
  }
}
```

Hinweis: Rufen Sie programmgesteuert eine Liste aller Geräte-Tags ab über `GET /devices/{id}/tags` Betrieb.

In der folgenden Beispielantwort ist der `name` Wert für das Tag ist `Custom Tag`.

```
[
  {
    "mod_time": 1521577040934,
    "id": 19,
    "name": "Custom Tag"
  }
]
```

vlan

Um Geräte anhand der ID eines VLAN auszuwählen, geben Sie die `field` Wert als `vlan` und der `operand` Wert als ID des VLAN.

```
{
  "filter": {
    "field": "vlan",
    "operand": "0",
    "operator": "="
  }
}
```

Suche mit regulären Ausdrücken (Regex)

Mit Sicherheit `field` Werte, die Zeichenfolge kann in Regex-Syntax sein. Spezifizieren Sie die `operand` Wert als Objekt mit einem `value` Parameter mit der Regex-Syntax, die Sie abgleichen möchten, und einem `is_regex` Parameter, der auf gesetzt ist `true`. Im folgenden Beispiel werden Geräte mit DNS-Namen ausgewählt, die auf enden `com`.

```
{
    "filter":  {
```

```
        "field": "dns_name",
        "operand": {
            "value": ".*?com",
            "is_regex": true
        },
        "operator": "="
    }
}
```

Ein `operand` Feld mit Regex-Syntax ist gültig für Folgendes `field` Werte:

- cdp_name
- Benutzerdefinierter_Name
- DNS-Name
- DHCP-Name
- Modell
- Name
- netbios_name
- Software
- Tag
- Lieferant

Geben Sie mehrere Kriterien an

Sie können mehrere Kriterien angeben mit dem `rules` Feld. Das folgende Beispiel gibt Ergebnisse für Geräte mit einer IP-Adresse von `192.168.12.0` oder `192.168.12.1`.

```
{
  "filter": {
    "operator": "or",
    "rules": [
      {
        "field": "ipaddr",
        "operand": "192.168.12.0",
        "operator": "="
      },
      {
        "field": "ipaddr",
        "operand": "192.168.12.1",
        "operator": "="
      }
    ]
  }
}
```

Hinweis Sie können nicht mehr als 1000 Regeln für eine Gerätegruppe angeben.

Erkennungen

Mit der Ressource Erkennungen können Sie Erkennungen abrufen, die vom ExtraHop-System identifiziert wurden.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /Erkennungen	Ruft alle Funde ab.

Bedienung	Beschreibung
GET /Erkennungen/Formate	Ruft alle Erkennungstypen ab.
GET /detections/formats/ {id}	Ruft einen bestimmten Erkennungstyp ab.
POST /Erkennungen/Formate	Erstellen Sie einen neuen benutzerdefinierten Erkennungstyp.
LÖSCHE /detections/formats/ {id}	Löscht einen bestimmten benutzerdefinierten Erkennungstyp.
PATCH /Erkennungen/Formate/ {id}	Aktualisieren Sie einen bestimmten benutzerdefinierten Erkennungstyp.
GET /Erkennungen/Regeln/Verbergen	Ruft alle Tuning-Regeln ab.
GET /detections/rules/hiding/ {id}	Ruft eine bestimmte Tuning-Regel ab.
POST /Erkennungen/Regeln/Verbergen	Erstellen Sie eine Optimierungsregel.
LÖSCHEN /detections/rules/hiding/ {id}	Löschen Sie eine Tuning-Regel.
PATCH /Erkennungen/Regeln/Ausblenden/ {id}	Aktualisieren Sie eine Tuning-Regel.
POST /Erkennungen/Suche	Ruft Erkennungen ab, die den angegebenen Suchkriterien entsprechen.
PATCH /Erkennungen/Tickets	Aktualisieren Sie ein Ticket, das mit Erkennungen verknüpft ist.
GET /Erkennungen/ {id}	Ruft eine bestimmte Erkennung ab.
GET /Erkennungen/ {id} /untersuchungen	Ruft alle Untersuchungen ab, in denen sich eine bestimmte Erkennung befindet
PATCH /Erkennungen/ {id}	Aktualisieren Sie eine Erkennung.
/detections/ {id} /notes LÖSCHEN	Löscht die Notizen für eine bestimmte Erkennung.
GET /detections/ {id} /notes	Ruft die Notizen für eine bestimmte Erkennung ab.
PUT /Erkennungen/ {id} /notes	Erstellen oder ersetzen Sie Notizen für eine bestimmte Erkennung.
GET /detections/ {id} /related	Ruft alle Funde ab, die sich auf eine bestimmte Erkennung beziehen.

Einzelheiten der Operation

`GET /detections/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Erkennung.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "appliance_id": 0,
    "assignee": "string",
    "categories": [
        "string"
    ],
```

```
    "create_time": 0,
    "description": "string",
    "end_time": 0,
    "id": 0,
    "is_user_created": true,
    "mitre_tactics": [],
    "mitre_techniques": [],
    "mod_time": 0,
    "participants": [],
    "properties": {},
    "recommended": true,
    "recommended_factors": [],
    "resolution": "string",
    "risk_score": 0,
    "start_time": 0,
    "status": "string",
    "ticket_id": "string",
    "ticket_url": "string",
    "title": "string",
    "type": "string",
    "update_time": 0,
    "url": "string"
}
```

`GET /detections`

Geben Sie die folgenden Parameter an.

`limit:` ***Zahl***

(Optional) Beschränken Sie die Anzahl der zurückgegebenen Erkennungen auf die angegebene Höchstzahl. Eine zufällige Auswahl von Erkennungen wird zurückgegeben.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "appliance_id": 0,
    "assignee": "string",
    "categories": [
        "string"
    ],
    "create_time": 0,
    "description": "string",
    "end_time": 0,
    "id": 0,
    "is_user_created": true,
    "mitre_tactics": [],
    "mitre_techniques": [],
    "mod_time": 0,
    "participants": [],
    "properties": {},
    "recommended": true,
    "recommended_factors": [],
    "resolution": "string",
    "risk_score": 0,
    "start_time": 0,
    "status": "string",
    "ticket_id": "string",
    "ticket_url": "string",
    "title": "string",
    "type": "string",
    "update_time": 0,
    "url": "string"
```

```
}
```

`POST /detections/search`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Suchparameter für die Erkennung.

`filter:` ***Objekt***

Erkennungsspezifische Filter.

`category:` ***Schnur***

Veraltet. Ersetzt durch das Feld Kategorien.

`categories:` ***Reihe von Zeichenketten***

Gibt Erkennungen aus den angegebenen Kategorien zurück.

`assignee:` ***Reihe von Zeichenketten***

Gibt Erkennungen zurück, die dem angegebenen Benutzer zugewiesen sind. Geben Sie „.none“ an, um nach nicht zugewiesenen Funden zu suchen, oder geben Sie „.me“ an, um nach Funden zu suchen, die dem authentifizierten Benutzer zugewiesen sind.

`ticket_id:` ***Reihe von Zeichenketten***

Gibt Erkennungen zurück, die den angegebenen Tickets zugeordnet sind. Geben Sie „.none“ an, um nach Entdeckungen zu suchen, die nicht mit Tickets verknüpft sind.

`status:` ***Reihe von Zeichenketten***

Gibt Erkennungen mit dem angegebenen Status zurück. Um nach Erkennungen mit einem Nullstatus zu suchen, der im ExtraHop-System als Offen angezeigt wird, geben Sie „.none“ an. Sie können den Status einer Erkennung nur dann über die REST-API auf „neu“ ändern, wenn Ticket-Tracking durch Dritte ist aktiviert.

Die folgenden Werte sind gültig:

- `new`
- `in_progress`
- `closed`
- `acknowledged`

`resolution:` ***Reihe von Zeichenketten***

Gibt Erkennungen für Tickets mit der angegebenen Auflösung zurück. Geben Sie „.none“ an, um nach Erkennungen ohne Auflösung zu suchen.

Die folgenden Werte sind gültig:

- `action_taken`
- `no_action_taken`

`types:` ***Reihe von Zeichenketten***

Gibt Erkennungen mit den angegebenen Typen zurück.

`risk_score_min:` ***Zahl***

Gibt Erkennungen mit Risikoeinstufungen zurück, die größer oder gleich dem angegebenen Wert sind.

`recommended:` ***Boolesch***

Gibt die für die Triage empfohlenen Erkennungen zurück. Dieses Feld ist nur auf einer Konsole gültig.

`from:` ***Zahl***

Gibt Erkennungen zurück, die nach dem angegebenen Datum aufgetreten sind, ausgedrückt in Millisekunden seit der Epoche. Erkennungen, die vor dem angegebenen Datum begonnen

haben, werden zurückgegeben, wenn die Erkennung zu diesem Zeitpunkt noch nicht abgeschlossen war.

`limit:` ***Zahl***

Gibt nicht mehr als die angegebene Anzahl von Erkennungen zurück.

`offset:` ***Zahl***

Die Anzahl der Erkennungen, die bei der Paginierung übersprungen werden sollen.

`sort:` ***Reihe von Objekten***

Sortiert die zurückgegebenen Erkennungen nach den angegebenen Feldern. Standardmäßig werden Erkennungen nach dem Zeitpunkt der letzten Aktualisierung und dann nach der ID in aufsteigender Reihenfolge sortiert.

`direction:` ***Schnur***

Die Reihenfolge, in der zurückgegebene Erkennungen sortiert werden.

Die folgenden Werte sind gültig:

- `asc`
- `desc`

`field:` ***Schnur***

Das Feld, nach dem Erkennungen sortiert werden sollen.

`until:` ***Zahl***

Gibt Erkennungen zurück, die vor dem angegebenen Datum endeten, ausgedrückt in Millisekunden seit der Epoche.

`update_time:` ***Zahl***

Gibt Erkennungen zurück, die sich auf Ereignisse beziehen, die nach dem angegebenen Datum eingetreten sind, ausgedrückt in Millisekunden seit der Epoche. Beachten Sie, dass der ExtraHop Machine Learning Service historische Daten analysiert, um Erkennungen zu generieren. Daher gibt es eine Zeitverzögerung zwischen dem Auftreten der Ereignisse, die diese Erkennungen verursachen, und dem Zeitpunkt, an dem die Erkennungen generiert werden. Wenn Sie mehrmals im gleichen update_time-Fenster nach Entdeckungen suchen, werden bei der späteren Suche möglicherweise Erkennungen zurückgegeben, die bei der vorherigen Suche nicht gefunden wurden.

`mod_time:` ***Zahl***

Gibt Erkennungen zurück, die nach dem angegebenen Datum aktualisiert wurden, ausgedrückt in Millisekunden seit der Epoche.

`create_time:` ***Zahl***

Gibt Erkennungen zurück, die nach dem angegebenen Datum erstellt wurden, ausgedrückt in Millisekunden seit der Epoche. Für Sensoren gibt dies Erkennungen zurück, die nach dem angegebenen Datum generiert wurden. Bei Konsolen gibt dies Erkennungen zurück, die nach dem angegebenen Datum zum ersten Mal mit der Konsole synchronisiert wurden.

`id_only:` ***Boolesch***

(Optional) Gibt nur die IDs der Funde zurück.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "create_time": 0,
    "filter": {
        "category": "string",
        "categories": [],
        "assignee": [],
        "ticket_id": [],
        "status": [],
        "resolution": [],
        "types": [],
```

```
        "risk_score_min": 0,
        "recommended": true
    },
    "from": 0,
    "id_only": true,
    "limit": 0,
    "mod_time": 0,
    "offset": 0,
    "sort": {
        "direction": "string",
        "field": "string"
    },
    "until": 0,
    "update_time": 0
}
```

`PATCH /detections/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Erkennung.

`body:` ***Objekt***

Die zu aktualisierenden Erkennungsparameter.

`ticket_id:` ***Schnur***

Die ID des Tickets, das mit der Erkennung verknüpft ist.

`assignee:` ***Schnur***

Der Empfänger der Erkennung oder des Tickets, das mit der Erkennung verknüpft ist.

`status:` ***Schnur***

Der Status der Erkennung oder des Tickets, das mit der Erkennung verknüpft ist. Wenn der Wert Null ist, lautet der im ExtraHop-System angezeigte Status Offen. Der Wert „new“ kann nur über die REST-API angegeben werden, wenn Ticket-Tracking durch Dritte ist aktiviert.

Die folgenden Werte sind gültig:

- `new`
- `in_progress`
- `closed`
- `acknowledged`

`resolution:` ***Schnur***

Die Auflösung der Erkennung oder des mit der Erkennung verknüpften Tickets.

Die folgenden Werte sind gültig:

- `action_taken`
- `no_action_taken`

`participants:` ***Reihe von Objekten***

Eine Liste der Geräte und Anwendungen, die mit der Erkennung verknüpft sind. Sie können bestimmte Felder für einen Teilnehmer ändern, aber Sie können einer Erkennung keine neuen Teilnehmer hinzufügen.

`id:` ***Zahl***

Die ID des Teilnehmer, der mit der Erkennung verknüpft ist.

`usernames:` ***Reihe von Zeichenketten***

Die Benutzernamen, die dem Teilnehmer über die REST-API zugeordnet sind.

`origins:` ***Reihe von Zeichenketten***

Die Quell-IP-Adressen, die dem Teilnehmer über die REST-API zugeordnet sind.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assignee": "string",
    "participants": {
        "id": 0,
        "usernames": [],
        "origins": []
    },
    "resolution": "string",
    "status": "string",
    "ticket_id": "string"
}
```

`PATCH /detections/tickets`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die zu aktualisierenden Erkennungsticketwerte.

`ticket_id:` ***Schnur***

Die ID des Tickets, das mit der Erkennung verknüpft ist.

`assignee:` ***Schnur***

Der Empfänger des Tickets, das mit der Erkennung verknüpft ist.

`status:` ***Schnur***

Der Status des Tickets, das mit der Erkennung verknüpft ist.

Die folgenden Werte sind gültig:

- `new`
- `in_progress`
- `closed`
- `acknowledged`

`resolution:` ***Schnur***

Die Auflösung des Tickets, das mit der Erkennung verknüpft ist.

Die folgenden Werte sind gültig:

- `action_taken`
- `no_action_taken`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assignee": "string",
    "resolution": "string",
    "status": "string",
    "ticket_id": "string"
}
```

`GET /detections/{id}/related`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die ID der Erkennung, für die verwandte Erkennungen abgerufen werden sollen.

`from:` ***Zahl***

Gibt Erkennungen zurück, die nach dem angegebenen Datum aufgetreten sind, ausgedrückt in Millisekunden seit der Epoche. Erkennungen, die vor dem angegebenen Datum begonnen haben, werden zurückgegeben, wenn die Erkennung zu diesem Zeitpunkt noch nicht abgeschlossen war.

`until:` ***Zahl***

Gibt Erkennungen zurück, die vor dem angegebenen Datum endeten, ausgedrückt in Millisekunden seit der Epoche.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "appliance_id": 0,
    "assignee": "string",
    "categories": [
        "string"
    ],
    "create_time": 0,
    "description": "string",
    "end_time": 0,
    "id": 0,
    "is_user_created": true,
    "mitre_tactics": [],
    "mitre_techniques": [],
    "mod_time": 0,
    "participants": [],
    "properties": {},
    "recommended": true,
    "recommended_factors": [],
    "resolution": "string",
    "risk_score": 0,
    "start_time": 0,
    "status": "string",
    "ticket_id": "string",
    "ticket_url": "string",
    "title": "string",
    "type": "string",
    "update_time": 0,
    "url": "string"
}
```

`GET /detections/{id}/investigations`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die ID der Erkennung, für die verwandte Untersuchungen abgerufen werden sollen.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "appliance_id": 0,
    "assignee": "string",
    "categories": [
        "string"
    ],
    "create_time": 0,
    "description": "string",
    "end_time": 0,
```

```
    "id": 0,
    "is_user_created": true,
    "mitre_tactics": [],
    "mitre_techniques": [],
    "mod_time": 0,
    "participants": [],
    "properties": {},
    "recommended": true,
    "recommended_factors": [],
    "resolution": "string",
    "risk_score": 0,
    "start_time": 0,
    "status": "string",
    "ticket_id": "string",
    "ticket_url": "string",
    "title": "string",
    "type": "string",
    "update_time": 0,
    "url": "string"
}
```


`GET /detections/formats`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.


```
{
    "author": "string",
    "categories": [],
    "display_name": "string",
    "is_user_created": true,
    "last_updated": 0,
    "mitre_categories": [],
    "properties": {},
    "released": 0,
    "status": "string",
    "type": "string"
}
```


`GET /detections/formats/{id}`

Geben Sie die folgenden Parameter an.

`id`: ***Schnur***

Der Zeichenkettenbezeichner des Erkennungsformats.

`built_in_only`: ***Boolesch***

(Optional) Wenn dieses Feld den Wert true hat, werden nur integrierte Erkennungsformate zurückgegeben. Wenn dieses Feld falsch ist und sowohl ein benutzerdefiniertes Format als auch ein integriertes Format dieselbe ID haben, wird das benutzerdefinierte Format zurückgegeben. Der Standardwert ist falsch.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.


```
{
    "author": "string",
    "categories": [],
    "display_name": "string",
    "is_user_created": true,
    "last_updated": 0,
    "mitre_categories": [],
```

```
    "properties": {},
    "released": 0,
    "status": "string",
    "type": "string"
}
```

`POST /detections/formats`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Parameter des Erkennungsformats.

`type:` ***Schnur***

Ein Zeichenkettenbezeichner für den Erkennungstyp. Die Zeichenfolge darf nur Buchstaben, Zahlen und Unterstriche enthalten. Obwohl Erkennungstypen in integrierten Formaten einzigartig sind und Erkennungstypen in benutzerdefinierten Formaten eindeutig sind, können ein integriertes und ein benutzerdefiniertes Format denselben Erkennungstyp gemeinsam haben.

`display_name:` ***Schnur***

Der Anzeigename des Erkennungstyps, der auf der Seite „Erkennungen“ im ExtraHop-System angezeigt wird.

`mitre_categories:` ***Reihe von Zeichenketten***

(Optional) Die IDs der MITRE-Techniken, die mit der Erkennung verknüpft sind.

`author:` ***Schnur***

(Optional) Der Autor des Erkennungsformats.

`categories:` ***Reihe von Zeichenketten***

(Optional) Die Liste der Kategorien, zu denen die Erkennung gehört. Geben Sie für POST- und PATCH-Operationen eine Liste mit einer einzigen Zeichenfolge an. Sie können nicht mehr als eine Kategorie für benutzerdefinierte Erkennungsformate angeben. Die Kategorie „Perf“ oder „Sek“ wird automatisch zu allen Erkennungsformaten hinzugefügt.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "author": "string",
    "categories": [],
    "display_name": "string",
    "mitre_categories": [],
    "type": "string"
}
```

`DELETE /detections/formats/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***

Der Zeichenkettenbezeichner des Erkennungsformats.

`PATCH /detections/formats/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***

Der Zeichenkettenbezeichner des Erkennungsformats.

`body:` ***Objekt***
Die Parameter des Erkennungsformats.

`GET /detections/rules/hiding`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "author": "string",
    "create_time": 0,
    "description": "string",
    "detection_type": "string",
    "detections_hidden": 0,
    "enabled": true,
    "expiration": 0,
    "hide_past_detections": true,
    "id": 0,
    "offender": {},
    "participants_hidden": 0,
    "properties": [],
    "victim": {}
}
```

`GET /detections/rules/hiding/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Der eindeutige Bezeichner für die Tuning-Regel.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "author": "string",
    "create_time": 0,
    "description": "string",
    "detection_type": "string",
    "detections_hidden": 0,
    "enabled": true,
    "expiration": 0,
    "hide_past_detections": true,
    "id": 0,
    "offender": {},
    "participants_hidden": 0,
    "properties": [],
    "victim": {}
}
```

`POST /detections/rules/hiding`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Parameter der Tuning-Regel.

`offender:` ***Objekt***

Der Täter, für den diese Tuning-Regel gilt. Geben Sie ein detection_hiding_participant-Objekt an, um die Regel auf ein bestimmtes Opfer anzuwenden, oder geben Sie „Any“ an, um die Regel auf einen beliebigen Täter anzuwenden.

`object_type:` ***Schnur***

Die Art des Teilnehmer.

Die folgenden Werte sind gültig:

- `device`
- `device_group`
- `ipaddr`
- `locality_type`
- `network_locality`
- `hostname`
- `scanner_service`

`object_id:` ***Zahl***

Die ID für das Gerät, die Gerätegruppe oder den Netzwerkstandort. Diese Option ist nur gültig, wenn der Objekttyp „Gerät“, „device_group“ oder „network_locality“ ist.

`object_value:` ***Array oder String***

Die IP-Adresse oder der CIDR-Block des Teilnehmer. Sie können eine einzelne Adresse oder einen Block in einer Zeichenfolge oder mehrere Adressen oder Blöcke in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „ipaddr“ ist.

`object_locality:` ***Schnur***

Der Netzwerklokalitätstyp des Teilnehmer. Geben Sie entweder „extern“ oder „intern“ an. Diese Option ist nur gültig, wenn der Objekttyp „locality_type“ ist.

Die folgenden Werte sind gültig:

- `internal`
- `external`

`object_scanner:` ***Array oder String***

Der Name eines externen Scandienstes. Sie können einen einzelnen Dienst in einer Zeichenfolge oder mehrere Werte in einem Array angeben. Sie können auch „Beliebig“ angeben, um einen beliebigen Scandienst auszuwählen. Diese Option ist nur gültig, wenn der Objekttyp „scanner_service“ ist.

`object_hostname:` ***Array oder String***

Der Hostname eines Teilnehmer. Sie können einen einzelnen Hostnamen in einer Zeichenfolge oder mehrere Hostnamen in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „hostname“ ist.

`victim:` ***Objekt***

Das Opfer, für das diese Tuning-Regel gilt. Geben Sie ein detection_hiding_participant-Objekt an, um die Regel auf ein bestimmtes Opfer anzuwenden, oder geben Sie „Any“ an, um die Regel auf ein beliebiges Opfer anzuwenden.

`object_type:` ***Schnur***

Die Art des Teilnehmer.

Die folgenden Werte sind gültig:

- `device`
- `device_group`
- `ipaddr`
- `locality_type`

- `network_locality`
- `hostname`
- `scanner_service`

`object_id:` ***Zahl***

Die ID für das Gerät, die Gerätegruppe oder den Netzwerkstandort. Diese Option ist nur gültig, wenn der Objekttyp „Gerät“, „device_group“ oder „network_locality“ ist.

`object_value:` ***Array oder String***

Die IP-Adresse oder der CIDR-Block des Teilnehmer. Sie können eine einzelne Adresse oder einen Block in einer Zeichenfolge oder mehrere Adressen oder Blöcke in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „ipaddr“ ist.

`object_locality:` ***Schnur***

Der Netzwerklokalitätstyp des Teilnehmer. Geben Sie entweder „extern“ oder „intern“ an. Diese Option ist nur gültig, wenn der Objekttyp „locality_type“ ist.

Die folgenden Werte sind gültig:

- `internal`
- `external`

`object_scanner:` ***Array oder String***

Der Name eines externen Scandienstes. Sie können einen einzelnen Dienst in einer Zeichenfolge oder mehrere Werte in einem Array angeben. Sie können auch „Beliebig“ angeben, um einen beliebigen Scandienst auszuwählen. Diese Option ist nur gültig, wenn der Objekttyp „scanner_service“ ist.

`object_hostname:` ***Array oder String***

Der Hostname eines Teilnehmer. Sie können einen einzelnen Hostnamen in einer Zeichenfolge oder mehrere Hostnamen in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „hostname“ ist.

`expiration:` ***Zahl***

Die Zeit, in der die Tuning-Regel abläuft, ausgedrückt in Millisekunden seit der Epoche. Ein Wert von Null oder 0 gibt an, dass die Regel nicht abläuft.

`description:` ***Schnur***

(Optional) Die Beschreibung der Tuning-Regel.

`detection_type:` ***Schnur***

Der Erkennungstyp, für den diese Optimierungsregel gilt. Zeigen Sie eine Liste der gültigen Felder für „type“ an, indem Sie die Operation GET /detections/formats ausführen. Geben Sie „all_performance“ oder „all_security“ an, um die Regel auf alle Leistungs- oder Sicherheitserkennungen anzuwenden.

`properties:` ***Reihe von Objekten***

(Optional) Die Filterkriterien für Erkennungseigenschaften.

`property:` ***Schnur***

Der Name der Eigenschaft, die gefiltert werden soll.

`operator:` ***Schnur***

Die Vergleichsmethode wird angewendet, wenn der Operandenwert mit dem Wert der Erkennungseigenschaft verglichen wird.

Die folgenden Werte sind gültig:

- `=`
- `!=`
- `~`
- `!~`
- `in`

`operand:` ***Zeichenfolge oder Zahl oder Objekt***

Der Wert, den der Filter abzugleichen versucht. Der Filter vergleicht den Wert des Operanden mit dem Wert der Erkennungseigenschaft und wendet die im Operatorparameter angegebene Vergleichsmethode an. Sie können den Operanden als Zeichenfolge, Ganzzahl oder Objekt angeben. Weitere Informationen finden Sie in der REST-API-Leitfaden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "detection_type": "string",
    "expiration": 0,
    "offender": {
        "object_type": "string",
        "object_id": 0,
        "object_value": "array",
        "object_locality": "string",
        "object_scanner": "array",
        "object_hostname": "array"
    },
    "properties": {
        "property": "string",
        "operator": "string",
        "operand": "string"
    },
    "victim": {
        "object_type": "string",
        "object_id": 0,
        "object_value": "array",
        "object_locality": "string",
        "object_scanner": "array",
        "object_hostname": "array"
    }
}
```

`PATCH /detections/rules/hiding/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Der eindeutige Bezeichner für die Tuning-Regel.

`body:` ***Objekt***

Die zu aktualisierenden Tuning-Regelfelder.

`enabled:` ***Boolesch***

Gibt an, ob die Optimierungsregel aktiviert ist.

`expiration:` ***Zahl***

Die Zeit, in der die Tuning-Regel abläuft, ausgedrückt in Millisekunden seit der Epoche. Ein Wert von Null oder 0 gibt an, dass die Regel nicht abläuft.

`description:` ***Schnur***

Die Beschreibung der Tuning-Regel.

`offender:` ***Objekt***

Der Täter, für den diese Tuning-Regel gilt. Geben Sie ein detection_hiding_participant-Objekt an, um die Regel auf ein bestimmtes Opfer anzuwenden, oder geben Sie „Any“ an, um die Regel auf einen beliebigen Täter anzuwenden.

`object_type:` ***Schnur***

Die Art des Teilnehmer.

Die folgenden Werte sind gültig:

- `device`
- `device_group`
- `ipaddr`
- `locality_type`
- `network_locality`
- `hostname`
- `scanner_service`

`object_id:` ***Zahl***

Die ID für das Gerät, die Gerätegruppe oder den Netzwerkstandort. Diese Option ist nur gültig, wenn der Objekttyp „Gerät“, „device_group“ oder „network_locality“ ist.

`object_value:` ***Array oder String***

Die IP-Adresse oder der CIDR-Block des Teilnehmer. Sie können eine einzelne Adresse oder einen Block in einer Zeichenfolge oder mehrere Adressen oder Blöcke in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „ipaddr“ ist.

`object_locality:` ***Schnur***

Der Netzwerklokalitätstyp des Teilnehmer. Geben Sie entweder „extern“ oder „intern“ an. Diese Option ist nur gültig, wenn der Objekttyp „locality_type“ ist.

Die folgenden Werte sind gültig:

- `internal`
- `external`

`object_scanner:` ***Array oder String***

Der Name eines externen Scandienstes. Sie können einen einzelnen Dienst in einer Zeichenfolge oder mehrere Werte in einem Array angeben. Sie können auch „Beliebig“ angeben, um einen beliebigen Scandienst auszuwählen. Diese Option ist nur gültig, wenn der Objekttyp „scanner_service“ ist.

`object_hostname:` ***Array oder String***

Der Hostname eines Teilnehmer. Sie können einen einzelnen Hostnamen in einer Zeichenfolge oder mehrere Hostnamen in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „hostname“ ist.

`victim:` ***Objekt***

Das Opfer, für das diese Tuning-Regel gilt. Geben Sie ein detection_hiding_participant-Objekt an, um die Regel auf ein bestimmtes Opfer anzuwenden, oder geben Sie „Any“ an, um die Regel auf ein beliebiges Opfer anzuwenden.

`object_type:` ***Schnur***

Die Art des Teilnehmer.

Die folgenden Werte sind gültig:

- `device`
- `device_group`
- `ipaddr`
- `locality_type`
- `network_locality`
- `hostname`
- `scanner_service`

`object_id:` ***Zahl***

Die ID für das Gerät, die Gerätegruppe oder den Netzwerkstandort. Diese Option ist nur gültig, wenn der Objekttyp „Gerät“, „device_group“ oder „network_locality“ ist.

`object_value:` ***Array oder String***

Die IP-Adresse oder der CIDR-Block des Teilnehmer. Sie können eine einzelne Adresse oder einen Block in einer Zeichenfolge oder mehrere Adressen oder Blöcke in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „ipaddr“ ist.

`object_locality:` ***Schnur***

Der Netzwerklokalitätstyp des Teilnehmer. Geben Sie entweder „extern“ oder „intern“ an. Diese Option ist nur gültig, wenn der Objekttyp „locality_type“ ist.

Die folgenden Werte sind gültig:

- `internal`
- `external`

`object_scanner:` ***Array oder String***

Der Name eines externen Scandienstes. Sie können einen einzelnen Dienst in einer Zeichenfolge oder mehrere Werte in einem Array angeben. Sie können auch „Beliebig“ angeben, um einen beliebigen Scandienst auszuwählen. Diese Option ist nur gültig, wenn der Objekttyp „scanner_service“ ist.

`object_hostname:` ***Array oder String***

Der Hostname eines Teilnehmer. Sie können einen einzelnen Hostnamen in einer Zeichenfolge oder mehrere Hostnamen in einem Array angeben. Diese Option ist nur gültig, wenn der Objekttyp „hostname“ ist.

`properties:` ***Reihe von Objekten***

Die Filterkriterien für Erkennungseigenschaften.

`property:` ***Schnur***

Der Name der Eigenschaft, die gefiltert werden soll.

`operator:` ***Schnur***

Die Vergleichsmethode wird angewendet, wenn der Operandenwert mit dem Wert der Erkennungseigenschaft verglichen wird.

Die folgenden Werte sind gültig:

- `=`
- `!=`
- `~`
- `!~`
- `in`

`operand:` ***Zeichenfolge oder Zahl oder Objekt***

Der Wert, den der Filter abzugleichen versucht. Der Filter vergleicht den Wert des Operanden mit dem Wert der Erkennungseigenschaft und wendet die im Operatorparameter angegebene Vergleichsmethode an. Sie können den Operanden als Zeichenfolge, Ganzzahl oder Objekt angeben. Weitere Informationen finden Sie in der REST-API-Leitfaden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "enabled": true,
    "expiration": 0,
    "offender": {
        "object_type": "string",
```

```
        "object_id": 0,
        "object_value": "array",
        "object_locality": "string",
        "object_scanner": "array",
        "object_hostname": "array"
    },
    "properties": {
        "property": "string",
        "operator": "string",
        "operand": "string"
    },
    "victim": {
        "object_type": "string",
        "object_id": 0,
        "object_value": "array",
        "object_locality": "string",
        "object_scanner": "array",
        "object_hostname": "array"
    }
}
```

`DELETE /detections/rules/hiding/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Der eindeutige Bezeichner für die Tuning-Regel.

`GET /detections/{id}/notes`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Erkennung.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "author": "string",
    "note": "string",
    "update_time": 0
}
```

`DELETE /detections/{id}/notes`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Erkennung.

`PUT /detections/{id}/notes`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die Erkennung.

`body:` ***Objekt***
Die Parameter der Erkennungsnotiz.

Operandenwerte für Regeln zur Abstimmung von Erkennungseigenschaften

Die `POST /detections/rules/hiding` Mithilfe dieses Vorgangs können Sie Optimierungsregeln erstellen, die Erkennungen auf der Grundlage von Erkennungseigenschaften filtern. Sie können Filterkriterien für Erkennungseigenschaften in Objekten angeben. Jedes Objekt sollte einen eindeutigen Wert für die enthalten `operand` Feld, das für das angegebene Feld gültig ist `property` Wert.

Hinweis: Sie können gültige Eigenschaftswerte abrufen über `GET /detections/formats` Betrieb. Sehen Sie die Schlüssel des `properties` Objekt in der Antwort. Im folgenden Beispiel ist der `property` Wert ist `s3_bucket`:

```
    "properties": {
      "s3_bucket": {
        "is_optional": true,
        "status": "active",
        "is_tunable": true,
        "data_type": "string"
      }
    }
```

Die `is_tunable` Feld gibt an, ob Sie eine Optimierungsregel auf der Grundlage der Eigenschaft erstellen können.

`registered_domain_name`

Um Regeln für einen registrierten Domänenname auszublenden, geben Sie den `property` Wert als `registered_domain_name` und der `operand` Wert als Domänenname.

Die folgende Beispielregel verbirgt DNS-Tunnelerkennungen für `example.com`.

```
{
    "detection_type": "dns_tunnel",
    "expiration": null,
    "offender": "Any",
    "victim": "Any",
    "properties": [
        {
            "operand": "example.com",
            "operator": "=",
            "property": "registered_domain_name"
        }
    ]
}
```

`uris`

Um Regeln anhand eines URI auszublenden, geben Sie den `property` Wert als `uris` und der `operand` Wert als URI.

Die folgende Beispielregel verbirgt Erkennungen von SQL-Injection-Angriffen (SQLi) für `http://example.com/test`.

```
{
    "detection_type": "sqli_attack",
    "expiration": null,
    "offender": "Any",
    "victim": "Any",
    "properties": [
        {
            "operand": "http://example.com/test",
            "operator": "=",
            "property": "uris"
```

```
        }
      ]
}
```

`top_level_domain`

Um Regeln für einen Top-Level-Domainnamen auszublenden, geben Sie den `property` Wert als `top_level_domain` und der `operand` Wert als Top-Level-Domainname.

Die folgende Beispielregel verbirgt Erkennungen verdächtiger Top-Level-Domains für `org` Top-Level-Domain.

```
{
    "detection_type": "suspicious_tld",
    "expiration": null,
    "offender": "Any",
    "victim": "Any",
    "properties": [
        {
            "operand": "org",
            "operator": "=",
            "property": "top_level_domain"
        }
    ]
}
```

Suche mit regulären Ausdrücken (Regex)

Mit Sicherheit `property` Werte, die Zeichenfolge kann in Regex-Syntax sein. Spezifizieren Sie die `operand` Wert als Objekt, das eine `value` Parameter mit der Regex-Syntax, die Sie abgleichen möchten, und einem `is_regex` Parameter, der auf gesetzt ist `true`. Die folgende Regel filtert DNS-Tunnelerkennungen mit Domainnamen, die mit enden `example.com`.

```
{
    "detection_type": "dns_tunnel",
    "expiration": null,
    "offender": "Any",
    "victim": "Any",
    "properties": [
        {
            "operand": {
                "value": ".*?example.com",
                "is_regex": true
            },
            "operator": "=",
            "property": "registered_domain_name"
        }
    ]
}
```

Groß- und Kleinschreibung deaktivieren

Sucht standardmäßig nach einer Zeichenfolge `property` Bei Werten wird zwischen Groß- und Kleinschreibung unterschieden. Sie können jedoch die Berücksichtigung von Groß- und Kleinschreibung deaktivieren, indem Sie den Operandenwert als Objekt angeben, das eine `case_sensitive` Parameter, der auf gesetzt ist `false`.

Die folgende Regel verbirgt Erkennungen von Hacking-Tool-Domänenzugriffen mit dem ArchStrike-Hacking-Tool.

```
{
    "detection_type": "hacking_tools",
    "expiration": null,
    "offender": "Any",
    "victim": "Any",
    "properties": [
        {
            "operand": {
                "value": "archstrike",
                "case_sensitive": false
            },
            "operator": "=",
            "property": "hacking_tool"
        }
    ]
}
```

Erkennungskategorien

Das Feld Kategorien ist ein Array, das in Antworten für zurückgegeben wird `GET /detections` und `POST /detections/search` Operationen. In der folgenden Tabelle sind gültige Einträge im Array aufgeführt:

Wert	Kategorie
`sec`	Sicherheit
`sec.action`	Zielgerichtete Maßnahmen
`sec.attack`	Attacke
`sec.botnet`	Botnetz
`sec.caution`	Vorsicht
`sec.command`	Befehl und Steuerung
`sec.cryptomining`	Krypto-Mining
`sec.dos`	Diensteverweigerung
`sec.exfil`	Exfiltration
`sec.exploit`	Ausbeutung
`sec.hardening`	Aushärten
`sec.lateral`	Seitliche Bewegung
`sec.ransomware`	Ransomware
`sec.recon`	Aufklärung
`perf`	Aufführung
`perf.auth`	Autorisierung und Zugriffskontrolle
`perf.db`	Datenbank
`perf.network`	Netzwerk-Infrastruktur
`perf.service`	Verschlechterung des Dienstes

Wert	Kategorie
`perf.storage`	Aufbewahrung
`perf.virtual`	Desktop- und Anwendungsvirtualisierung
`perf.web`	Web-Applikation

E-Mail-Gruppe

Sie können einzelne oder Gruppen-E-Mail-Adressen zu einer E-Mail-Gruppe hinzufügen und sie einem System zuweisen. Alarm. Wenn diese Alarm ausgelöst wird, sendet das System eine E-Mail an alle Adressen in der E-Mail-Gruppe.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /emailgroups	Rufen Sie alle E-Mail-Gruppen ab.
POST /emailgroups	Erstellen Sie eine neue E-Mail-Gruppe.
/emailgroups/ {id} LÖSCHEN	Löschen Sie eine E-Mail-Gruppe mit einer eindeutigen Kennung.
GET /emailgroups/ {id}	Rufen Sie eine bestimmte E-Mail-Gruppe anhand einer eindeutigen Kennung ab.
PATCH /emailgroups/ {id}	Wenden Sie Updates auf eine bestimmte E-Mail-Gruppe an.

Einzelheiten der Operation

`GET /emailgroups`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "email_addresses": [],
    "group_name": "string",
    "id": 0,
    "system_notifications": true
}
```

`POST /emailgroups`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die angegebenen Eigenschaftswerte auf die neue E-Mail-Gruppe an.

`group_name:` ***Schnur***

Der freundliche Name für die E-Mail-Gruppe.

`email_addresses:` ***Reihe von Zeichenketten***

Die Liste der E-Mail-Adressen in der E-Mail-Gruppe.

`system_notifications:` ***Boolescher Wert***

Gibt an, ob die Gruppe Systembenachrichtigungen erhalten soll.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "email_addresses": [],
    "group_name": "string",
    "system_notifications": true
}
```

`GET /emailgroups/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung der E-Mail-Gruppe.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "email_addresses": [],
    "group_name": "string",
    "id": 0,
    "system_notifications": true
}
```

`DELETE /emailgroups/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für die E-Mail-Gruppe.

`PATCH /emailgroups/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Wendet die angegebenen Eigenschaftswertaktualisierungen auf die E-Mail-Gruppe an.

`id:` ***Zahl***
Die eindeutige Kennung für die E-Mail-Gruppe.

Ausschlussintervalle

Ein Ausschlussintervall kann erstellt werden, um einen Zeitraum für die Unterdrückung eines Alarm.

Wenn Sie beispielsweise außerhalb der Geschäftszeiten oder am Wochenende nicht über Benachrichtigungen informiert werden möchten, kann ein Ausschlussintervall eine Regel erstellen, um die Alarm während dieses Zeitraums zu unterdrücken. Weitere Informationen finden Sie unter Warnmeldungen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /exclusioninterval	Ruft alle Ausschlussintervalle ab.
POST/Ausschlussintervalle	Erstellen Sie ein neues Ausschlussintervall.
LÖSCHEN /exclusionintervals/ {id}	Löscht ein bestimmtes Ausschlussintervall.

Betrieb	Beschreibung
GET /exclusionintervals/ {id}	Rufen Sie ein bestimmtes Ausschlussintervall ab.
PATCH /exclusionintervals/ {id}	Wenden Sie Updates für ein bestimmtes Ausschlussintervall an.

Einzelheiten der Operation

`GET /exclusionintervals`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "alert_apply_all": true,
    "author": "string",
    "description": "string",
    "end": 0,
    "id": 0,
    "interval_type": "string",
    "mod_time": 0,
    "name": "string",
    "start": 0,
    "trend_apply_all": true
}
```

`POST /exclusionintervals`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Legt die angegebenen Eigenschaftswerte für das neue Ausschlussintervall fest.

`name:` ***Schnur***

Der freundliche Name für das Ausschlussintervall.

`author:` ***Schnur***

(Optional) Der Name des Erstellers des Ausschlussintervalls.

`description:` ***Schnur***

(Optional) Eine optionale Beschreibung des Ausschlussintervalls.

`interval_type:` ***Schnur***

Das Zeitfenster, in dem das Ausschlussintervall ausgewertet wurde.

Die folgenden Werte sind gültig:

- `onetime`
- `weekly`
- `daily`

`start:` ***Zahl***

Der Beginn des Zeitbereichs für das Ausschlussintervall, ausgedrückt in Sekunden. Dieser Wert bezieht sich bei einmaligen Ausschlüssen auf die Epoche, bei täglichen Ausschlüssen auf Mitternacht und bei wöchentlichen Ausschlüssen auf Montag um Mitternacht.

`end:` ***Zahl***

Das Ende des Zeitbereichs für das Ausschlussintervall, ausgedrückt in Sekunden. Dieser Wert bezieht sich bei einmaligen Ausschlüssen auf die Epoche, bei täglichen Ausschlüssen auf Mitternacht und bei wöchentlichen Ausschlüssen auf Montag um Mitternacht.

`alert_apply_all:` ***Boolescher Wert***

Gibt an, ob dieses Ausschlussintervall auf alle Warnungen angewendet werden soll.

`trend_apply_all:` ***Boolescher Wert***

Gibt an, ob dieses Ausschlussintervall auf alle Trends angewendet werden soll.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "alert_apply_all": true,
    "author": "string",
    "description": "string",
    "end": 0,
    "interval_type": "string",
    "name": "string",
    "start": 0,
    "trend_apply_all": true
}
```

`GET /exclusionintervals/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung des Ausschlussintervalls.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "alert_apply_all": true,
    "author": "string",
    "description": "string",
    "end": 0,
    "id": 0,
    "interval_type": "string",
    "mod_time": 0,
    "name": "string",
    "start": 0,
    "trend_apply_all": true
}
```

`DELETE /exclusionintervals/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung des Ausschlussintervalls.

`PATCH /exclusionintervals/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die angegebenen Eigenschaftswertaktualisierungen auf das Ausschlussintervall an.

`id:` ***Zahl***

Die eindeutige Kennung für das Ausschlussintervall.

ExtraHop

Diese Ressource enthält Metadaten über das ExtraHop-System.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
HOLEN SIE SICH /extrahop	Ruft Metadaten über die Firmware ab, die auf dem ExtraHop-System läuft.
POST/extrahop/cloudresources	Aktualisieren Sie die Ressourcen auf dem ExtraHop-System manuell. Diese Ressourcen werden automatisch aktualisiert, wenn das System mit ExtraHop Cloud Services verbunden wird.
HOLEN SIE SICH /extrahop/cluster	Rufen Sie die Explore-Cluster-Konfigurationseinstellungen ab.
PATCH /extrahop/cluster	Aktualisieren Sie die Explore-Cluster-Konfigurationseinstellungen.
GET /extrahop/detections/access	Rufen Sie die Zugriffskontrolleinstellungen für Erkennungen ab.
PUT /extrahop/Erkennungen/Zugriff	Aktualisieren Sie die Zugriffskontrolleinstellungen für Erkennungen.
HOLEN SIE SICH /extrahop/edition	Ruft die Edition des ExtraHop-Systems ab. **Hinweis:** Für diesen Vorgang ist kein API-Schlüssel erforderlich.
POST/extrahop/firmware	Laden Sie ein neues Firmware-Image auf das ExtraHop-System hoch. Weitere Informationen finden Sie unter Aktualisieren Sie die ExtraHop-Firmware über die REST-API.
POST/extrahop/firmware/download/url	Laden Sie ein neues Firmware-Image von einer URL auf das ExtraHop-System herunter.
POST /extrahop/firmware/herunterladen/version	Laden Sie ein neues Firmware-Image von ExtraHop Cloud Services auf das ExtraHop-System herunter.
POST /extrahop/firmware/neuest/upgrade	Aktualisieren Sie das ExtraHop-System auf das zuletzt hochgeladene Firmware-Image.
GET /extrahop/firmware/next	Aktualisieren Sie das ExtraHop-System auf das zuletzt hochgeladene Firmware-Image.
GET /extrahop/firmware/previous	Rufen Sie Informationen über eine Firmware-Version ab, auf die Sie das ExtraHop-System zurücksetzen können.
POST /extrahop/firmware/previous/rollback	Setzen Sie das ExtraHop-System auf die vorherige Firmware-Version zurück.
HOLEN SIE SICH /extrahop/flowlogs/secret	Ruft das Flow-Log-Geheimnis ab.
POST /extrahop/flowlogs/secret	Generieren Sie ein neues Flow-Log-Geheimnis.
HOLEN SIE SICH /extrahop/idrac	Ruft die iDRAC-IP-Adresse des ExtraHop-Systems ab.

Bedienung	Beschreibung
GET /extrahop/platform	Ruft den Plattformnamen des ExtraHop-Systems ab. **Hinweis:** Für diesen Vorgang ist kein API-Schlüssel erforderlich.
GET /extrahop/processes	Ruft eine Liste der Prozesse ab, die auf dem ExtraHop-System ausgeführt werden.
POST /extrahop/processes/ {process} /restart	Starten Sie einen Prozess neu, der auf dem ExtraHop-System läuft.
GET /extrahop/services	Ruft die Einstellungen für alle Dienste ab.
PATCH /extrahop/services	Aktualisieren Sie die Einstellungen für Dienste.
POST /extrahop/restart	Starten Sie das ExtraHop-System neu.
POST /extrahop/shutdown	Fahren Sie das ExtraHop-System herunter.
POST/extrahop/sslcert	Generieren Sie das TLS-Zertifikat auf dem ExtraHop-System neu. Weitere Informationen finden Sie unter Erstellen Sie ein vertrauenswürdiges TLS-Zertifikat über die REST-API
PUT /extrahop/sslcert	Ersetzen Sie das TLS-Zertifikat auf dem ExtraHop-System.
POST /extrahop/sslcert/signingrequest	Erstellen Sie eine Anfrage zum Signieren eines TLS-Zertifikats. Weitere Informationen finden Sie unter Erstellen Sie ein vertrauenswürdiges TLS-Zertifikat über die REST-API.
GET /extrahop/ticketing	Rufen Sie den Status der Ticketing-Integration ab.
PATCH /extrahop/Ticketverkauf	Aktiviere oder deaktiviere die Ticketing-Integration.
HOLEN SIE SICH /extrahop/version	Rufen Sie die Version der Firmware ab, die auf dem ExtraHop-System läuft. **Hinweis:** Für diesen Vorgang ist kein API-Schlüssel erforderlich.

Einzelheiten der Operation

`GET /extrahop/version`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "version": "string"
}
```

`GET /extrahop/platform`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "platform": "string"
}
```

`GET /extrahop/edition`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "edition": "string"
}
```

`GET /extrahop`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "display_host": "string",
    "external_hostname": "string",
    "hostname": "string",
    "mgmt_ipaddr": "string",
    "platform": "string",
    "version": "string"
}
```

`GET /extrahop/idrac`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "ipaddr": "string"
}
```

`POST /extrahop/sslcert`

Für diesen Vorgang gibt es keine Parameter.

`PUT /extrahop/sslcert`

Geben Sie die folgenden Parameter an.

`body`: ***Schnur***

Das SSL-Zertifikat und optional der private Schlüssel. Geben Sie es als Klartext ein, getrennt durch einen Zeilenumbruch.

`POST /extrahop/sslcert/signingrequest`

Geben Sie die folgenden Parameter an.

`body`: ***Objekt***

Parameter für die Anforderung zum Signieren des SSL-Zertifikats.

`subject_alternative_names:` ***Reihe von Objekten***

Eine Liste von Namen, für die das Zertifikat gilt, z. B. {"type“: „dns“, „name“: „www.example.com"}.

`type:` ***Schnur***

Art des Betreffs Alternativer Name.

Die folgenden Werte sind gültig:

- `dns`
- `ip`

`name:` ***Schnur***

Name des Betreffs Alternativer Name.

`subject:` ***Objekt***

Der Betreff des SSL-Zertifikats. Eine Liste der Felder für Zertifikatsanträge finden Sie unten.

`common_name:` ***Schnur***

Der allgemeine Name (CN) des Subjekts.

`country_code:` ***Schnur***

(Optional) Das Betreff Land (C).

`state_or_province_name:` ***Schnur***

(Optional) Das betreffende Bundesland oder die Provinz (ST).

`locality_name:` ***Schnur***

(Optional) Die Lokalität des Betreffs (L).

`organization_name:` ***Schnur***

(Optional) Die Fachorganisation (O).

`organizational_unit_name:` ***Schnur***

(Optional) Die betreffende Organisationseinheit (OU).

`email_address:` ***Schnur***

(Optional) Die Betreff-E-Mail-Adresse (EmailAddress).

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "subject": {
        "common_name": "string",
        "country_code": "string",
        "state_or_province_name": "string",
        "locality_name": "string",
        "organization_name": "string",
        "organizational_unit_name": "string",
        "email_address": "string"
    },
    "subject_alternative_names": {
        "type": "string",
        "name": "string"
    }
}
```

`GET /extrahop/ticketing`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
```

```
    "enabled": true,
    "external_ticketing_enabled": true,
    "internal_ticketing_enabled": true,
    "url_template": "string"
}
```

`PATCH /extrahop/ticketing`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Einstellungen zur Ticketverfolgung.

`enabled:` ***Boolesch***

(Optional) Veraltet. Ersetzt durch die Felder external_ticketing_enabled und internal_ticketing_enabled.

`external_ticketing_enabled:` ***Boolesch***

(Optional) Gibt an, ob Erkennungen von einem externen Ticketsystem aus verfolgt werden. Dieses Feld ist erforderlich, wenn das Feld internal_ticketing_enabled angegeben ist.

`internal_ticketing_enabled:` ***Boolesch***

(Optional) Gibt an, ob Erkennungen innerhalb des ExtraHop-Systems verfolgt werden. Dieses Feld ist erforderlich, wenn das Feld external_ticketing_enabled angegeben ist.

`url_template:` ***Schnur***

(Optional) Die URL-Vorlage, die Erkennungen mit externen Tickets verknüpft. Die Vorlage muss die Variable $ticket_id enthalten. Dieses Feld gilt nur, wenn Erkennungen von einem externen Ticketsystem aus verfolgt werden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "enabled": true,
    "external_ticketing_enabled": true,
    "internal_ticketing_enabled": true,
    "url_template": "string"
}
```

`PUT /extrahop/detections/access`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Erkennungen greifen auf Einstellungen für die Appliance zu.

`enabled:` ***Boolesch***

Gibt an, ob die Einstellungen für den Erkennungszugriff aktiviert sind. Wenn diese Option aktiviert ist, können Administratoren den Erkennungszugriff für bestimmte Benutzer einschränken. Sie können die Einstellungen für den Erkennungszugriff nicht deaktivieren, nachdem die Einstellungen aktiviert wurden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "enabled": true
}
```

`GET /extrahop/detections/access`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "enabled": true
}
```

`POST /extrahop/firmware`

Geben Sie die folgenden Parameter an.

`firmware:` ***Dateiname***

Die .tar-Datei, die das Firmware-Image enthält. Hinweis: Sie können kein Firmware-Image über den REST-API-Explorer hochladen. Weitere Informationen zum Hochladen eines Bilds über cURL oder ein Python-Skript finden Sie unter Aktualisieren Sie die ExtraHop-Firmware über die REST-API.

`POST /extrahop/firmware/latest/upgrade`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

(Optional) Die Installationsoptionen für das Upgrade der Appliance.

`restart_after:` ***Boolesch***

(Optional) Gibt an, ob die Appliance nach Abschluss des Upgrades neu gestartet werden soll.

`silent:` ***Boolesch***

(Optional) Gibt an, ob die ExtraHop Web UI während des Upgrade-Vorgangs deaktiviert werden soll. Wenn ein Upgrade fehlschlägt, kehrt die Appliance automatisch zur vorherigen Firmware-Version zurück.

`force:` ***Boolesch***

(Optional) Gibt an, ob die Kompatibilitätsüberprüfung übersprungen werden soll. Überspringen Sie die Überprüfung nur, wenn der ExtraHop Support das Upgrade geprüft und genehmigt hat.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "force": true,
    "restart_after": true,
    "silent": true
}
```

`POST /extrahop/firmware/download/url`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Download-Optionen.

`firmware_url:` ***Schnur***

Die URL der Firmware, die heruntergeladen werden soll. HTTPS-, HTTP- und FTP-Schemata werden unterstützt.

`upgrade:` ***Boolesch***

(Optional) Gibt an, ob die Appliance aktualisiert werden soll, nachdem der Firmware-Download abgeschlossen ist.

`force:` ***Boolesch***

(Optional) Gibt an, ob die Kompatibilitätsüberprüfung übersprungen werden soll. Überspringen Sie die Überprüfung nur, wenn der ExtraHop Support das Upgrade geprüft und genehmigt hat.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "firmware_url": "string",
    "force": true,
    "upgrade": true
}
```

`POST /extrahop/restart`

Für diesen Vorgang gibt es keine Parameter.

`POST /extrahop/shutdown`

Für diesen Vorgang gibt es keine Parameter.

`GET /extrahop/services`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "admin": {
        "enabled": true
    },
    "keyreceiver": {
        "enabled": true
    },
    "snmp": {
        "enabled": true
    },
    "ssh": {
        "enabled": true
    }
}
```

`PATCH /extrahop/services`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Einstellungen für Dienste.

`admin:` ***Objekt***

(Optional) Die Einstellungen des Management-GUI-Dienstes, der den browserbasierten Zugriff auf die Appliance ermöglicht.

`enabled:` ***Boolesch***

Gibt an, ob der Dienst aktiviert ist.

`snmp:` ***Objekt***

(Optional) Die Einstellungen des SNMP-Dienstes, der es Ihrer Netzwerkgeräteüberwachungssoftware ermöglicht, Informationen aus dem ExtraHop-System zu sammeln.

`enabled:` ***Boolesch***

Gibt an, ob der Dienst aktiviert ist.

`ssh:` ***Objekt***

(Optional) Die Einstellungen des SSH-Dienstes, der es Benutzern ermöglicht, sich sicher an der ExtraHop-Befehlszeilenschnittstelle (CLI) anzumelden.

`enabled:` ***Boolesch***

Gibt an, ob der Dienst aktiviert ist.

`keyreceiver:` ***Objekt***

(Optional) Die Einstellungen des SSL-Sitzungsschlüsselempfängers, die es der Appliance ermöglichen, Sitzungsschlüssel von der Sitzungsschlüsselweiterleitung zu empfangen und zu entschlüsseln.

`enabled:` ***Boolesch***

Gibt an, ob der Dienst aktiviert ist.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "admin": {
        "enabled": true
    },
    "keyreceiver": {
        "enabled": true
    },
    "snmp": {
        "enabled": true
    },
    "ssh": {
        "enabled": true
    }
}
```

`GET /extrahop/processes`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "can_restart": true,
    "cpu": 0.0,
    "cpu_time": 0,
    "mem_percent": 0.0,
    "mem_res": 0,
    "mem_virt": 0,
    "process": "string",
    "start_time": 0
}
```

`POST /extrahop/processes/{process}/restart`

Geben Sie die folgenden Parameter an.

`process:` ***Schnur***

Der Name des Prozesses.

Die folgenden Werte sind gültig:

- `exadmin`

- exalerts
- examf
- exapi
- exbridge
- excap
- exconfig
- exflowlogs
- exsnmpq
- exnotify
- exportal
- exremote
- exsearch
- exstatmirror
- extrend
- webserver
- hopcloud-api

`GET /extrahop/cluster`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "ingest_enabled": true,
    "replication_policy": 0
}
```

`PATCH /extrahop/cluster`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die EXA-Cluster-Konfigurationseinstellungen.

`ingest_enabled:` ***Boolesch***

(Optional) Gibt an, ob die Datensatzaufnahme für den Explore-Cluster aktiviert ist.

`replication_policy:` ***Zahl***

(Optional) Die Replikationsstufe, die bestimmt, wie viele Kopien jedes Datensatz gespeichert werden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "ingest_enabled": true,
    "replication_policy": 0
}
```

`GET /extrahop/firmware/previous`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "backup_time": 0,
    "version": "string"
```

```
}
```

`POST /extrahop/firmware/previous/rollback`

Für diesen Vorgang gibt es keine Parameter.

`POST /extrahop/cloudresources`

Geben Sie die folgenden Parameter an.

`cloudresources:` ***Dateiname***
Die Ressourcenpaketdatei.

`GET /extrahop/flowlogs/secret`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "secret": "string"
}
```

`POST /extrahop/flowlogs/secret`

Für diesen Vorgang gibt es keine Parameter.

`GET /extrahop/firmware/next`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "current_release": true,
    "release": "string",
    "versions": []
}
```

`POST /extrahop/firmware/download/version`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
(Optional) Die Download-Optionen.

`version:` ***Schnur***
Die Version der Firmware, die heruntergeladen werden soll.

`upgrade:` ***Boolesch***
(Optional) Gibt an, ob die Appliance aktualisiert werden soll, nachdem der Firmware-Download abgeschlossen ist.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "upgrade": true,
    "version": "string"
}
```

Ermittlungen

Mithilfe von Untersuchungen können Sie mehrere Funde in einer einzigen Zeitleiste und Karte hinzufügen und anzeigen. Weitere Informationen finden Sie unter Ermittlungen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /untersuchungen	Rufen Sie alle Untersuchungen ab.
POST /Ermittlungen	Erstelle eine Untersuchung.
POST /investigations/search	Suchen Sie nach Ermittlungen.
LÖSCHE /investigations/ {id}	Löschen Sie eine bestimmte Untersuchung.
GET /investigations/ {id}	Rufen Sie eine bestimmte Untersuchung ab.
PATCH /investigations/ {id}	Aktualisiere eine Untersuchung.

Einzelheiten der Operation

`GET /investigations/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für die Untersuchung.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "assessment": "string",
    "assignee": "string",
    "created_by": "string",
    "creation_time": 0,
    "description": "string",
    "detections": [
        "string"
    ],
    "end_time": 0,
    "id": 0,
    "investigation_types": [
        "string"
    ],
    "is_user_created": true,
    "last_interaction_by": "string",
    "name": "string",
    "notes": "string",
    "start_time": 0,
    "status": "string",
    "update_time": 0,
    "url": "string"
}
```

`GET /investigations`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "assessment": "string",
    "assignee": "string",
    "created_by": "string",
    "creation_time": 0,
    "description": "string",
    "detections": [
        "string"
    ],
    "end_time": 0,
    "id": 0,
    "investigation_types": [
        "string"
    ],
    "is_user_created": true,
    "last_interaction_by": "string",
    "name": "string",
    "notes": "string",
    "start_time": 0,
    "status": "string",
    "update_time": 0,
    "url": "string"
}
```

`POST /investigations/search`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Parameter für die Untersuchung.

`update_time:` ***Zahl***

Gibt Untersuchungen zurück, die nach dem angegebenen Datum aktualisiert wurden, ausgedrückt in Millisekunden seit der Epoche.

`creation_time:` ***Zahl***

Gibt Untersuchungen zurück, die nach dem angegebenen Datum erstellt wurden, ausgedrückt in Millisekunden seit der Epoche.

`is_user_created:` ***Boolesch***

(Optional) Gibt nur Untersuchungen zurück, die manuell von einem Benutzer erstellt wurden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "creation_time": 0,
    "is_user_created": true,
    "update_time": 0
}
```

`PATCH /investigations/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die ID der Untersuchung, die aktualisiert werden soll.

`body:` ***Objekt***

Die zu aktualisierenden Untersuchungsfelder.

`name:` ***Schnur***

(Optional) Der Name der Untersuchung.

`status:` ***Schnur***

(Optional) Der Status der Untersuchung.

Die folgenden Werte sind gültig:

- `open`
- `in_progress`
- `closed`

`notes:` ***Schnur***

(Fakultativ) Optionale Hinweise zur Untersuchung.

`event_ids:` ***Reihe von Zahlen***

(Optional) Die Liste der IDs für Erkennungen in der Untersuchung. Wenn Sie dieses Feld angeben, ersetzt die neue Liste von IDs die bestehende Liste.

`assignee:` ***Schnur***

(Optional) Der Benutzername des mit der Untersuchung beauftragten Mitarbeiters.

`assessment:` ***Schnur***

(Fakultativ) Die Bewertung der Untersuchung.

Die folgenden Werte sind gültig:

- `malicious_true_positive`
- `benign_true_positive`
- `false_positive`
- `undecided`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assessment": "string",
    "assignee": "string",
    "event_ids": [],
    "name": "string",
    "notes": "string",
    "status": "string"
}
```

`POST /investigations`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Bereiche der neuen Untersuchung.

`name:` ***Schnur***

Der Name der Untersuchung.

`status:` ***Schnur***

(Optional) Der Status der Untersuchung.

Die folgenden Werte sind gültig:

- `open`
- `in_progress`
- `closed`

`notes:` ***Schnur***

(Fakultativ) Optionale Hinweise zur Untersuchung.

`event_ids:` ***Reihe von Zahlen***

(Optional) Die Liste der IDs für Erkennungen in der Untersuchung.

`assignee:` ***Schnur***

(Optional) Der Benutzername des mit der Untersuchung beauftragten Mitarbeiters.

`assessment:` ***Schnur***

(Fakultativ) Die Bewertung der Untersuchung.

Die folgenden Werte sind gültig:

- `malicious_true_positive`
- `benign_true_positive`
- `false_positive`
- `undecided`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assessment": "string",
    "assignee": "string",
    "event_ids": [],
    "name": "string",
    "notes": "string",
    "status": "string"
}
```

`DELETE /investigations/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die ID der zu löschenden Untersuchung.

Jobs

Sie können den Fortschritt einiger Verwaltungsaufgaben überwachen, die über die REST-API gestartet wurden. Wenn eine REST-Anfrage einen Job erstellt, wird die Job-ID zurückgegeben in `location` Header der Antwort. Die folgenden Operationen schaffen Arbeitsplätze:

- `POST /extrahop/firmware/latest/upgrade`
- `POST /extrahop/sslcert`

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /jobs	Ruft den Status aller Jobs ab.
GET /jobs/ {id}	Rufen Sie den Status eines bestimmten Jobs ab.

Einzelheiten der Operation

`GET /jobs/{id}`

Geben Sie die folgenden Parameter an.

`id`: ***Schnur***

Die eindeutige Kennung für den Job.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "details": "string",
    "id": "string",
    "remote_jobs": [],
    "status": "string",
    "step_description": "string",
    "step_number": 0,
    "total_steps": 0,
    "type": "string"
}
```

`GET /jobs`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "details": "string",
    "id": "string",
    "remote_jobs": [],
    "status": "string",
    "step_description": "string",
    "step_number": 0,
    "total_steps": 0,
    "type": "string"
}
```

Arten von Aufträgen

Die `GET /jobs` Operation gibt die folgenden Werte zurück in `type` Feld der Antwort.

extrahop_firmware_herunterladen

Das ExtraHop-System lädt ein neues Firmware-Image entweder von einer URL oder von ExtraHop Cloud Services herunter.

extrahop_firmware_upgrade

Das ExtraHop-System wird auf eine neue Firmware-Version aktualisiert.

extrahop_firmware_download_upgrade

Das ExtraHop-System lädt ein Firmware-Image herunter und aktualisiert auf eine neue Firmware-Version. Das Bild wird entweder von einer URL oder von ExtraHop Cloud Services abgerufen.

Hinweis: Die `type` Das Feld ist für einige Jobs leer.

Lizenz

Diese Ressource ermöglicht es Ihnen, Produktschlüssel abzurufen und festzulegen oder eine Lizenz abzurufen und festzulegen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /license	Rufen Sie die Lizenz ab, die auf dieses ExtraHop-System angewendet wurde.
PUT /Lizenz	Beantragen und registrieren Sie eine neue Lizenz für das ExtraHop-System.
HOLEN SIE SICH /license/productkey	Rufen Sie den Produktschlüssel für dieses ExtraHop-System ab.
PUT /license/productkey	Wenden Sie den angegebenen Produktschlüssel auf das ExtraHop-System an und registrieren Sie die Lizenz.

Einzelheiten der Operation

`PUT /license`

Geben Sie die folgenden Parameter an.

`body:` ***Schnur***

(Optional) Der Lizenztext, der Ihnen vom ExtraHop Support zur Verfügung gestellt wurde, einschließlich der BEGIN- und END-Zeilen.

`GET /license`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "dossier": "string",
    "edition": "string",
    "expires_at": 0,
    "expires_in": 0,
    "modules": {},
    "options": {},
    "platform": "string",
    "product_key": "string",
    "serial": "string"
}
```

`PUT /license/productkey`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

(Optional) Wenden Sie den angegebenen Produktschlüssel auf die Appliance an.

`GET /license/productkey`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "product_key": "string"
}
```

Metriken

Zu jedem Objekt, das vom ExtraHop-System identifiziert wird, werden Metrikinformationen gesammelt.

Beachten Sie, dass Metriken über die POST-Methode abgerufen werden, die eine Abfrage erstellt, um die angeforderten Informationen über die API zu sammeln. Weitere Informationen finden Sie unter Extrahieren Sie Metriken über die REST-API.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
POST /Metriken	Ruft Metriken für jedes angegebene Objekt ab.
GET /metrics/next/ {xid}	Wenn Sie Metriken von einem anfordern Konsole mit dem `POST /metrics`, `POST /metrics/total`, oder `POST /metrics/totalbyobject` Operation, und Sie geben Objekte an, die von mehreren Sensoren beobachtet wurden, die Antwort enthält `xid` Feld statt Metrik Daten. Sie können Metrikdaten abrufen, indem Sie Folgendes angeben `xid` Feld in der `GET /metrics/next/{xid}` Operation, die Metriken von einem der an die Konsole angeschlossenen Sensoren zurückgibt. Wiederhole das `GET /metrics/next/{xid}` Betrieb, um Metriken von zusätzlichen Sensoren zurückzugeben. Nachdem alle Metriken abgerufen wurden, gibt der Vorgang Null zurück. Wenn vom Sensor noch keine Metriken verfügbar sind, wird die Zeichenfolge `again` wird zurückgegeben. Warten Sie ein paar Sekunden und versuchen Sie es dann erneut. **Hinweis:** Die Antwort könnte eine enthalten `xid` Feld, auch wenn Sie nur Messwerte für eine einzelne Gerätegruppe angefordert haben, da Gerätegruppen Geräte von mehreren Sensoren enthalten können.
POST /Metriken/insgesamt	Ruft kombinierte Metriksummen für alle angegebenen Objekte ab.
POST /metrics/totalbyobject	Ruft Metriksummen für jedes angegebene Objekt ab.

Der folgende Anforderungstext ruft beispielsweise HTTP-Antworten ab, die zwei Geräte in den letzten 30 Minuten gesendet haben.

```
{
  "cycle": "auto",
  "from": -1800000,
  "metric_category": "http_server",
  "metric_specs": [
    {
      "name": "rsp"
    }
  ],
  "object_ids": [
```

```
    180, 177
  ],
  "object_type": "device",
  "until": 0
}
```

Für die `POST /metrics` Operation, der vorherige Beispielanforderungstext gibt die Anzahl der HTTP-Antworten zurück, die in jedem Zeitintervall aufgetreten sind. Sie sind mit der Uhrzeit jedes Ereignis und der ID des Gerät, das die Antworten gesendet hat, beschriftet, ähnlich der folgenden Beispielantwort:

```
{
  "cycle": "30sec",
  "node_id": 0,
  "clock": 1709659320000,
  "from": 1709657520000,
  "until": 1709659320000,
  "stats": [
    {
      "oid": 177,
      "time": 1709657520000,
      "duration": 30000,
      "values": [
        4
      ]
    },
    {
      "oid": 177,
      "time": 1709657550000,
      "duration": 30000,
      "values": [
        4
      ]
    },
    {
      "oid": 180,
      "time": 1709657520000,
      "duration": 30000,
      "values": [
        4
      ]
    },
    {
      "oid": 180,
      "time": 1709657550000,
      "duration": 30000,
      "values": [
        4
      ]
    }
  ]
}
```

Für die `POST /metrics/totalbyobject` Operation, derselbe vorherige Beispielanforderungstext ruft die Gesamtsumme für jedes Gerät über den gesamten Zeitraum ab, ähnlich der folgenden Beispielantwort:

```
{
  "cycle": "30sec",
  "node_id": 0,
  "clock": 1709659620000,
  "from": 1709657820000,
  "until": 1709659620000,
  "stats": [
```

```
    {
      "oid": 180,
      "time": 1709659620000,
      "duration": 1830000,
      "values": [
        8
      ]
    },
    {
      "oid": 177,
      "time": 1709659620000,
      "duration": 1830000,
      "values": [
        8
      ]
    }
  ]
}
```

Für die `POST /metrics/total` Operation, derselbe vorherige Beispiel-Anforderungstext ruft die Gesamtsumme beider Geräte über den gesamten Zeitraum ab, ähnlich der folgenden Beispielantwort:

```
{
  "cycle": "30sec",
  "node_id": 0,
  "clock": 1709659830000,
  "from": 1709658030000,
  "until": 1709659830000,
  "stats": [
    {
      "oid": -1,
      "time": 1709659830000,
      "duration": 1830000,
      "values": [
        16
      ]
    }
  ]
}
```

Beachten Sie, dass das Verhalten des `/metrics/total` und `/metrics/totalbyobject` Endpunkte hängen vom Typ der Metrik ab. Für Zählmetriken ist der `values` Das Feld enthält eine Gesamtsumme der Werte über das angegebene Zeitintervall, wie im obigen Beispiel gezeigt. Für Datensatzmetriken ist jedoch `values` Das Feld enthält eine Liste von Werten und die Häufigkeit, mit der diese Werte auftauchten. Zum Beispiel eine Abfrage nach Serververarbeitungszeiten mit dem `POST /metrics/total` operation gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt:

```
{
  "cycle": "30sec",
  "node_id": 0,
  "clock": 1494541440000,
  "from": 1494539640000,
  "until": 1494541440000,
  "stats": [
    {
      "oid": -1,
      "time": 1494541380000,
      "duration": 1800000,
      "values": [
        [
          {
            "value": 2.271,
```

```
            "freq": 5
          },
          {
            "value": 48.903,
            "freq": 1
          }
        ]
      ]
    }
  ]
}
```

Wenn im angegebenen Zeitraum mehr als 1.000 unterschiedliche Datensatzwerte vorliegen, werden ähnliche Werte konsolidiert, um die Antwortvariablen auf 1.000 Werte zu reduzieren. Wenn es beispielsweise weniger als 1.000 Werte gibt, kann die Antwort die folgenden Einträge enthalten:

```
{
  "value": 2.571,
  "freq": 4
},
{
  "value": 2.912,
  "freq": 2
}
```

Wenn die Antwort jedoch mehr als 1.000 Werte enthält, können diese Einträge zu dem folgenden Eintrag konsolidiert werden:

```
{
  "value": 2.571,
  "freq": 6
}
```

Wenn der `calc_type` Ein Feld ist angegeben und die Antwort enthält mehr als 1.000 Werte. Das Perzentil oder der Mittelwert wird anhand des konsolidierten Datensatzes berechnet.

Einzelheiten der Operation

`POST /metrics`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Beschreibung der Metrikanforderung.

`from:` ***Zahl***

Der Anfangszeitstempel für die Anfrage. Gibt nur Metriken zurück, die nach dieser Zeit erfasst wurden. Die Zeit wird in Millisekunden seit der Epoche ausgedrückt. 0 gibt den Zeitpunkt der Anfrage an. Ein negativer Wert wird relativ zur aktuellen Uhrzeit ausgewertet. Die Standardeinheit für einen negativen Wert ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`until:` ***Zahl***

Der Endzeitstempel für die Anfrage. Gibt nur Metriken zurück, die vor diesem Zeitpunkt erfasst wurden. Folgt den gleichen Zeitwertrichtlinien wie der From Parameter.

`cycle:` ***Schnur***

Der Aggregationszeitraum für Metriken.

Die folgenden Werte sind gültig:

- `auto`
- `1sec`
- `30sec`
- `5min`
- `1hr`
- `24hr`

`object_type:` ***Schnur***

Gibt den Objekttyp der eindeutigen Bezeichner an, die in der Eigenschaft object_ids angegeben sind.

Die folgenden Werte sind gültig:

- `network`
- `device`
- `application`
- `vlan`
- `device_group`
- `system`

`object_ids:` ***Reihe von Zahlen***

Die Liste der numerischen Werte, die eindeutige Identifikatoren darstellen. Eindeutige Identifikatoren können über die Ressourcen /networks, /devices, /applications, /vlans, /devicegroups, /activitygroups und /appliances abgerufen werden. Geben Sie für Systemintegritätsmetriken die ID des Sensor oder der Konsole an und setzen Sie den Parameter object_type auf „system“.

`metric_category:` ***Schnur***

Die Gruppe von Metriken, die im Metrikkatalog durchsucht werden können.

`metric_specs:` ***Reihe von Objekten***

Ein Array von Metrik Spezifikationsobjekten.

`name:` ***Schnur***

Der Feldname für die Metrik. Wenn Sie im Metrikkatalog nach einer metric_category filtern, ist jedes Ergebnis ein potenzieller metric_spec-Name. Wenn ein Ergebnis aus dem Katalog ausgewählt wird, ist der Feldwert „Metrik“ eine gültige Option für dieses Feld.

`key1:` ***Schnur***

(Optional) Filtern Sie Detailmetriken. Detailmetriken unterteilen Daten anhand von Schlüsseln, bei denen es sich um Zeichenketten oder IP-Adressen handelt. Beispielsweise akzeptiert die Metrik „HTTP Requests by Method“ den key1-Wert „GET“. Schlüssel können auch reguläre Ausdrücke sein, die durch Schrägstriche („/GET/“) getrennt sind.

`key2:` ***Schnur***

(Optional) Aktivieren Sie zusätzliche Filterung für Detailmetriken.

`calc_type:` ***Schnur***

(Optional) Die Art der auszuführenden Berechnung.

Die folgenden Werte sind gültig:

- `mean`
- `percentiles`

`percentiles:` ***Reihe von Zahlen***

(Optional) Die in aufsteigender Reihenfolge sortierte Liste der Perzentile, die zurückgegeben werden sollen. Dieser Parameter ist nur erforderlich, wenn der

Parameter calc_type auf „Perzentile“ gesetzt ist. Wenn der Parameter calc_type auf mean gesetzt ist, kann die Percentile-Eigenschaft nicht festgelegt werden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "cycle": "string",
    "from": 0,
    "metric_category": "string",
    "metric_specs": {
        "name": "string",
        "key1": "string",
        "key2": "string",
        "calc_type": "string",
        "percentiles": []
    },
    "object_ids": [],
    "object_type": "string",
    "until": 0
}
```

`POST /metrics/total`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Beschreibung der Metrikanforderung.

`from:` ***Zahl***

Der Anfangszeitstempel für die Anfrage. Gibt nur Metriken zurück, die nach dieser Zeit erfasst wurden. Die Zeit wird in Millisekunden seit der Epoche ausgedrückt. 0 gibt den Zeitpunkt der Anfrage an. Ein negativer Wert wird relativ zur aktuellen Uhrzeit ausgewertet. Die Standardeinheit für einen negativen Wert ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`until:` ***Zahl***

Der Endzeitstempel für die Anfrage. Gibt nur Metriken zurück, die vor diesem Zeitpunkt erfasst wurden. Folgt den gleichen Zeitwertrichtlinien wie der From Parameter.

`cycle:` ***Schnur***

Der Aggregationszeitraum für Metriken.

Die folgenden Werte sind gültig:

- `auto`
- `1sec`
- `30sec`
- `5min`
- `1hr`
- `24hr`

`object_type:` ***Schnur***

Gibt den Objekttyp der eindeutigen Bezeichner an, die in der Eigenschaft object_ids angegeben sind.

Die folgenden Werte sind gültig:

- `network`
- `device`
- `application`

- `vlan`
- `device_group`
- `system`

`object_ids:` ***Reihe von Zahlen***

Die Liste der numerischen Werte, die eindeutige Identifikatoren darstellen. Eindeutige Identifikatoren können über die Ressourcen /networks, /devices, /applications, /vlans, /devicegroups, /activitygroups und /appliances abgerufen werden. Geben Sie für Systemintegritätsmetriken die ID des Sensor oder der Konsole an und setzen Sie den Parameter object_type auf „system“.

`metric_category:` ***Schnur***

Die Gruppe von Metriken, die im Metrikkatalog durchsucht werden können.

`metric_specs:` ***Reihe von Objekten***

Ein Array von Metrik Spezifikationsobjekten.

`name:` ***Schnur***

Der Feldname für die Metrik. Wenn Sie im Metrikkatalog nach einer metric_category filtern, ist jedes Ergebnis ein potenzieller metric_spec-Name. Wenn ein Ergebnis aus dem Katalog ausgewählt wird, ist der Feldwert „Metrik“ eine gültige Option für dieses Feld.

`key1:` ***Schnur***

(Optional) Filtern Sie Detailmetriken. Detailmetriken unterteilen Daten anhand von Schlüsseln, bei denen es sich um Zeichenketten oder IP-Adressen handelt. Beispielsweise akzeptiert die Metrik „HTTP Requests by Method“ den key1-Wert „GET“. Schlüssel können auch reguläre Ausdrücke sein, die durch Schrägstriche („/GET/“) getrennt sind.

`key2:` ***Schnur***

(Optional) Aktivieren Sie zusätzliche Filterung für Detailmetriken.

`calc_type:` ***Schnur***

(Optional) Die Art der auszuführenden Berechnung.

Die folgenden Werte sind gültig:

- `mean`
- `percentiles`

`percentiles:` ***Reihe von Zahlen***

(Optional) Die in aufsteigender Reihenfolge sortierte Liste der Perzentile, die zurückgegeben werden sollen. Dieser Parameter ist nur erforderlich, wenn der Parameter calc_type auf „Perzentile“ gesetzt ist. Wenn der Parameter calc_type auf mean gesetzt ist, kann die Percentile-Eigenschaft nicht festgelegt werden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "cycle": "string",
    "from": 0,
    "metric_category": "string",
    "metric_specs": {
        "name": "string",
        "key1": "string",
        "key2": "string",
        "calc_type": "string",
        "percentiles": []
    },
    "object_ids": [],
    "object_type": "string",
    "until": 0
```

```
}
```

`POST /metrics/totalbyobject`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Beschreibung der Metrikanforderung.

`from:` ***Zahl***

Der Anfangszeitstempel für die Anfrage. Gibt nur Metriken zurück, die nach dieser Zeit erfasst wurden. Die Zeit wird in Millisekunden seit der Epoche ausgedrückt. 0 gibt den Zeitpunkt der Anfrage an. Ein negativer Wert wird relativ zur aktuellen Uhrzeit ausgewertet. Die Standardeinheit für einen negativen Wert ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`until:` ***Zahl***

Der Endzeitstempel für die Anfrage. Gibt nur Metriken zurück, die vor diesem Zeitpunkt erfasst wurden. Folgt den gleichen Zeitwertrichtlinien wie der From Parameter.

`cycle:` ***Schnur***

Der Aggregationszeitraum für Metriken.

Die folgenden Werte sind gültig:

- `auto`
- `1sec`
- `30sec`
- `5min`
- `1hr`
- `24hr`

`object_type:` ***Schnur***

Gibt den Objekttyp der eindeutigen Bezeichner an, die in der Eigenschaft object_ids angegeben sind.

Die folgenden Werte sind gültig:

- `network`
- `device`
- `application`
- `vlan`
- `device_group`
- `system`

`object_ids:` ***Reihe von Zahlen***

Die Liste der numerischen Werte, die eindeutige Identifikatoren darstellen. Eindeutige Identifikatoren können über die Ressourcen /networks, /devices, /applications, /vlans, /devicegroups, /activitygroups und /appliances abgerufen werden. Geben Sie für Systemintegritätsmetriken die ID des Sensor oder der Konsole an und setzen Sie den Parameter object_type auf „system“.

`metric_category:` ***Schnur***

Die Gruppe von Metriken, die im Metrikkatalog durchsucht werden können.

`metric_specs:` ***Reihe von Objekten***

Ein Array von Metrik Spezifikationsobjekten.

`name:` ***Schnur***

Der Feldname für die Metrik. Wenn Sie im Metrikkatalog nach einer metric_category filtern, ist jedes Ergebnis ein potenzieller metric_spec-Name. Wenn ein Ergebnis aus dem Katalog ausgewählt wird, ist der Feldwert „Metrik“ eine gültige Option für dieses Feld.

`key1:` ***Schnur***

(Optional) Filtern Sie Detailmetriken. Detailmetriken unterteilen Daten anhand von Schlüsseln, bei denen es sich um Zeichenketten oder IP-Adressen handelt. Beispielsweise akzeptiert die Metrik „HTTP Requests by Method“ den key1-Wert „GET“. Schlüssel können auch reguläre Ausdrücke sein, die durch Schrägstriche („/GET/“) getrennt sind.

`key2:` ***Schnur***

(Optional) Aktivieren Sie zusätzliche Filterung für Detailmetriken.

`calc_type:` ***Schnur***

(Optional) Die Art der auszuführenden Berechnung.

Die folgenden Werte sind gültig:

- `mean`
- `percentiles`

`percentiles:` ***Reihe von Zahlen***

(Optional) Die in aufsteigender Reihenfolge sortierte Liste der Perzentile, die zurückgegeben werden sollen. Dieser Parameter ist nur erforderlich, wenn der Parameter calc_type auf „Perzentile“ gesetzt ist. Wenn der Parameter calc_type auf mean gesetzt ist, kann die Percentile-Eigenschaft nicht festgelegt werden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "cycle": "string",
    "from": 0,
    "metric_category": "string",
    "metric_specs": {
        "name": "string",
        "key1": "string",
        "key2": "string",
        "calc_type": "string",
        "percentiles": []
    },
    "object_ids": [],
    "object_type": "string",
    "until": 0
}
```

`GET /metrics/next/{xid}`

Geben Sie die folgenden Parameter an.

`xid:` ***Zahl***

Der eindeutige Bezeichner, der von einer Metrikabfrage zurückgegeben wird.

Unterstützte Zeiteinheiten

Für die meisten Parameter ist die Standardeinheit für die Zeitmessung Millisekunden. Die folgenden Parameter geben jedoch alternative Zeiteinheiten wie Minuten und Stunden zurück oder akzeptieren diese:

- Gerät

    - aktive_von
    - aktiv_bis
- Gerätegruppe
    - aktive_von
    - aktiv_bis
- Metriken
    - von
    - bis
- Protokoll aufzeichnen
    - von
    - bis
    - kontext_ttl

Die folgende Tabelle zeigt die unterstützten Zeiteinheiten:

Zeiteinheit	Einheitensuffix
Jahr	`y`
Monat	`M`
Woche	`w`
Tag	`d`
Stunde	`h`
Minute	`m`
Zweiter	`s`
Millisekunde	`ms`

Um für einen Parameter eine andere Zeiteinheit als Millisekunden anzugeben, hängen Sie das Einheitensuffix an den Wert an. Um beispielsweise Geräte anzufordern, die in den letzten 30 Minuten aktiv waren, geben Sie den folgenden Parameterwert an:

```
GET /api/v1/devices?active_from=-30m
```

Das folgende Beispiel spezifiziert eine Suche nach HTTP Datensätze, die vor 1 bis 2 Stunden erstellt wurden:

```
{
    "from": "-2h",
    "until": "-1h",
    "types": ["~http"]
}
```

Netzwerk

Netzwerke sind mit der Netzwerkschnittstellenkarte korreliert, die Eingaben von allen vom ExtraHop-System identifizierten Objekten empfängt.

Auf einem Konsole, jeder angeschlossene Sensor wird als Netzwerkaufnahme identifiziert. Weitere Informationen finden Sie unter Netzwerke.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /netzwerke	Ruft alle Netzwerke ab.
GET /networks/ {id}	Ruft ein bestimmtes Netzwerk anhand der ID ab.
PATCH /Netzwerke/ {id}	Aktualisieren Sie ein bestimmtes Netzwerk anhand der ID.
GET /networks/ {id} /alerts	Alles abrufen Warnungen die einem bestimmten Netzwerk zugewiesen sind.
POST /networks/ {id} /alerts	Weisen Sie Alerts einem bestimmten Netzwerk zu und heben Sie die Zuweisung auf.
LÖSCHEN Sie /networks/ {id} /alerts/ {child-id}	Heben Sie die Zuweisung einer Alarm zu einem bestimmten Netzwerk auf.
POST /networks/ {id} /alerts/ {child-id}	Weisen Sie einem bestimmten Netzwerk eine Alarm zu.
GET /networks/ {id} /vlans	Ruft alle VLANS ab, die einem bestimmten Netzwerk zugewiesen sind.

Einzelheiten der Operation

`GET /networks`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "appliance_uuid": "string",
    "description": "string",
    "id": 0,
    "idle": true,
    "mod_time": 0,
    "name": "string",
    "node_id": 0
}
```

`PATCH /networks/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Eigenschaftswertaktualisierungen, die auf das Netzwerk angewendet werden sollen.

`id:` ***Zahl***
Eindeutige Kennung des Netzwerk.

`GET /networks/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Eindeutige Kennung des Netzwerk.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "appliance_uuid": "string",
    "description": "string",
    "id": 0,
    "idle": true,
    "mod_time": 0,
    "name": "string",
    "node_id": 0
}
```

`GET /networks/{id}/alerts`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Eindeutige Kennung des Netzwerk.

`direct_assignments_only:` ***Boolescher Wert***
(Optional) Beschränken Sie die Ergebnisse auf Warnmeldungen, die dem Netzwerk direkt zugewiesen sind.

`POST /networks/{id}/alerts`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Listen von Alert-IDs, die zugewiesen und/oder aufgehoben werden sollen.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Eindeutige Kennung des Netzwerk.

`POST /networks/{id}/alerts/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Eindeutige Kennung der Alarm.

`id:` ***Zahl***
Eindeutige Kennung des Netzwerk.

`DELETE /networks/{id}/alerts/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Eindeutige Kennung der Alarm.

`id:` ***Zahl***
Eindeutige Kennung des Netzwerk.

`GET /networks/{id}/vlans`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Eindeutige Kennung des Netzwerk.

Netzwerk-Lokalitätseintrag

Sie können eine Liste verwalten, die die Netzwerklokalität von IP-Adressen angibt.

Sie können beispielsweise einen Eintrag in der Netzwerklokalitätsliste erstellen, der angibt, dass eine IP-Adresse oder ein CIDR-Block intern oder extern ist.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /networklocations	Ruft alle Netzwerklokalitätseinträge ab.
POST /Netzwerkorte	Erstellen Sie einen Netzwerklokalitätseintrag.
LÖSCHE /networklocalities/ {id}	Löscht einen Netzwerklokalitätseintrag. **Hinweis** Dieser Vorgang ist bei Sensoren, die an RevealX 360 angeschlossen sind, nicht verfügbar. Diese Operation ist jedoch verfügbar in der RevealX 360 REST-API.
GET /networklocalities/ {id}	Ruft einen bestimmten Netzwerklokalitätseintrag ab.
PATCH /networklocalities/ {id}	Wenden Sie Aktualisierungen auf einen bestimmten Netzwerklokalitätseintrag an. **Hinweis** Dieser Vorgang ist bei Sensoren, die an RevealX 360 angeschlossen sind, nicht verfügbar. Diese Operation ist jedoch verfügbar in der RevealX 360 REST-API.

Einzelheiten der Operation

`GET /networklocalities`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "external": true,
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "network": "string",
    "networks": []
```

```
}
```

`POST /networklocalities`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die angegebenen Eigenschaftswerte auf den neuen Eintrag für die Netzwerklokalität an.

`name:` ***Schnur***

(Optional) Der Name der Netzwerklokalität. Wenn dieses Feld nicht angegeben ist, wird die Netzwerklokalität im folgenden Format benannt: „Locality_ID“, wobei ID die eindeutige Kennung der Netzwerklokalität ist.

`network:` ***Schnur***

(Optional) Veraltet. Geben Sie CIDR-Blöcke oder IP-Adressen im Feld Netzwerke an.

`networks:` ***Reihe von Zeichenketten***

(Optional) Eine Reihe von CIDR-Blöcken oder IP-Adressen, die die Netzwerklokalität definieren.

`external:` ***Boolescher Wert***

Gibt an, ob das Netzwerk intern oder extern ist.

`description:` ***Schnur***

(Optional) Eine optionale Beschreibung des Eintrags zur Netzwerklokalität.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "external": true,
    "name": "string",
    "network": "string",
    "networks": []
}
```

`GET /networklocalities/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für den Eintrag zur Netzwerklokalität.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "external": true,
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "network": "string",
    "networks": []
}
```

`DELETE /networklocalities/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für den Eintrag zur Netzwerklokalität.

`PATCH /networklocalities/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die angegebenen Eigenschaftswertaktualisierungen auf den Eintrag für die Netzwerklokalität an.

`network:` ***Schnur***

(Optional) Veraltet. Geben Sie CIDR-Blöcke oder IP-Adressen im Feld Netzwerke an.

`networks:` ***Reihe von Zeichenketten***

(Optional) Eine Reihe von CIDR-Blöcken oder IP-Adressen, die die Netzwerklokalität definieren.

`name:` ***Schnur***

(Optional) Der Name der Netzwerklokalität.

`external:` ***Boolescher Wert***

(Optional) Gibt an, ob das Netzwerk intern oder extern ist.

`description:` ***Schnur***

(Optional) Eine optionale Beschreibung des Eintrags zur Netzwerklokalität.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "description": "string",
    "external": true,
    "name": "string",
    "network": "string",
    "networks": []
}
```

`id:` ***Zahl***

Die eindeutige Kennung für den Eintrag zur Netzwerklokalität.

Knoten

Ein Knoten ist ein Sensor das ist verbunden mit einem Konsole.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /nodes	Alles abrufen Sensoren verbunden damit Konsole.
GET /nodes/ {id}	Rufen Sie ein bestimmtes ab Sensor das ist damit verbunden Konsole.
PATCH /nodes/ {id}	Aktualisieren Sie ein bestimmtes Sensor das ist damit verbunden Konsole.

Einzelheiten der Operation

`GET /nodes`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "add_time": 0,
    "display_name": "string",
    "enabled": true,
    "firmware_version": "string",
    "hostname": "string",
    "id": 0,
    "license_status": "string",
    "nickname": "string",
    "ntp_sync": true,
    "product_key": "string",
    "status_code": "string",
    "status_message": "string",
    "time_added": 0,
    "time_offset": 0,
    "uuid": "string"
}
```

`GET /nodes/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die ID des Sensor.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "add_time": 0,
    "display_name": "string",
    "enabled": true,
    "firmware_version": "string",
    "hostname": "string",
    "id": 0,
    "license_status": "string",
    "nickname": "string",
    "ntp_sync": true,
    "product_key": "string",
    "status_code": "string",
    "status_message": "string",
    "time_added": 0,
    "time_offset": 0,
    "uuid": "string"
}
```

`PATCH /nodes/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wenden Sie die angegebenen Updates auf den Discover-Knoten an.

`id:` ***Zahl***

Der eindeutige Bezeichner für den Discover-Knoten.

Beobachtungen

Eine Beobachtung verknüpft die IP-Adresse eines Gerät auf dem ExtraHop-System mit einer IP-Adresse außerhalb Ihres Netzwerk. Sie können beispielsweise die Aktivität eines VPN-Benutzers verfolgen, indem Sie die IP-Adresse des VPN-Clients in Ihrem internen Netzwerk mit der externen IP-Adresse verknüpfen, die dem Benutzer im öffentlichen Internet zugewiesen wurde.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
BEITRAG /observations/associatedipaddrs	Fügen Sie eine Beobachtung hinzu, um eine Zuordnung zwischen Geräte-IP-Adressen herzustellen.

Einzelheiten der Operation

```
POST /observations/associatedipaddrs
```

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Beobachtungsparameter.

`observations:` ***Reihe von Objekten***

Eine Reihe von Beobachtungen.

`ipaddr:` ***Schnur***

Die vom Sensor oder der Konsole beobachtete Geräte-IP-Adresse.

`associated_ipaddr:` ***Schnur***

Die zugehörige IP-Adresse.

`timestamp:` ***Zahl***

Die Zeit, in der die Beobachtung von der Quelle erstellt wurde, ausgedrückt in Millisekunden seit der Epoche.

`source:` ***Schnur***

Die Quelle der Beobachtungen.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "observations": {
        "ipaddr": "string",
        "associated_ipaddr": "string",
        "timestamp": 0
    },
    "source": "string"
}
```

Datenstrom öffnen

Ein offener Datenstrom (ODS) ist ein Kanal, über den Sie bestimmte Metrik Daten von einem senden können Sensor an ein externes System eines Drittanbieters. Möglicherweise möchten Sie Metrikdaten mit einem Remote-Tool wie Splunk, MongoDB oder Amazon Web Services (AWS) speichern oder analysieren.

Das Senden von Daten über einen offenen Datenstrom ist ein zweistufiges Verfahren. Zunächst konfigurieren Sie eine Verbindung zum Zielsystem, das die Daten empfängt. Zweitens schreiben Sie einen Auslöser, der festlegt, welche Daten an das Zielsystem gesendet werden sollen und wann sie gesendet werden sollen. Weitere Informationen finden Sie unter Offene Datenströme.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /odstargets	Ruft alle Open Data Stream-Ziele ab.
GET /odstargets/http	Ruft alle HTTP Open Data Stream-Ziele ab.
BEITRAG /odstargets/http	Erstellen Sie ein neues HTTP Open Data Stream-Ziel.
LÖSCHEN Sie /odstargets/http/ {name}	Löschen Sie ein HTTP Open Data Stream-Ziel.
GET /odstargets/http/ {name}	Rufen Sie ein bestimmtes HTTP Open Data Stream-Ziel ab.
GET /odstargets/kafka	Ruft alle Kafka Open Data Stream-Ziele ab.
BEITRAG /odstargets/kafka	Erstellen Sie ein neues Kafka Open Data Stream-Ziel.
LÖSCHE /odstargets/kafka/ {name}	Löschen Sie ein Kafka Open Data Stream-Ziel.
GET /odstargets/kafka/ {name}	Rufen Sie ein bestimmtes Kafka Open Data Stream-Ziel ab.
GET /odstargets/mongodb	Ruft alle MongoDB Open Data Stream-Ziele ab.
BEITRAG /odstargets/mongodb	Erstellen Sie ein neues MongoDB Open Data Stream-Ziel.
LÖSCHEN Sie /odstargets/mongodb/ {name}	Löschen Sie ein MongoDB Open Data Stream-Ziel.
GET /odstargets/mongodb/ {name}	Rufen Sie ein bestimmtes MongoDB Open Data Stream-Ziel ab.
GET /odstargets/raw	Ruft alle Raw Open Data Stream-Ziele ab.
BEITRAG /odstargets/raw	Erstellen Sie ein neues Raw Open Data Stream-Ziel.
LÖSCHE /odstargets/raw/ {name}	Löscht ein Raw Open Data Stream-Ziel.
GET /odstargets/raw/ {name}	Rufen Sie ein bestimmtes Raw Open Data Stream-Ziel ab.
GET /odstargets/syslog	Ruft alle Syslog Open Data Stream-Ziele ab.
POST /odstargets/syslog	Erstellen Sie ein neues Syslog Open Data Stream-Ziel.
LÖSCHEN Sie /odstargets/syslog/ {name}	Löschen Sie ein Syslog Open Data Stream-Ziel.

Betrieb	Beschreibung
GET /odstargets/syslog/ {name}	Rufen Sie ein bestimmtes Syslog Open Data Stream-Ziel ab.

Einzelheiten der Operation

`GET /odstargets`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{}
```

`GET /odstargets/http`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{}
```

`GET /odstargets/http/{name}`

Geben Sie die folgenden Parameter an.

`name`: ***Schnur***
Der Name des Ziels.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{}
```

`GET /odstargets/kafka`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "brokers": [],
    "compression": "string",
    "name": "string",
    "partition_strategy": "string",
    "protocol": "string",
    "skip_cert_verification": true,
    "tls_ca_certs": "string",
    "tls_client_cert": "string",
    "tls_client_key": "string"
}
```

`GET /odstargets/kafka/{name}`

Geben Sie die folgenden Parameter an.

`name`: ***Schnur***
Der Name des Ziels.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "brokers": [],
    "compression": "string",
```

```
    "name": "string",
    "partition_strategy": "string",
    "protocol": "string",
    "skip_cert_verification": true,
    "tls_ca_certs": "string",
    "tls_client_cert": "string",
    "tls_client_key": "string"
}
```

`GET /odstargets/mongodb`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{}
```

`GET /odstargets/mongodb/{name}`

Geben Sie die folgenden Parameter an.

`name`: ***Schnur***
Der Name des Ziels.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{}
```

`GET /odstargets/raw`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{}
```

`GET /odstargets/raw/{name}`

Geben Sie die folgenden Parameter an.

`name`: ***Schnur***
Der Name des Ziels.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{}
```

`GET /odstargets/syslog`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "batch_min_bytes": 0,
    "concurrent_connections": 0,
    "host": "string",
    "localtime": true,
    "name": "string",
    "port": 0,
    "protocol": "string",
    "skip_cert_verification": true,
    "tcp_length_prefix_framing": true,
    "tls_ca_certs": "string",
    "tls_client_cert": "string",
```

```
    "tls_client_key": "string"
}
```

`GET /odstargets/syslog/{name}`

Geben Sie die folgenden Parameter an.

`name:` ***Schnur***
Der Name des Ziels.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "batch_min_bytes": 0,
    "concurrent_connections": 0,
    "host": "string",
    "localtime": true,
    "name": "string",
    "port": 0,
    "protocol": "string",
    "skip_cert_verification": true,
    "tcp_length_prefix_framing": true,
    "tls_ca_certs": "string",
    "tls_client_cert": "string",
    "tls_client_key": "string"
}
```

`POST /odstargets/http`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

`name:` ***Schnur***
Der Name für das Ziel.

`host:` ***Schnur***
Der Hostname oder die IP-Adresse des Remote-HTTP-Servers.

`port:` ***Zahl***
Die TCP-Portnummer des HTTP-Servers.

`protocol:` ***Schnur***
Das Protokoll, über das Daten übertragen werden.

Die folgenden Werte sind gültig:

- `http`
- `https`

`skip_cert_verification:` ***Boolesch***
(Optional) Gibt an, ob die TLS-Zertifikatsüberprüfung für verschlüsselte Daten umgangen werden soll. Dieser Parameter ist nur gültig, wenn `protocol` auf `https` gesetzt ist.

`pipeline:` ***Boolesch***
Gibt an, ob mehrere gleichzeitige HTTP-Verbindungen aktiviert sind, wodurch die Durchsatzgeschwindigkeit verbessert werden kann.

`additional_header:` ***Schnur***
(Optional) Gibt einen zusätzlichen HTTP-Header an, der in jede Anfrage aufgenommen werden soll. Header müssen im folgenden Format angegeben werden: "<key>:<value>“. Zum Beispiel: „additional_header“: „Accept: text/html“.

`authentication:` ***Objekt***

Ein Objekt, das HTTP-Authentifizierungsdaten enthält.

`auth_type:` ***Schnur***

Die Art der HTTP-Authentifizierung.

Die folgenden Werte sind gültig:

- `none`
- `basic`
- `aws`
- `azure_storage`
- `azure_ad`
- `crowdstrike`

`username:` ***Schnur***

(Optional) Der Name des Benutzers. Diese Option ist erforderlich, wenn `auth_type` auf `basic` gesetzt ist oder wenn `auth_type` auf `azure_ad` und `grant_type` auf `resource_owner` gesetzt ist.

`password:` ***Schnur***

(Optional) Das Passwort des Benutzers. Diese Option ist erforderlich, wenn `auth_type` auf `basic` gesetzt ist oder wenn `auth_type` auf `azure_ad` und `grant_type` auf `resource_owner` gesetzt ist.

`access_key:` ***Schnur***

(Optional) Die Zugriffsschlüssel-ID. Diese Option ist für die AWS- und Azure Storage-Authentifizierung erforderlich.

`secret_key:` ***Schnur***

(Optional) Der geheime Zugriffsschlüssel. Diese Option ist für die AWS-Authentifizierung erforderlich.

`service:` ***Schnur***

(Optional) Der Servicecode des AWS-Service, z. B. „AmazonEC2". Diese Option ist für die AWS-Authentifizierung erforderlich.

`region:` ***Schnur***

(Optional) Der Name der AWS-Region, z. B. „us-west-1". Diese Option ist für die AWS-Authentifizierung erforderlich.

`grant_type:` ***Schnur***

(Optional) Der OAuth 2.0-Grant-Typ. Diese Option ist für die Microsoft Entra ID-Authentifizierung erforderlich.

Die folgenden Werte sind gültig:

- `client`
- `resource_owner`

`client_id:` ***Schnur***

(Optional) Die Client-ID. Diese Option ist für die Microsoft Entra ID- und Crowdstrike-Authentifizierung erforderlich.

`client_secret:` ***Schnur***

(Optional) Der geheime Client-Schlüssel. Diese Option ist für die Microsoft Entra ID- und Crowdstrike-Authentifizierung erforderlich.

`resource:` ***Schnur***

(Optional) Die Microsoft Entra ID-Ressourcen-URI. Diese Option ist für die Microsoft Entra ID-Authentifizierung erforderlich.

`token_endpoint:` ***Schnur***

(Optional) Der Endpunkt Microsoft Entra ID /token. Zum Beispiel: "https://login.microsoftonline.com/ <tenant_id>/oauth2/token“. Diese Option ist für die Microsoft Entra ID-Authentifizierung erforderlich.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "additional_header": "string",
    "authentication": {
        "auth_type": "string",
        "username": "string",
        "password": "string",
        "access_key": "string",
        "secret_key": "string",
        "service": "string",
        "region": "string",
        "grant_type": "string",
        "client_id": "string",
        "client_secret": "string",
        "resource": "string",
        "token_endpoint": "string"
    },
    "host": "string",
    "name": "string",
    "pipeline": true,
    "port": 0,
    "protocol": "string",
    "skip_cert_verification": true
}
```

`POST /odstargets/kafka`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

`name:` ***Schnur***

Der Name für das Ziel.

`brokers:` ***Reihe von Objekten***

Ein Array von einem oder mehreren Objekten, die Informationen über Kafka Brokers enthalten.

`host:` ***Schnur***

Der Hostname oder die IP-Adresse des Remote-Kafka-Brokers.

`port:` ***Zahl***

Die TCP-Portnummer des Kafka-Brokers.

`compression:` ***Schnur***

(Optional) Die Komprimierungsmethode, die auf übertragene Daten angewendet werden soll.

Die folgenden Werte sind gültig:

- `none`
- `gzip`
- `snappy`

`partition_strategy:` ***Schnur***

(Optional) Die Partitionierungsmethode, die auf übertragene Daten angewendet werden soll.

Die folgenden Werte sind gültig:

- `hash_key`
- `manual`
- `random`
- `round_robin`

`protocol:` ***Schnur***

Das Protokoll, über das Daten übertragen werden.

Die folgenden Werte sind gültig:

- `tcp`
- `tls`

`tls_client_cert:` ***Schnur***

(Optional) Das TLS-Client-Zertifikat, das während des TLS-Handshakes an den Kafka-Server gesendet wird. Geben Sie diese Option an, wenn die Client-Authentifizierung auf dem Kafka-Server aktiviert ist.

`tls_client_key:` ***Schnur***

(Optional) Der private Schlüssel des TLS-Client-Zertifikats, das durch den Parameter tls_client_cert angegeben wird. Geben Sie diese Option an, wenn die Client-Authentifizierung auf dem Kafka-Server aktiviert ist.

`skip_cert_verification:` ***Boolesch***

(Optional) Gibt an, ob die TLS-Zertifikatsüberprüfung für verschlüsselte Daten umgangen werden soll. Dieser Parameter ist nur gültig, wenn das Protokoll auf tls gesetzt ist.

`tls_ca_certs:` ***Schnur***

(Optional) Die vertrauenswürdigen Zertifikate, mit denen das Kafka-Serverzertifikat validiert werden soll, im PEM-Format. Geben Sie diese Option an, wenn Ihr Kafka-Serverzertifikat nicht von einer gültigen Zertifizierungsstelle (CA) signiert wurde. Wenn diese Option nicht angegeben ist, wird das Serverzertifikat mit der integrierten Liste gültiger CA-Zertifikate validiert. Diese Option ist nur gültig, wenn das Protokoll TLS ist.

`authentication:` ***Objekt***

(Optional) Ein Objekt, das Kafka-Authentifizierungsdaten enthält.

`auth_type:` ***Schnur***

Die Art der SASL-Authentifizierung.

Die folgenden Werte sind gültig:

- `scram`

`username:` ***Schnur***

Der Benutzername des SASL-Benutzers.

`password:` ***Schnur***

Das Passwort des SASL-Benutzers.

`algorithm:` ***Schnur***

Der Hashing-Algorithmus für die SASL-Authentifizierung.

Die folgenden Werte sind gültig:

- `sha256`
- `sha512`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "authentication": {
        "auth_type": "string",
        "username": "string",
        "password": "string",
```

```
        "algorithm": "string"
    },
    "brokers": {
        "host": "string",
        "port": 0
    },
    "compression": "string",
    "name": "string",
    "partition_strategy": "string",
    "protocol": "string",
    "skip_cert_verification": true,
    "tls_ca_certs": "string",
    "tls_client_cert": "string",
    "tls_client_key": "string"
}
```

`POST /odstargets/mongodb`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

`name:` ***Schnur***

Der Name für das Ziel.

`host:` ***Schnur***

Der Hostname oder die IP-Adresse des Remote-MongoDB-Servers.

`port:` ***Zahl***

Die TCP-Portnummer des MongoDB-Servers.

`encrypt:` ***Boolesch***

(Optional) Gibt an, ob Daten mit TLS verschlüsselt sind.

`skip_cert_verification:` ***Boolesch***

(Optional) Gibt an, ob die TLS-Zertifikatsüberprüfung für verschlüsselte Daten umgangen werden soll. Dieser Parameter ist nur gültig, wenn `encrypt` auf `true` gesetzt ist.

`authentication:` ***Reihe von Objekten***

(Optional) Ein Array von Objekten, die MongoDB-Authentifizierungsdaten enthalten.

`database:` ***Schnur***

Der Name der MongoDB-Datenbank.

`user:` ***Schnur***

Der Name des Benutzers, der berechtigt ist, die angegebene Datenbank zu ändern.

`password:` ***Schnur***

Das Passwort des Benutzers.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "authentication": {
        "database": "string",
        "user": "string",
        "password": "string"
    },
    "encrypt": true,
    "host": "string",
    "name": "string",
    "port": 0,
    "skip_cert_verification": true
```

```
}
```

`POST /odstargets/raw`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

`name:` ***Schnur***
Der Name für das Ziel.

`host:` ***Schnur***
Der Hostname oder die IP-Adresse des Remoteservers.

`port:` ***Zahl***
Die TCP- oder UDP-Portnummer des Remoteservers.

`protocol:` ***Schnur***
Das Protokoll, über das Daten übertragen werden.

Die folgenden Werte sind gültig:

- `tcp`
- `udp`

`compression:` ***Boolesch***
(Optional) Gibt an, ob die GZIP-Komprimierung auf übertragene Daten angewendet wird.

`gzip_threshold_bytes:` ***Zahl***
(Optional) Die Anzahl der Byte, die den Schwellenwert für die Erstellung einer neuen Nachricht angibt. Alle 30 Sekunden sendet der Sensor oder die Konsole Nachrichten, die die angegebene Größe überschreiten, um zu verhindern, dass Nachrichten zu groß werden. Diese Option ist nur gültig, wenn `compression` auf `true` gesetzt ist.

`gzip_threshold_seconds:` ***Zahl***
(Optional) Die Anzahl der Sekunden, die den Schwellenwert für das Erstellen einer neuen Nachricht angibt. Alle 30 Sekunden sendet der Sensor oder die Konsole Nachrichten, die länger als den angegebenen Zeitraum geschrieben wurden, um zu verhindern, dass Nachrichten zu groß werden. Diese Option ist nur gültig, wenn `compression` auf `true` gesetzt ist.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "compression": true,
    "gzip_threshold_bytes": 0,
    "gzip_threshold_seconds": 0,
    "host": "string",
    "name": "string",
    "port": 0,
    "protocol": "string"
}
```

`POST /odstargets/syslog`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

`name:` ***Schnur***
Der Name für das Ziel.

`host:` ***Schnur***

Der Hostname oder die IP-Adresse des Remote-Syslog-Servers.

`port:` ***Zahl***

Die TCP- oder UDP-Portnummer des Remote-Syslog-Servers.

`tcp_length_prefix_framing:` ***Boolesch***

(Optional) Gibt an, ob die Anzahl der Byte in einer Nachricht dem Anfang der Nachricht vorangestellt werden soll. Wenn dieser Parameter auf false gesetzt ist, wird das Ende jeder Nachricht durch einen abschließenden Zeilenumbruch begrenzt.

`batch_min_bytes:` ***Zahl***

(Optional) Die Mindestanzahl von Bytes, die gleichzeitig an den Syslog-Server gesendet werden müssen.

`concurrent_connections:` ***Zahl***

(Optional) Die Anzahl der gleichzeitigen Verbindungen, über die Nachrichten gesendet werden.

`localtime:` ***Boolesch***

(Optional) Gibt an, ob Zeitstempel auf die lokale Zeitzone des Sensor oder der Konsole verweisen. Wenn dieser Parameter auf false gesetzt ist, verweisen Zeitstempel auf GMT.

`protocol:` ***Schnur***

Das Protokoll, über das Daten übertragen werden.

Die folgenden Werte sind gültig:

- `tcp`
- `udp`
- `tls`

`tls_client_cert:` ***Schnur***

(Optional) Das TLS-Client-Zertifikat, das während des TLS-Handshakes an den Syslog-Server gesendet wird. Geben Sie diese Option an, wenn die Client-Authentifizierung auf dem Syslog-Server aktiviert ist.

`tls_client_key:` ***Schnur***

(Optional) Der private Schlüssel des TLS-Client-Zertifikats, das durch den Parameter tls_client_cert angegeben wird. Geben Sie diese Option an, wenn die Client-Authentifizierung auf dem Syslog-Server aktiviert ist.

`skip_cert_verification:` ***Boolesch***

(Optional) Gibt an, ob die TLS-Zertifikatsüberprüfung für verschlüsselte Daten umgangen werden soll. Dieser Parameter ist nur gültig, wenn das Protokoll auf tls gesetzt ist.

`tls_ca_certs:` ***Schnur***

(Optional) Die vertrauenswürdigen Zertifikate, mit denen das Syslog-Serverzertifikat validiert werden soll, im PEM-Format. Geben Sie diese Option an, wenn Ihr Syslog-Serverzertifikat nicht von einer gültigen Zertifizierungsstelle (CA) signiert wurde. Wenn diese Option nicht angegeben ist, wird das Serverzertifikat mit der integrierten Liste gültiger CA-Zertifikate validiert. Diese Option ist nur gültig, wenn das Protokoll TLS ist und skip_cert_verification falsch ist.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "batch_min_bytes": 0,
    "concurrent_connections": 0,
    "host": "string",
    "localtime": true,
    "name": "string",
    "port": 0,
```

```
    "protocol": "string",
    "skip_cert_verification": true,
    "tcp_length_prefix_framing": true,
    "tls_ca_certs": "string",
    "tls_client_cert": "string",
    "tls_client_key": "string"
}
```

`DELETE /odstargets/http/{name}`

Geben Sie die folgenden Parameter an.

`name` : ***Schnur***
Der Name des Ziels.

`DELETE /odstargets/kafka/{name}`

Geben Sie die folgenden Parameter an.

`name` : ***Schnur***
Der Name des Ziels.

`DELETE /odstargets/mongodb/{name}`

Geben Sie die folgenden Parameter an.

`name` : ***Schnur***
Der Name des Ziels.

`DELETE /odstargets/raw/{name}`

Geben Sie die folgenden Parameter an.

`name` : ***Schnur***
Der Name des Ziels.

`DELETE /odstargets/syslog/{name}`

Geben Sie die folgenden Parameter an.

`name` : ***Schnur***
Der Name des Ziels.

Paketsuche

Sie können nach Paketen suchen und diese herunterladen, die auf dem ExtraHop-System gespeichert sind. Das heruntergeladene Pakete können dann mit einem Drittanbieter-Tool wie Wireshark analysiert werden.

Weitere Informationen zu Paketen finden Sie unter Pakete.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /Pakete/Suche	Suchen Sie nach Paketen, indem Sie Parameter in einer URL angeben.

Bedienung	Beschreibung
POST /Pakete/Suche	Suchen Sie nach Paketen, indem Sie Parameter in einer JSON-Zeichenfolge angeben.

Einzelheiten der Operation

```
GET /packets/search
```

Geben Sie die folgenden Parameter an.

`output:` ***Schnur***

(Optional) Das Ausgabeformat. * `pcap` - Eine PCAP-Datei, die Pakete enthält. * `keylog_txt` - Eine Keylog-Textdatei, die Geheimnisse für die Entschlüsselung enthält. * `pcapng` - Eine PCAPNG-Datei, die sowohl Pakete als auch Geheimnisse für die Entschlüsselung enthalten kann. * `zip` - Eine ZIP-Datei, die sowohl eine PCAP- als auch eine Keylog-Textdatei enthält. * `extract` - Eine ZIP-Datei, die Dateien enthält, die aus Paketen extrahiert wurden, die der Abfrage entsprachen. Diese Option ist nur gültig, wenn Sie vollen Zugriff auf das NDR-Modul haben.

Die folgenden Werte sind gültig:

- `pcap`
- `keylog_txt`
- `pcapng`
- `zip`
- `extract`

`include_secrets:` ***Boolesch***

(Optional) Gibt an, ob Geheimnisse in die PCAPNG-Datei aufgenommen werden sollen. Diese Option ist nur gültig, wenn `output` auf `pcapng` gesetzt ist.

`decrypt_files:` ***Boolesch***

(Optional) Gibt an, ob entpackte Dateien mit gespeicherten Geheimnissen entschlüsselt werden sollen. Diese Option ist nur gültig, wenn der `output`-Parameter `extract` ist.

`limit_bytes:` ***Schnur***

(Optional) Die ungefähre maximale Anzahl von Byte, die zurückgegeben werden sollen. Nachdem das ExtraHop-System Pakete gefunden hat, die der in den Suchkriterien angegebenen Größe entsprechen, stoppt das System die Suche nach weiteren Paketen. Da das System jedoch mehrere Pakete gleichzeitig analysiert, kann die Gesamtgröße der zurückgegebenen Pakete größer als die angegebene Größe sein. Die Standardeinheit ist Byte, aber Sie können auch andere Einheiten mit einem Einheitensuffix angeben. Der Standardwert ist „100 MB“. **Hinweis**: Wenn die Ausgabe „extract“ lautet, gibt es einen Maximalwert für dieses Feld. Das Standardmaximum ist „100 MB“, aber das Maximum kann in der laufenden Konfiguration geändert werden. Wenn die Ausgabe nicht „Extrahieren“ ist, gibt es keinen Maximalwert.

`limit_search_duration:` ***Schnur***

(Optional) Die ungefähre maximale Zeit für die Durchführung der Paketsuche. Nach Ablauf der angegebenen Zeit hört das ExtraHop-System auf, nach weiteren Paketen zu suchen. Das System verlängert jedoch die angegebene Zeit, um die Analyse von Paketen abzuschließen, die vor Ablauf der Zeit durchsucht wurden, und das System analysiert mehrere Pakete gleichzeitig. Daher kann die Suche länger als die angegebene Zeit dauern. Die Standardeinheit ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe. Der Standardwert ist „5m“. **Hinweis**: Wenn die Ausgabe „extract“ lautet, gibt es einen Maximalwert für dieses Feld. Das Standardmaximum ist „5 m“, aber das Maximum kann in der laufenden Konfiguration geändert werden. Wenn die Ausgabe nicht „Extrahieren“ ist, gibt es keinen Maximalwert.

`always_return_body:` ***Boolesch***

(Optional) Gibt das Verhalten an, wenn die Abfrage keinen Paketen entspricht oder wenn die Pakete, auf die die Abfrage zutrifft, keine Dateien enthalten. Wenn der Wert wahr ist, gibt das System eine leere Datei und einen 200-Statuscode zurück. Wenn der Wert falsch ist, gibt das System einen 204-Statuscode zurück, aber keine Datei.

`from:` ***Schnur***

Der Anfangszeitstempel des Zeitbereichs, den die Suche umfassen wird, ausgedrückt in Millisekunden seit der Epoche. Ein negativer Wert gibt an, dass die Suche mit Paketen beginnt, die zu einem Zeitpunkt in der Vergangenheit erfasst wurden. Geben Sie beispielsweise -10m an, um die Suche mit Paketen zu beginnen, die 10 Minuten vor dem Zeitpunkt der Anfrage erfasst wurden. Negative Werte können mit einer anderen Zeiteinheit als Millisekunden angegeben werden, z. B. Sekunden oder Stunden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`until:` ***Schnur***

(Optional) Der Endzeitstempel des Zeitbereichs, den die Suche einschließen wird, ausgedrückt in Millisekunden seit der Epoche. Ein Wert 0 gibt an, dass die Suche mit Paketen endet, die zum Zeitpunkt der Suche erfasst wurden. Ein negativer Wert gibt an, dass die Suche mit Paketen endet, die zu einem Zeitpunkt in der Vergangenheit erfasst wurden. Geben Sie beispielsweise -5m an, um die Suche mit Paketen zu beenden, die 5 Minuten vor dem Zeitpunkt der Anfrage erfasst wurden. Negative Werte können mit einer anderen Zeiteinheit als Millisekunden angegeben werden, z. B. Sekunden oder Stunden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`bpf:` ***Schnur***

(Optional) Die Berkeley Paket Filter (BPF) -Syntax für die Paketsuche. Weitere Informationen zur BPF-Syntax finden Sie in der REST-API-Leitfaden.

`ip1:` ***Schnur***

(Optional) Gibt Pakete zurück, die an die angegebene IP-Adresse gesendet oder von dieser empfangen wurden.

`port1:` ***Schnur***

(Optional) Gibt Pakete zurück, die vom angegebenen Port gesendet oder empfangen wurden.

`ip2:` ***Schnur***

(Optional) Gibt Pakete zurück, die an die angegebene IP-Adresse gesendet oder von dieser empfangen wurden.

`port2:` ***Schnur***

(Optional) Gibt Pakete zurück, die vom angegebenen Port gesendet oder empfangen wurden.

`POST /packets/search`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Parameter der Paketsuche.

`output:` ***Schnur***

(Optional) Das Ausgabeformat.

Die folgenden Werte sind gültig:

- `pcap`
- `keylog_txt`
- `pcapng`
- `zip`
- `extract`

`include_secrets:` ***Boolesch***

(Optional) Ob TLS-Geheimnisse zusammen mit Paketdaten in .pcapng-Dateien aufgenommen werden sollen oder nicht. Nur gültig, wenn „output“ „pcapng“ ist.

`decrypt_files:` ***Boolesch***

(Optional) Gibt an, ob entpackte Dateien mit gespeicherten Geheimnissen entschlüsselt werden sollen. Diese Option ist nur gültig, wenn der `output`-Parameter `extract` ist.

`limit_bytes:` ***Schnur***

(Optional) Die ungefähre maximale Anzahl von Byte, die zurückgegeben werden sollen. Nachdem das ExtraHop-System Pakete gefunden hat, die der in den Suchkriterien angegebenen Größe entsprechen, stoppt das System die Suche nach weiteren Paketen. Da das System jedoch mehrere Pakete gleichzeitig analysiert, kann die Gesamtgröße der zurückgegebenen Pakete größer als die angegebene Größe sein. Die Standardeinheit ist Byte, aber Sie können auch andere Einheiten mit einem Einheitensuffix angeben. Der Standardwert ist „100 MB“. **Hinweis**: Wenn die Ausgabe „extract“ lautet, gibt es einen Maximalwert für dieses Feld. Das Standardmaximum ist „100 MB“, aber das Maximum kann in der laufenden Konfiguration geändert werden. Wenn die Ausgabe nicht „Extrahieren“ ist, gibt es keinen Maximalwert.

`limit_search_duration:` ***Schnur***

(Optional) Die ungefähre maximale Zeit für die Durchführung der Paketsuche. Nach Ablauf der angegebenen Zeit hört das ExtraHop-System auf, nach weiteren Paketen zu suchen. Das System verlängert jedoch die angegebene Zeit, um die Analyse von Paketen abzuschließen, die vor Ablauf der Zeit durchsucht wurden, und das System analysiert mehrere Pakete gleichzeitig. Daher kann die Suche länger als die angegebene Zeit dauern. Die Standardeinheit ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe. Der Standardwert ist „5m“. **Hinweis**: Wenn die Ausgabe „extract“ lautet, gibt es einen Maximalwert für dieses Feld. Das Standardmaximum ist „5 m“, aber das Maximum kann in der laufenden Konfiguration geändert werden. Wenn die Ausgabe nicht „Extrahieren“ ist, gibt es keinen Maximalwert.

`always_return_body:` ***Boolesch***

(Optional) Gibt das Verhalten an, wenn die Abfrage keinen Paketen entspricht oder wenn die Pakete, auf die die Abfrage zutrifft, keine Dateien enthalten. Wenn der Wert wahr ist, gibt das System eine leere Datei und einen 200-Statuscode zurück. Wenn der Wert falsch ist, gibt das System einen 204-Statuscode zurück, aber keine Datei.

`from:` ***Schnur***

Der Anfangszeitstempel des Zeitbereichs, den die Suche umfassen wird, ausgedrückt in Millisekunden seit der Epoche. Ein negativer Wert gibt an, dass die Suche mit Paketen beginnt, die zu einem Zeitpunkt in der Vergangenheit erfasst wurden. Geben Sie beispielsweise -10m an, um die Suche mit Paketen zu beginnen, die 10 Minuten vor dem Zeitpunkt der Anfrage erfasst wurden. Negative Werte können mit einer anderen Zeiteinheit als Millisekunden angegeben werden, z. B. Sekunden oder Stunden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`until:` ***Schnur***

(Optional) Der Endzeitstempel des Zeitbereichs, den die Suche einschließen wird, ausgedrückt in Millisekunden seit der Epoche. Ein Wert 0 gibt an, dass die Suche mit Paketen endet, die zum Zeitpunkt der Suche erfasst wurden. Ein negativer Wert gibt an, dass die Suche mit Paketen endet, die zu einem Zeitpunkt in der Vergangenheit erfasst wurden. Geben Sie beispielsweise -5m an, um die Suche mit Paketen zu beenden, die 5 Minuten vor dem Zeitpunkt der Anfrage erfasst wurden. Negative Werte können mit einer anderen Zeiteinheit als Millisekunden angegeben werden, z. B. Sekunden oder Stunden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`bpf:` ***Schnur***

(Optional) Die Berkeley Paket Filter (BPF) -Syntax für die Paketsuche. Weitere Hinweise zur BPF-Syntax finden Sie unter Filtert Pakete mit der Berkeley-Paketfilter-Syntax.

`ip1:` ***Schnur***

(Optional) Gibt Pakete zurück, die an die angegebene IP-Adresse gesendet oder von dieser empfangen wurden.

`port1:` ***Schnur***

(Optional) Gibt Pakete zurück, die vom angegebenen Port gesendet oder empfangen wurden.

`ip2:` ***Schnur***

(Optional) Gibt Pakete zurück, die an die angegebene IP-Adresse gesendet oder von dieser empfangen wurden.

`port2:` ***Schnur***

(Optional) Gibt Pakete zurück, die vom angegebenen Port gesendet oder empfangen wurden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "always_return_body": true,
    "bpf": "string",
    "decrypt_files": true,
    "from": "string",
    "include_secrets": true,
    "ip1": "string",
    "ip2": "string",
    "limit_bytes": "string",
    "limit_search_duration": "string",
    "output": "string",
    "port1": "string",
    "port2": "string",
    "until": "string"
}
```

Pakete mit der Berkeley-Paketfilter-Syntax filtern

Suchen Sie nach Paketen mit der Berkeley Packet Filter (BPF) -Syntax allein oder in Kombination mit den integrierten Filtern.

Berkeley-Paketfilter sind eine einfache Schnittstelle zu Datenverbindungsebenen und ein leistungsstarkes Tool für die Analyse der Erkennung von Eindringlingen. Die BPF-Syntax ermöglicht es Benutzern, Filter zu schreiben, die schnell nach bestimmten Paketen suchen, um die wichtigsten Informationen zu sehen.

Das ExtraHop-System erstellt einen synthetischen Paket-Header aus den Paketindexdaten und führt dann die BPF-Syntaxabfragen für den Paket-Header aus, um sicherzustellen, dass Abfragen viel schneller sind als das Scannen der gesamten Paketnutzlast. Beachten Sie, dass ExtraHop nur eine Teilmenge der BPF-Syntax unterstützt. siehe Unterstützte BPF-Syntax.

Die BPF-Syntax besteht aus einem oder mehreren Primitiven, denen ein oder mehrere Qualifikatoren vorangestellt sind. Primitive bestehen normalerweise aus einer ID (Name oder Nummer), der ein oder mehrere Qualifikatoren vorangestellt sind. Es gibt drei verschiedene Arten von Qualifikationsspielen:

Art

Qualifikatoren, die angeben, auf welchen Typ sich der ID-Name oder die ID-Nummer bezieht. Zum Beispiel `host`, `net`, `port`, und `portrange`. Wenn es kein Qualifikationsmerkmal gibt, `host` wird angenommen.

dir

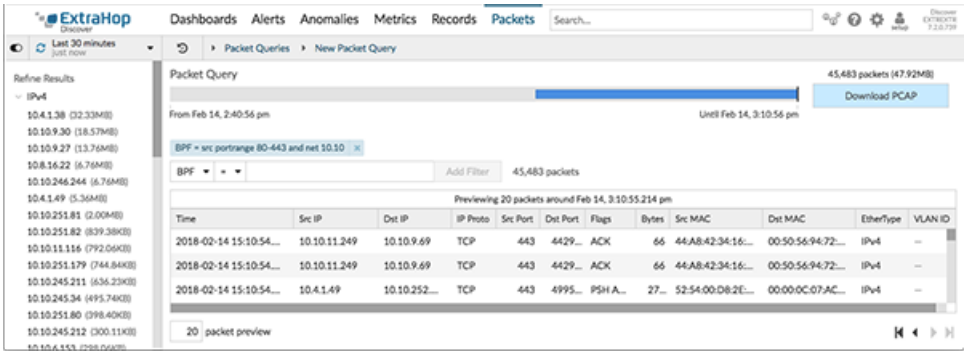
Qualifier, die eine bestimmte Übertragungsrichtung zu und/oder von einer ID angeben. Mögliche Richtungen sind `src`, `dst`, `src and dst`, und `src or dst`. Zum Beispiel `dst net 128.3`.

Proto

Qualifikatoren, die die Übereinstimmung auf das jeweilige Protokoll beschränken. Mögliche Protokolle sind `ether`, `ip`, `ip6`, `tcp`, und `udp`.

Fügen Sie einen Filter mit BPF-Syntax hinzu

1. Loggen Sie sich in das ExtraHop-System ein über `https://<extrahop-hostname-or-IP-address>`.
2. Klicken Sie im oberen Menü auf **Pakete**.
3. Wählen Sie im Bereich Dreifeld-Filter **BPF**, und geben Sie dann Ihre Filtersyntax ein. Geben Sie beispielsweise `src portrange 80-443 and net 10.10`.
4. klicken **PCAP herunterladen** um die PCAP mit Ihren gefilterten Ergebnissen zu speichern.



Unterstützte BPF-Syntax

Das ExtraHop-System unterstützt die folgende Teilmenge der BPF-Syntax zum Filtern von Paketen.

Hinweis

- ExtraHop unterstützt nur numerische IP-Adresssuchen. Hostnamen sind nicht erlaubt.
- Indizierung in Header, `[...]`, wird nur unterstützt für `tcpflags` und `ip_offset`. Zum Beispiel `tcp[tcpflags] & (tcp-syn|tcp-fin) != 0`
- ExtraHop unterstützt sowohl numerische als auch hexadezimale Werte für VLAN-ID-, EtherType- und IP-Protokollfelder. Stellen Sie Hexadezimalwerten 0x voran, z. B. 0x11.

Primitiv	Beispiele	Beschreibung
`[src|dst] host <host ip>`	`host 203.0.113.50` `dst host 198.51.100.200`	Entspricht einem Host als IP-Quelle, Ziel oder einer der beiden. Diese Host-Ausdrücke können in Verbindung mit anderen Protokollen wie ip, arp, rarp oder ip6 angegeben werden.
`ether [src|dst] host <MAC>`	`ether host 00:00:5E:00:53:00` `ether dst host 00:00:5E:00:53:00`	Entspricht einem Host als Ethernet-Quelle, Ziel oder einer der beiden.
`vlan <ID>`	`vlan 100`	Entspricht einem VLAN. Gültige ID-Nummern sind `0-4095`. Die VLAN-Prioritätsbits sind Null. Wenn das ursprüngliche Paket mehr als ein VLAN-Tag hatte, hat das synthetische Paket, mit dem der BPF übereinstimmt, nur das innerste VLAN-Tag.

Primitiv	Beispiele	Beschreibung

`[src|dst] portrange <p1>-<p2>`
oder
`[tcp|udp] [src|dst] portrange <p1>-<p2>`
	`src portrange 80-88`
	`tcp dst portrange 1501-1549`
		Ordnet Pakete zu oder von einem Port im angegebenen Bereich zu. Protokolle können auf einen Portbereich angewendet werden, um bestimmte Pakete innerhalb des Bereichs zu filtern.

`[ip|ip6][src|dst] proto <protocol>`	`proto 1`
	`src 10.4.9.40 and proto ICMP`
	`ip6 and src fe80::aebc:32ff:fe84:70b7 and proto 47`
	`ip and src 10.4.9.40 and proto 0x0006`
		Entspricht anderen IPv4- oder IPv6-Protokollen als TCP und UDP. Das Protokoll kann eine Zahl oder ein Name sein.

`[ip|ip6][tcp|udp] [src|dst] port <port>`	`udp and src port 2005` `ip6 and tcp and src port 80`	Entspricht IPv4- oder IPv6-Paketen an einem bestimmten Port.

`[src|dst] net <network>`	`dst net 192.168.1.0`
	`src net 10`
	`net 192.168.1.0/24`
		Ordnet Pakete zu oder von einer Quelle oder einem Ziel oder beidem zu, die sich in einem Netzwerk befinden. Eine IPv4-Netzwerknummer kann als einer der folgenden Werte angegeben werden:
		• Gepunktetes Viereck (x.x.x.x)
		• Dreifach gepunktet (x.x.x)
		• Gepunktetes Paar (x.x)
		• Einzelne Zahl (x)

`[ip|ip6] tcp tcpflags & (tcp-[ack|fin|syn|rst|push|urg|)`	`tcp[tcpflags] & (tcp-ack) !=0`
	`tcp[13] & 16 !=0`
	`ip6 and (ip6[40+13] & (tcp-syn) != 0)`
		Entspricht allen Paketen mit dem angegebenen TCP-Flag

Fragmentierte IPv4-Pakete (ip_offset! = 0)	`ip[6:2] & 0x3fff != 0x0000`	Stimmt mit allen Paketen mit Fragmenten überein.

Paarung

Mit dieser Ressource können Sie ein Token generieren, das für die Verbindung mit einem erforderlich ist Sensor zu einem Konsole.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
POST /pairing/token	Generieren Sie ein Token, das für die Verbindung mit dem erforderlich ist Sensor zu einem Konsole.

Einzelheiten der Operation

`POST /pairing/token`

Für diesen Vorgang gibt es keine Parameter.

Protokoll aufzeichnen

Aufzeichnungen sind strukturierte Fluss- und Transaktionsinformationen über Ereignisse in Ihrem Netzwerk.

Nachdem Sie das ExtraHop-System mit einem Datensatz Store verbunden haben, können Sie Datensatzinformationen generieren und an den Recordstore senden, und Sie können Datensätze abfragen, um Informationen über jedes Objekt in Ihrem Netzwerk abzurufen. Weitere Informationen finden Sie unter Abfragen von Datensätzen über die REST-API.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /records/cursor/ {cursor}	Veraltet. Ersetzt durch `POST /records/cursor`.
POST /Datensätze/Cursor	Ruft Datensätze ab einem bestimmten Cursor ab. Dieser Vorgang wird nur unterstützt, wenn Datensätze in einem ExtraHop-Recordstore (wie dem EXA 5300) oder auf CrowdStrike LogScale gespeichert sind.
POST /Datensätze/Suche	Führen Sie eine Datensatzprotokollabfrage durch.

Einzelheiten der Operation

`POST /records/search`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Datensatzprotokollabfrage.

`from:` ***Zahl***

Der Anfangszeitstempel des Zeitbereichs, den die Abfrage durchsucht, ausgedrückt in Millisekunden seit der Epoche. Ein negativer Wert gibt an, dass die Suche mit Datensätzen beginnt, die zu einem Zeitpunkt in der Vergangenheit erstellt wurden. Geben Sie beispielsweise -600000ms an, um die Suche mit Datensätzen zu beginnen, die 10 Minuten vor dem Zeitpunkt der Anfrage erstellt wurden. Negative Werte können mit einer anderen Zeiteinheit als Millisekunden angegeben werden, z. B. Sekunden oder Stunden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`until:` ***Zahl***

Der Endzeitstempel des Zeitbereichs, den die Abfrage durchsucht, ausgedrückt in Millisekunden seit der Epoche. Ein Wert 0 gibt an, dass die Suche mit Datensätzen endet, die zum Zeitpunkt der Anfrage erstellt wurden. Ein negativer Wert gibt an, dass die Suche mit Datensätzen endet, die zu einem Zeitpunkt in der Vergangenheit erstellt wurden. Geben

Sie beispielsweise -300000ms an, um die Suche mit Datensätzen zu beenden, die 5 Minuten vor dem Zeitpunkt der Anfrage erstellt wurden. Negative Werte können mit einer anderen Zeiteinheit als Millisekunden angegeben werden, z. B. Sekunden oder Stunden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe.

`types:` ***Reihe von Zeichenketten***

(Optional) Ein Array mit einem oder mehreren Datensatzformaten. Die Abfrage gibt nur Datensätze zurück, die den angegebenen Formaten entsprechen. Wenn kein Wert angegeben ist, gibt die Abfrage Datensätze eines beliebigen Typs zurück. Gültige Werte für dieses Feld werden im Feld Datensatztyp auf der Seite Datensatzformate angezeigt. Zum Beispiel: „~cifs“.

`limit:` ***Zahl***

Die maximale Anzahl von Datensätzen, die von der Abfrage zurückgegeben wurden. Der Höchstwert darf 10000 nicht überschreiten. Der Standardwert ist 100.

`offset:` ***Zahl***

Die Anzahl der Datensätze, die in den Abfrageergebnissen übersprungen werden sollen. Die Abfrage gibt Datensätze zurück, die mit dem Offsetwert beginnen. Dieser Parameter wird häufig mit den Grenzwert- und Sortierparametern kombiniert. Der Standardwert ist 0. Für ExtraHop-Recordstores ist der Höchstwert 10.000; Informationen zum Abrufen von Datensätzen, die nach den ersten 10.000 zurückgegeben wurden, finden Sie unter POST / records/cursor/. Für Recordstores von Drittanbietern gibt es keinen Maximalwert.

`sort:` ***Reihe von Objekten***

Die Liste von einem oder mehreren Sortierobjekten, die Sortierprioritäten angeben. Die zurückgegebenen Datensätze werden in der Reihenfolge sortiert, in der die Objekte aufgelistet sind. Die Parameter sind im Abschnitt sort_item unten definiert. Wenn keine sort_item-Werte angegeben werden, werden die Datensätze in absteigender Reihenfolge nach Zeitstempel sortiert.

`field:` ***Zeichenfolge***

Der Feldname, nach dem Datensätze zurückgegeben wurden, wird sortiert.

`direction:` ***Zeichenfolge***

Die Reihenfolge, in der die zurückgegebenen Datensätze sortiert werden. Die Standardreihenfolge ist absteigend. Nachdem alle anderen Sortierkriterien angewendet wurden oder wenn keine Sortierkriterien angegeben wurden, ist die Standardreihenfolge nach Zeitstempel absteigend.

Die folgenden Werte sind gültig:

- `asc`
- `desc`

`filter:` ***Objekt***

Das Objekt, das die Parameter enthält, die die Filterkriterien angeben. Die Parameter werden im Filterabschnitt unten definiert. Wenn keine Filterwerte angegeben werden, gibt die Abfrage alle Datensätze zurück, die dem Zeitbereich und allen angegebenen Datensatzformaten entsprechen.

`field:` ***Zeichenfolge***

Der Name des Feldes in dem Datensatz, der gefiltert werden soll. Die Abfrage vergleicht den Inhalt des Feldparameters mit dem Wert des Operandenparameters. Wenn der angegebene Feldname „.any“ ist, wird die Vereinigung aller Feldwerte durchsucht. Wenn der angegebene Feldname „.ipaddr“ oder „.port“ lautet, werden die Client-, Server-, Sender- und Empfängerrollen in die Suche einbezogen. Feldnamen befinden sich in Datensatzformaten, die im ExtraHop-System eingesehen werden können.

`operator:` ***Zeichenfolge***

Die Vergleichsmethode, die angewendet wird, wenn der Operandenwert mit dem Feldinhalt verglichen wird. Alle Filterobjekte benötigen einen Operator.

Die folgenden Werte sind gültig:

- `>`
- `<`
- `<=`
- `>=`
- `=`
- `!=`
- `startswith`
- `~`
- `!~`
- `and`
- `or`
- `not`
- `exists`
- `not_exists`
- `in`
- `not_in`

`operand:` ***Zeichenfolge oder Zahl oder Objekt***

Der Wert, den die Abfrage abzugleichen versucht. Die Abfrage vergleicht den Wert des Operanden mit dem Inhalt des Feldparameters und wendet die durch den Operatorparameter angegebene Vergleichsmethode an. Sie können den Operanden-Datentyp explizit angeben, wie in der REST-API-Leitfaden.

`rules:` ***Reihe von Objekten***

Die Liste von einem oder mehreren Filterobjekten innerhalb eines einzelnen Filterobjekts. Filterobjekte können rekursiv eingebettet werden. Für diesen Parameter sind nur die Operatoren „und“, „oder“ oder „nicht“ zulässig.

`context_ttl:` ***Zahl***

Die Zeitspanne, in der der Suchkontext aktiv bleibt. Der angegebene Wert wird als eine Dauer in der Zukunft interpretiert. Die Standardeinheit ist Millisekunden, aber andere Einheiten können mit einem Einheitensuffix angegeben werden. Sehen Sie die REST-API-Leitfaden für unterstützte Zeiteinheiten und Suffixe. Wenn ein Wert ungleich Null angegeben wird, enthält die Antwort eine Cursor-ID, die von POST /records/cursor/ akzeptiert wird. Dieser Parameter wird für Recordstores von Drittanbietern nicht unterstützt.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "context_ttl": 0,
    "filter": {
        "field": "string",
        "operator": "string",
        "operand": "string",
        "rules": []
    },
    "from": 0,
    "limit": 0,
    "offset": 0,
    "sort": {
        "field": "string",
        "direction": "string"
```

```
    },
    "types": [],
    "until": 0
}
```

`POST /records/cursor`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Cursor-ID, die die nächste Seite mit Ergebnissen in der Abfrage angibt.

`cursor:` ***Zeichenfolge***

Der eindeutige Bezeichner des Cursors, der die nächste Seite mit Ergebnissen in der Abfrage angibt.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "cursor": "string"
}
```

`context_ttl:` ***Zahl***

(Optional) Die Zeitspanne, in der der Suchkontext aktiv bleibt, ausgedrückt in Millisekunden.

`GET /records/cursor/{cursor}`

Geben Sie die folgenden Parameter an.

`cursor:` ***Zeichenfolge***

Die Cursor-ID.

`context_ttl:` ***Zahl***

(Optional) Die Zeitspanne, in der der Suchkontext aktiv bleibt, ausgedrückt in Millisekunden.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "cursor": "string",
    "from": 0,
    "records": {},
    "total": 0,
    "until": 0,
    "warnings": {}
}
```

Operandenwerte in Datensatzabfragen

Die `operand` Feld in der `POST /records/search` Methode gibt den Wert an, den eine Datensatzabfrage zu finden versucht. Sie können entweder nur den Wert oder sowohl den Datentyp als auch den Wert angeben. Wenn Sie nur den Wert angeben, bezieht sich die Abfrage auf das Datensatzformat, das mit dem verknüpft ist `field` Parameter zur Bestimmung des Datentyps des Werts.

Wenn Sie beispielsweise nach einer IP-Adresse suchen möchten, können Sie einen IP-Adressdatentyp angeben und dann die tatsächliche Adresse als Wert angeben.

Das folgende Beispiel spezifiziert explizit den Datentyp und den Wert des Operanden:

```
{
    "from": -1000,
    "filter": {
```

```
        "field" : "senderAddr",
        "operator": "=",
        "operand" : { "type" : "ipaddr4", "value": "1.2.3.4" }
    }
}
```

Das folgende Beispiel spezifiziert nur den Wert des Operanden:

```
{
    "from": -1000,
    "filter": {
        "field" : "senderAddr",
        "operator": "=",
        "operand" : "1.2.3.4"
    }
}
```

Sie können die folgenden Datentypen explizit angeben in der `operand` Feld:

- Anwendung
- boolesch
- Gerät

  Hinweis: Sie müssen die Discovery-ID des Gerät im Wertfeld angeben. Sie finden die Discovery-ID eines Gerät über `POST /devices/search` Betrieb.
- Gerätefilter
- Gerätegruppe
- Flow-Schnittstelle
- Flow-Netzwerk
- iPad dr4
- iPad dr6
- Nummer
- Netzwerk_Lokalität
- Objekt
- Schnur

Die `operand` Feld unterstützt die CIDR-Notation beim Filtern nach IP-Adressen; das `operator` Feld muss auf „=“ oder „! =“.

Sie können mehrere Filter angeben, indem Sie den `rules` Option, wie im folgenden Beispiel gezeigt:

```
{
  "filter": {
    "operator": "and",
    "rules": [
      {
        "field": "method",
        "operand": "SMB2_READ",
        "operator": "="
      },
      {
        "field": "reqL2Bytes",
        "operand": "100",
        "operator": ">"
      }
    ]
  },
  "types": [
    "~cifs"
  ],
```

```
    "from": "-30m"
}
```

Datensätze mit einem Gerätegruppenfilter abfragen

Um Datensätze in der REST-API nach Gerätegruppe zu filtern, müssen Sie eine senden `POST` Anfrage an den `/records/search` Endpunkt mit einem Datensatzabfragefilter, der die folgenden Kriterien erfüllt:

- Die `field` muss Geräte angeben, wie `client`, `server`, `sender`, oder `receiver`.
- Die `operator` muss entweder sein `in` oder `not_in`.
- Die `operand type` muss sein `device_group`.
- Die `operand value` muss eine Zeichenkettendarstellung der numerischen Gerätegruppen-ID sein. Sie können Gerätegruppen-IDs abrufen, indem Sie den Vorgang GET /devicegroup ausführen und den Inhalt des `id` Feld in der Antwort.

Die folgende Abfrage sucht beispielsweise nach Datensätzen, in denen das Client-Gerät Mitglied einer Gerätegruppe mit der ID 200 war:

```
{
    "from": "-30m",
    "filter": {
        "field": "client",
        "operator": "in",
        "operand": {
            "type": "device_group",
            "value": "200"
        }
    }
}
```

Sie können Datensätze auch nach Gerätegruppenkriterien filtern, ohne eine Gerätegruppe zu erstellen, indem Sie den Operandentyp angeben als `device_filter`. Mit der folgenden Abfrage wird beispielsweise nach Datensätzen gesucht, in denen auf dem Client-Gerät Windows 10 ausgeführt wird:

```
{
    "from": "-30m",
    "filter": {
        "field": "client",
        "operator": "in",
        "operand": {
            "type": "device_filter",
            "value": {
                "field": "software",
                "operand": "windows_10",
                "operator": "="
            }
        }
    }
}
```

Hinweis: Operandenwerte mit Typ `device_filter` für die Datensatzsuche sind genauso formatiert wie Gerätesuchfilter. Weitere Informationen finden Sie unter Operandenwerte für Gerätegruppen.

Datensätze mit einem Netzwerk-Lokalitätsfilter abfragen

Um Datensätze in der REST-API nach Gerätegruppe zu filtern, müssen Sie eine POST-Anfrage an die `/records/search` Endpunkt mit einem Datensatzabfragefilter, der die folgenden Kriterien erfüllt:

- Das Feld muss ein Datensatzfeld sein, das eine IP-Adresse angibt, z. B. `clientAddr`, `serverAddr`, `senderAddr`, oder `receiverAddr`.

- Der Betreiber muss entweder `in` oder `not_in`.
- Der Operandentyp muss `network_locality`.
- Der Operandenwert muss eine Zeichenkettendarstellung einer numerischen Netzwerk-Lokalitäts-ID sein. Sie können Lokalitäts-IDs mit dem `GET /networklocalities` Betrieb.

Die folgende Abfrage sucht beispielsweise nach Datensätzen, bei denen sich das Client-Gerät in einer Netzwerklokalität mit der ID von befindet `123`:

```
{
    "from": "-30m",
    "filter": {
        "field": "clientAddr",
        "operand": {
            "type": "network_locality",
            "value": "123"
        },
        "operator": "in"
    }
}
```

Unterstützte Zeiteinheiten

Für die meisten Parameter ist die Standardeinheit für die Zeitmessung Millisekunden. Die folgenden Parameter geben jedoch alternative Zeiteinheiten wie Minuten und Stunden zurück oder akzeptieren diese:

- Gerät
  - aktive_von
  - aktiv_bis
- Gerätegruppe
  - aktive_von
  - aktiv_bis
- Metriken
  - von
  - bis
- Protokoll aufzeichnen
  - von
  - bis
  - kontext_ttl

Die folgende Tabelle zeigt die unterstützten Zeiteinheiten:

Zeiteinheit	Einheitensuffix
Jahr	`y`
Monat	`M`
Woche	`w`
Tag	`d`
Stunde	`h`
Minute	`m`
Zweiter	`s`
Millisekunde	`ms`

Um für einen Parameter eine andere Zeiteinheit als Millisekunden anzugeben, hängen Sie das Einheitensuffix an den Wert an. Um beispielsweise Geräte anzufordern, die in den letzten 30 Minuten aktiv waren, geben Sie den folgenden Parameterwert an:

```
GET /api/v1/devices?active_from=-30m
```

Das folgende Beispiel spezifiziert eine Suche nach HTTP Datensätze, die vor 1 bis 2 Stunden erstellt wurden:

```
{
    "from": "-2h",
    "until": "-1h",
    "types": ["~http"]
}
```

Bericht

Ein Bericht ist eine PDF-Datei mit einem Dashboard, das Sie für die E-Mail-Zustellung an einen oder mehrere Empfänger planen können. Sie können angeben, wie oft die Berichts-E-Mail zugestellt wird und in welchem Zeitintervall die in der PDF-Datei enthaltenen Dashboard-Daten angezeigt werden.

Wichtig: Sie können nur Berichte von einer ECA-VM aus planen.

Hier sind einige wichtige Überlegungen zu Dashboard-Berichten:

- Sie können nur einen Bericht für Dashboards erstellen, die Ihnen gehören oder die mit Ihnen geteilt wurden. Ihre Fähigkeit, einen Bericht zu erstellen, hängt von Ihren Benutzerrechten ab. Wenden Sie sich an Ihren ExtraHop-Administrator, um Hilfe zu erhalten.
- Jeder Bericht kann nur mit einem Dashboard verknüpft werden.
- Wenn Sie einen Bericht für ein Dashboard erstellt haben, das später gelöscht wurde oder auf das Sie nicht mehr zugreifen konnten, wird die geplante E-Mail weiterhin an die Empfänger gesendet. Die E-Mail wird die PDF-Datei jedoch nicht enthalten und stattdessen die Empfänger darüber informieren, dass das Dashboard für den Berichtsbesitzer nicht verfügbar ist.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /reports	Rufen Sie alle Berichte ab.
POST/Berichte	Erstellen Sie einen Bericht.
/reports/ {id} LÖSCHEN	Löschen Sie einen bestimmten Bericht.
GET /reports/ {id}	Rufen Sie einen bestimmten Bericht ab.
PATCH /reports/ {id}	Aktualisieren Sie einen bestimmten Bericht.
GET /reports/ {id} /contents	Rufen Sie den Inhalt eines bestimmten Berichts ab.
PUT /reports/ {id} /contents	Ersetzt den Inhalt eines bestimmten Berichts.
GET /reports/ {id} /herunterladen	Rufen Sie das PDF eines Berichts ab.
POST /reports/ {id} /emailgroups	Ändern Sie die E-Mail-Gruppe, die einem bestimmten Dashboard-Bericht zugewiesen ist.
GET /reports/ {id} /emailgroups	Rufen Sie eine Liste von E-Mail-Gruppen ab, die einem bestimmten Dashboard-Bericht zugewiesen sind.

Betrieb	Beschreibung
LÖSCHEN Sie /reports/ {id} /emailgroups/ {group-id}	Entfernen Sie eine E-Mail-Gruppe aus einem bestimmten Dashboard-Bericht.
POST /reports/ {id} /emailgroups/ {group-id}	Fügen Sie einem bestimmten Dashboard-Bericht eine E-Mail-Gruppe hinzu.
POST /reports/ {id} /Warteschlange	Generieren und senden Sie sofort einen bestimmten Bericht.

Einzelheiten der Operation

`GET /reports`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "cc": [],
    "description": "string",
    "email_message": "string",
    "email_subject": "string",
    "enabled": true,
    "from": "string",
    "id": 0,
    "include_links": "string",
    "name": "string",
    "output": {},
    "owner": "string",
    "schedule": {},
    "until": "string"
}
```

`POST /reports`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Der Inhalt des Berichts.

`name:` ***Schnur***

Der Name des Berichts.

`description:` ***Schnur***

(Optional) Die Beschreibung des Berichts.

`owner:` ***Schnur***

Der Benutzername des Berichtsbesitzers.

`cc:` ***Reihe von Zeichenketten***

Die Liste der E-Mail-Adressen, die nicht in einer E-Mail-Gruppe enthalten sind, für den Empfang von Berichten.

`enabled:` ***Boolesch***

(Optional) Gibt an, ob der Bericht aktiviert ist.

`from:` ***Schnur***

Der Anfangszeitstempel des Zeitintervalls für den Berichtsinhalt, relativ zur aktuellen Uhrzeit, ausgedrückt in Millisekunden.

`until:` ***Schnur***

(Optional) Der Endzeitstempel des Zeitintervalls für den Berichtsinhalt, relativ zur aktuellen Uhrzeit, ausgedrückt in Millisekunden.

`email_subject:` ***Schnur***

(Optional) Der Inhalt der Betreffzeile für die Berichts-E-Mail.

`schedule:` ***Objekt***

(Optional) Das Objekt, das die Parameter enthält, die den geplanten Zeitraum für die Generierung und das Senden des Berichts angeben. Die Parameter sind im Abschnitt schedule_type unten definiert.

`type:` ***Schnur***

Die Art des Lieferplans für den Bericht.

Die folgenden Werte sind gültig:

- `hourly`
- `daily`
- `weekly`
- `monthly`

`at:` ***Reihe von Objekten***

(Optional) Die Liste der Objekte, die die Übermittlungsparameter für den Bericht angeben. Die Parameter sind im Abschnitt at_type unten definiert.

`by_day:` ***Reihe von Zeichenketten***

(Optional) Die Wochentage, an denen der Bericht gesendet werden soll.

Die folgenden Werte sind gültig:

- `mo`
- `tu`
- `we`
- `th`
- `fr`
- `sa`
- `su`

`on_day:` ***Zahl***

(Optional) Der Tag des Monats, an dem der Bericht ausgeführt werden soll.

`tz:` ***Schnur***

(Optional) Die Zeitzone, in der der Bericht gesendet werden soll.

`hour:` ***Zahl***

(Optional) Die Stunde, zu der der Bericht gesendet werden soll.

`minute:` ***Zahl***

(Optional) Die Minute, in der der Bericht gesendet werden soll.

`interval:` ***Schnur***

(Optional) Das Intervall kann previous_week, previous_month oder nichts sein.

Die folgenden Werte sind gültig:

- `previous_week`
- `previous_month`

`output:` ***Objekt***

Das Objekt, das die Parameter enthält, die das Ausgabeformat für den Bericht angeben. Die Parameter sind im Abschnitt format_type unten definiert.

`type:` ***Schnur***

Das Ausgabeformat für den Bericht.

Die folgenden Werte sind gültig:

- `pdf`

`width:` ***Schnur***

(Optional) Die Breitenoption für die Berichtsausgabe.

Die folgenden Werte sind gültig:

- `narrow`
- `medium`
- `wide`

`pagination:` ***Schnur***

(Optional) Das Paginierungsschema für die Berichtsausgabe.

Die folgenden Werte sind gültig:

- `per_region`

`theme:` ***Schnur***

(Optional) Das Anzeigedesign für die Berichtsausgabe.

Die folgenden Werte sind gültig:

- `light`
- `dark`
- `space`
- `contrast`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "cc": [],
    "description": "string",
    "email_subject": "string",
    "enabled": true,
    "from": "string",
    "name": "string",
    "output": {
        "type": "string",
        "width": "string",
        "pagination": "string",
        "theme": "string"
    },
    "owner": "string",
    "schedule": {
        "type": "string",
        "at": {
            "by_day": [],
            "on_day": 0,
            "tz": "string",
            "hour": 0,
            "minute": 0
        },
        "interval": "string"
    },
    "until": "string"
}
```

`POST /reports/{id}/queue`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.

`PATCH /reports/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.

`body:` ***Objekt***
Der Inhalt des Berichts.

`name:` ***Schnur***
Der Name des Berichts.

`description:` ***Schnur***
(Optional) Die Beschreibung des Berichts.

`owner:` ***Schnur***
Der Benutzername des Berichtsbesitzers.

`cc:` ***Reihe von Zeichenketten***
Die Liste der E-Mail-Adressen, die nicht in einer E-Mail-Gruppe enthalten sind, für den Empfang von Berichten.

`enabled:` ***Boolesch***
(Optional) Gibt an, ob der Bericht aktiviert ist.

`from:` ***Schnur***
Der Anfangszeitstempel des Zeitintervalls für den Berichtsinhalt, relativ zur aktuellen Uhrzeit, ausgedrückt in Millisekunden.

`until:` ***Schnur***
(Optional) Der Endzeitstempel des Zeitintervalls für den Berichtsinhalt, relativ zur aktuellen Uhrzeit, ausgedrückt in Millisekunden.

`email_subject:` ***Schnur***
(Optional) Der Inhalt der Betreffzeile für die Berichts-E-Mail.

`schedule:` ***Objekt***
(Optional) Das Objekt, das die Parameter enthält, die den geplanten Zeitraum für die Generierung und das Senden des Berichts angeben. Die Parameter sind im Abschnitt schedule_type unten definiert.

`type:` ***Schnur***
Die Art des Lieferplans für den Bericht.

Die folgenden Werte sind gültig:

- `hourly`
- `daily`
- `weekly`
- `monthly`

`at:` ***Reihe von Objekten***
(Optional) Die Liste der Objekte, die die Übermittlungsparameter für den Bericht angeben. Die Parameter sind im Abschnitt at_type unten definiert.

`by_day:` ***Reihe von Zeichenketten***

(Optional) Die Wochentage, an denen der Bericht gesendet werden soll.

Die folgenden Werte sind gültig:

- `mo`
- `tu`
- `we`
- `th`
- `fr`
- `sa`
- `su`

`on_day:` ***Zahl***

(Optional) Der Tag des Monats, an dem der Bericht ausgeführt werden soll.

`tz:` ***Schnur***

(Optional) Die Zeitzone, in der der Bericht gesendet werden soll.

`hour:` ***Zahl***

(Optional) Die Stunde, zu der der Bericht gesendet werden soll.

`minute:` ***Zahl***

(Optional) Die Minute, in der der Bericht gesendet werden soll.

`interval:` ***Schnur***

(Optional) Das Intervall kann previous_week, previous_month oder nichts sein.

Die folgenden Werte sind gültig:

- `previous_week`
- `previous_month`

`output:` ***Objekt***

Das Objekt, das die Parameter enthält, die das Ausgabeformat für den Bericht angeben. Die Parameter sind im Abschnitt format_type unten definiert.

`type:` ***Schnur***

Das Ausgabeformat für den Bericht.

Die folgenden Werte sind gültig:

- `pdf`

`width:` ***Schnur***

(Optional) Die Breitenoption für die Berichtsausgabe.

Die folgenden Werte sind gültig:

- `narrow`
- `medium`
- `wide`

`pagination:` ***Schnur***

(Optional) Das Paginierungsschema für die Berichtsausgabe.

Die folgenden Werte sind gültig:

- `per_region`

`theme:` ***Schnur***

(Optional) Das Anzeigedesign für die Berichtsausgabe.

Die folgenden Werte sind gültig:

- `light`
- `dark`
- `space`
- `contrast`

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "cc": [],
    "description": "string",
    "email_subject": "string",
    "enabled": true,
    "from": "string",
    "name": "string",
    "output": {
        "type": "string",
        "width": "string",
        "pagination": "string",
        "theme": "string"
    },
    "owner": "string",
    "schedule": {
        "type": "string",
        "at": {
            "by_day": [],
            "on_day": 0,
            "tz": "string",
            "hour": 0,
            "minute": 0
        },
        "interval": "string"
    },
    "until": "string"
}
```

`GET /reports/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für den Bericht.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "cc": [],
    "description": "string",
    "email_message": "string",
    "email_subject": "string",
    "enabled": true,
    "from": "string",
    "id": 0,
    "include_links": "string",
    "name": "string",
    "output": {},
    "owner": "string",
    "schedule": {},
    "until": "string"
}
```

`GET /reports/{id}/download`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "cc": [],
    "description": "string",
    "email_message": "string",
    "email_subject": "string",
    "enabled": true,
    "from": "string",
    "id": 0,
    "include_links": "string",
    "name": "string",
    "output": {},
    "owner": "string",
    "schedule": {},
    "until": "string"
}
```

`DELETE /reports/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.

`GET /reports/{id}/contents`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "dashboard_id": 0,
    "params": {},
    "type": "string"
}
```

`PUT /reports/{id}/contents`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.

`body:` ***Objekt***
Der Inhalt des Berichts.

`POST /reports/{id}/emailgroups/{group-id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.
`group-id:` ***Zahl***
Die eindeutige Kennung für die E-Mail-Gruppe.

`POST /reports/{id}/emailgroups`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.
`body:` ***Objekt***
Die Liste der E-Mail-Gruppen-IDs, die dem Bericht zugewiesen oder deren Zuweisung aufgehoben werden soll.
`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen
`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`GET /reports/{id}/emailgroups`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.

`DELETE /reports/{id}/emailgroups/{group-id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Bericht.
`group-id:` ***Zahl***
Die eindeutige Kennung für die E-Mail-Gruppe.

Konfiguration ausführen

Die laufende Konfigurationsdatei ist ein JSON-Dokument, das wichtige Systemkonfigurationsinformationen für das ExtraHop-System enthält.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
Holen Sie sich /runningconfig	Ruft die aktuell laufende Konfigurationsdatei ab.

Betrieb	Beschreibung
PUT /runningconfig	Ersetzt die aktuell laufende Konfigurationsdatei. Änderungen an der Konfigurationsdatei werden nicht automatisch gespeichert.
POST/runningconfig/save	Speichert die aktuellen Änderungen in der laufenden Konfigurationsdatei.
GET /runningconfig/saved	Rufen Sie die gespeicherte laufende Konfigurationsdatei ab.

Einzelheiten der Operation

`GET /runningconfig/saved`

Für diesen Vorgang gibt es keine Parameter.

`POST /runningconfig/save`

Für diesen Vorgang gibt es keine Parameter.

`GET /runningconfig`

Geben Sie die folgenden Parameter an.

`section:` ***Schnur***

(Optional) (Optional) Der spezifische Abschnitt der laufenden Konfigurationsdatei, den Sie abrufen möchten.

`PUT /runningconfig`

Geben Sie die folgenden Parameter an.

`body:` ***Schnur***

(Optional) Die laufende Konfigurationsdatei.

Software

Sie können sich eine Liste der Software ansehen, die das ExtraHop-System in Ihrem Netzwerk beobachtet hat.

Betrieb	Beschreibung
Holen Sie sich /software	Rufen Sie die vom ExtraHop-System beobachtete Software ab.
GET /software/ {id}	Rufen Sie die vom ExtraHop-System beobachtete Software anhand der ID ab.

Einzelheiten der Operation

`GET /software`

Geben Sie die folgenden Parameter an.

`software_type:` ***Schnur***
(Optional) Die Art der Software.

`name:` ***Schnur***
(Optional) Der Name der Software.

`version:` ***Schnur***
(Optional) Die Version der Software.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": "string",
    "name": "string",
    "software_type": "string",
    "version": "string"
}
```

`GET /software/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Software.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": "string",
    "name": "string",
    "software_type": "string",
    "version": "string"
}
```

TLS-Entschlüsselungsschlüssel

Mit dieser Ressource können Sie einen Entschlüsselungsschlüssel für Ihren Netzwerkverkehr hinzufügen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /ssldecryptkeys	Ruft alle TLS-Entschlüsselungsschlüssel ab.
POST /ssldecryptkeys	Erstellen Sie einen neuen TLS-Entschlüsselungsschlüssel.
LÖSCHE /ssldecryptkeys/ {id}	Entfernen Sie einen TLS-Schlüssel aus dem ExtraHop-System.
HOLEN SIE SICH /ssldecryptkeys/ {id}	Rufen Sie ein TLS-PEM und Metadaten ab.
PATCH /ssldecryptkeys/ {id}	Aktualisieren Sie einen vorhandenen TLS-Entschlüsselungsschlüssel.
GET /ssldecryptkeys/ {id} /protocols	Alle abrufen Protokolle einem TLS-Entschlüsselungsschlüssel zugewiesen.

Bedienung	Beschreibung
POST /ssldecryptkeys/ {id} /protocols	Erstellen Sie ein neues Protokoll für einen TLS-Entschlüsselungsschlüssel.
LÖSCHE /ssldecryptkeys/ {id} /protocols/ {Protokoll}	Löscht ein Protokoll aus einem TLS-Entschlüsselungsschlüssel.

Einzelheiten der Operation

`GET /ssldecryptkeys`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "cert_pem": "string",
    "enabled": true,
    "id": "string",
    "name": "string"
}
```

`POST /ssldecryptkeys`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Legt die angegebenen Eigenschaftswerte für den neuen SSL-Entschlüsselungsschlüssel fest.

`enabled:` ***Boolescher Wert***

Geben Sie an, ob dieser SSL-Entschlüsselungsschlüssel aktiv ist.

`name:` ***Schnur***

Der benutzerfreundliche Name für den SSL-Entschlüsselungsschlüssel.

`certificate:` ***Schnur***

Das mit diesem Entschlüsselungsschlüssel verknüpfte SSL-Zertifikat.

`private_key:` ***Schnur***

Der private SSL-Schlüssel, der den Verkehr entschlüsselt.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "certificate": "string",
    "enabled": true,
    "name": "string",
    "private_key": "string"
}
```

`PATCH /ssldecryptkeys/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wenden Sie die angegebenen Eigenschaftenaktualisierungen auf den SSL-Entschlüsselungsschlüssel an.

`id:` ***Schnur***

Die hexadezimale Darstellung des SHA-1-Hashs des SSL-Entschlüsselungsschlüssels. Die Zeichenfolge darf keine Trennzeichen enthalten.

`GET /ssldecryptkeys/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***

Die hexadezimale Darstellung des SHA-1-Hashs des SSL-Entschlüsselungsschlüssels. Die Zeichenfolge darf keine Trennzeichen enthalten.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "cert_pem": "string",
    "enabled": true,
    "id": "string",
    "name": "string"
}
```

`DELETE /ssldecryptkeys/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***

Die hexadezimale Darstellung des SHA-1-Hashs des SSL-Entschlüsselungsschlüssels. Die Zeichenfolge darf keine Trennzeichen enthalten.

`GET /ssldecryptkeys/{id}/protocols`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***

Die hexadezimale Darstellung des SHA-1-Hashs des SSL-Entschlüsselungsschlüssels. Die Zeichenfolge darf keine Trennzeichen enthalten.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "port": 0,
    "protocol": "string"
}
```

`POST /ssldecryptkeys/{id}/protocols`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Der Hauptteil des Protokoll.

`protocol:` ***Schnur***

Der Name des Protokoll in Kleinbuchstaben.

`port:` ***Zahl***

Der Port, in dem der Verkehr überwacht werden soll.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "port": 0,
    "protocol": "string"
}
```

`id:` ***Schnur***

Die eindeutige Kennung für den SSL-Entschlüsselungsschlüssel.

`DELETE /ssldecryptkeys/{id}/protocols/{protocol}`

Geben Sie die folgenden Parameter an.

`protocol:` ***Schnur***

Der Name des Protokoll in Kleinbuchstaben.

`id:` ***Schnur***

Die hexadezimale Darstellung des SHA-1-Hashs des SSL-Entschlüsselungsschlüssels. Die Zeichenfolge darf keine Trennzeichen enthalten.

`port:` ***Zahl***

(Optional) Entfernen Sie nur die Protokolle, die diesem Port zugewiesen sind.

Unterstützungspaket

Ein Support Pack ist eine Datei, die vom ExtraHop Support bereitgestellte Konfigurationsanpassungen enthält.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
HOLEN SIE SICH /supportpacks	Rufen Sie Metadaten zu allen Support Packs ab.
POST /supportpacks	Laden Sie ein Support Pack hoch und führen Sie es aus.
POST /supportpacks/execute	Führen Sie ein neues Support Pack aus.
GET /supportpacks/queue/ {id}	Überprüfen Sie den Status eines laufenden, laufenden Support Packs.
GET /supportpacks/ {Dateiname}	Laden Sie ein vorhandenes Support Pack anhand des Dateinamens herunter.

Einzelheiten der Operation

`GET /supportpacks/queue/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***

Die eindeutige Kennung für das laufende Support Pack.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "created_time": 0,
    "filename": "string",
```

```
    "size": "string"
}
```

`GET /supportpacks/{filename}`

Geben Sie die folgenden Parameter an.

`filename:` ***Schnur***
Der Name des herunterzuladenden Support Packs.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "created_time": 0,
    "filename": "string",
    "size": "string"
}
```

`POST /supportpacks/execute`

`GET /supportpacks`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "created_time": 0,
    "filename": "string",
    "size": "string"
}
```

`POST /supportpacks`

Geben Sie die folgenden Parameter an.

`file:` ***Dateiname***
Der Dateiname für das Support Pack.

Tag

Mithilfe von Geräte-Tags können Sie ein Gerät oder eine Gruppe von Geräten anhand eines Merkmals zuordnen.

Sie könnten zum Beispiel alle Ihre taggen HTTP Server oder kennzeichnet alle Geräte, die sich in einem gemeinsamen Subnetz befinden. Weitere Informationen finden Sie unter Taggen Sie ein Gerät über die REST-API.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /tags	Ruft alle Tags ab.
POST /Schlagworte	Erstellen Sie ein neues Tag.
/tags/ {id} LÖSCHEN	Löscht ein bestimmtes Tag.
GET /tags/ {id}	Ruft ein bestimmtes Tag ab.

Betrieb	Beschreibung
PATCH /tags/ {id}	Wenden Sie Aktualisierungen auf ein bestimmtes Tag an.
GET /tags/ {id} /devices	Ruft alle Geräte ab, die einem bestimmten Tag zugewiesen sind.
POST /tags/ {id} /Geräte	Weisen Sie Geräten ein bestimmtes Tag zu und heben Sie die Zuweisung auf.
LÖSCHEN /tags/ {id} /devices/ {child-id}	Heben Sie die Zuweisung eines Gerät zu einem bestimmten Tag auf.
POST /tags/ {id} /devices/ {child-id}	Weisen Sie ein Gerät einem bestimmten Tag zu.

Einzelheiten der Operation

`GET /tags`

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "id": 0,
    "mod_time": 0,
    "name": "string"
}
```

`POST /tags`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wendet die angegebenen Eigenschaftswerte auf das neue Tag an.

`name:` ***Schnur***

Der Zeichenkettenwert für das Tag.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "name": "string"
}
```

`GET /tags/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das Tag.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "id": 0,
    "mod_time": 0,
    "name": "string"
}
```

`DELETE /tags/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Tag.

`PATCH /tags/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Wendet die angegebenen Eigenschaftswertaktualisierungen auf das Tag an.

`id:` ***Zahl***
Die eindeutige Kennung für das Tag.

`GET /tags/{id}/devices`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Tag.

`POST /tags/{id}/devices`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Listen mit eindeutigen Kennungen für Gerät zum Zuweisen und Aufheben der Zuweisung.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für das Tag.

`POST /tags/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Gerät.

`id:` ***Zahl***
die eindeutige Kennung für das Tag.

`DELETE /tags/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***

Die eindeutige Kennung für das Gerät.

`id:` ***Zahl***

Die eindeutige Kennung für das Tag.

Erfassung von Bedrohungen

Mit der Threat Collection-Ressource können Sie kostenlose und kommerzielle Inhalte hochladen Sammlungen von Bedrohungen wird von der Security Community für Ihr RevealX-System angeboten.

Wichtig: STIX-Datei-Uploads sind jetzt veraltet und werden voraussichtlich im März 2025 entfernt.

- Sie müssen Bedrohungssammlungen einzeln auf Ihre Command-Appliance oder RevealX 360 hochladen und auf alle verbundenen Sensoren.
- Benutzerdefinierte Bedrohungssammlungen müssen in Structured Threat Information eXpression (STIX) als TAR.GZ -Dateien formatiert werden. RevealX unterstützt derzeit die STIX-Versionen 1.0 - 1.2.
- Sie können Bedrohungssammlungen direkt auf die RevealX 360-Systeme hochladen, um sie selbst zu verwalten Sensoren. Wenden Sie sich an den ExtraHop-Support, um eine Bedrohungssammlung auf ExtraHop-Managed hochzuladen Sensoren.
- Die maximale Anzahl an Observables, die eine Bedrohungssammlung enthalten kann, hängt von Ihrer Plattform und Lizenz ab. Kontaktieren Sie Ihren ExtraHop-Vertreter für weitere Informationen.

Hinweis: Dieses Thema gilt nur für ExtraHop RevealX Premium und Ultra.

Informationen zum Hochladen von STIX-Dateien über das ExtraHop-System finden Sie unter Laden Sie STIX-Dateien über die REST-API hoch.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /threatcollections	Rufen Sie alle Bedrohungssammlungen ab.
POST /threat collections	Erstellen Sie eine neue Bedrohungssammlung.
/threatcollections/ {id} LÖSCHEN	Löschen Sie eine Bedrohungssammlung.
PUT /threatcollections/ {id}	Laden Sie eine neue Bedrohungssammlung hoch. ExtraHop unterstützt derzeit die STIX-Versionen 1.0 - 1.2. **Hinweis:** Wenn auf dem ExtraHop-System bereits eine Bedrohungssammlung mit demselben Namen existiert, wird die bestehende Bedrohungssammlung überschrieben.
GET /threatcollections/ {id} /observables	Ruft die Anzahl der STIX-Observables ab, die aus einer Bedrohungssammlung geladen wurden, z. B. IP-Adresse, Hostname oder URI.

Einzelheiten der Operation

`GET /threatcollections`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "id": 0,
    "last_updated": 0,
    "name": "string",
    "observables": 0,
    "user_key": "string"
}
```

`POST /threatcollections`

Geben Sie die folgenden Parameter an.

`user_key:` ***Schnur***
(Optional) Die vom Benutzer angegebene Kennung für die Bedrohungssammlung. Wenn dieser Parameter nicht angegeben ist, wird der Name der Bedrohungssammlung für diesen Wert ohne Leerzeichen oder Satzzeichen festgelegt.

`name:` ***Schnur***
Der Name für die Bedrohungssammlung.

`file:` ***Dateiname***
Der Dateiname für die Bedrohungssammlung.

`PUT /threatcollections/~{userKey}`

Geben Sie die folgenden Parameter an.

`userKey:` ***Schnur***
Die vom Benutzer angegebene Kennung für die Bedrohungssammlung.

`name:` ***Schnur***
(Optional) Der Name für die Bedrohungssammlung.

`file:` ***Dateiname***
(Optional) Der Dateiname für die Bedrohungssammlung.

`DELETE /threatcollections/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Bedrohungssammlung.

`GET /threatcollections/{id}/observables`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Bedrohungssammlung.

Auslösen

Trigger sind benutzerdefinierte Skripts, die bei einem vordefinierten Ereignis eine Aktion ausführen.

Sie können beispielsweise einen Auslöser schreiben, um jedes Mal eine benutzerdefinierte Metrik Datensatz, wenn HTTP Eine Anfrage erfolgt, oder klassifizieren Sie den Datenverkehr für einen bestimmten

Server als Anwendungsserver. Weitere Informationen finden Sie in der Trigger-API-Referenz. Zusätzliche Implementierungshinweise zu erweiterten Optionen finden Sie unter Erweiterte Trigger-Optionen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Bedienung	Beschreibung
GET /triggert	Ruft alle Trigger ab.
POST /Trigger	Erstellen Sie einen neuen Auslöser.
POST /triggers/externe Daten	Sendet Daten an die Trigger-API, indem das Ereignis EXTERNAL_DATA ausgeführt wird. Sie können auf die Daten zugreifen über ExternalData Trigger-Klasse. **Hinweis:** Dieser Vorgang ist für Command-Appliances oder RevealX 360 nicht verfügbar.
LÖSCHE /triggers/ {id}	Löscht einen bestimmten Bezeichner.
GET /triggers/ {id}	Ruft einen bestimmten Auslöser anhand einer eindeutigen Kennung ab.
PATCH /trigger/ {id}	Aktualisieren Sie einen vorhandenen Auslöser.
GET /triggers/ {id} /devicegroups	Alle abrufen Gerätegruppen die einem bestimmten Auslöser zugewiesen sind.
POST /triggers/ {id} /devicegroups	Weisen Sie Gerätegruppen einen bestimmten Auslöser zu und heben Sie die Zuweisung auf.
LÖSCHE /triggers/ {id} /devicegroups/ {child-id}	Heben Sie die Zuweisung einer Gerätegruppe zu einem bestimmten Auslöser auf.
POST /triggers/ {id} /devicegroups/ {child-id}	Ordnen Sie eine Gerätegruppe einem bestimmten Auslöser zu.
GET /triggers/ {id} /devices	Ruft alle Geräte ab, die einem bestimmten Auslöser zugewiesen sind.
POST /trigger/ {id} /geräte	Weisen Sie Geräten einen bestimmten Auslöser zu und heben Sie die Zuweisung auf.
LÖSCHE /triggers/ {id} /devices/ {child-id}	Hebt die Zuweisung eines Gerät zu einem bestimmten Auslöser auf.
POST /trigger/ {id} /devices/ {child-id}	Ordnen Sie ein Gerät einem bestimmten Auslöser zu.

Einzelheiten der Operation

`GET /triggers`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "apply_all": true,
    "author": "string",
    "debug": true,
    "description": "string",
```

```
    "disabled": true,
    "event": "string",
    "events": [
        "string"
    ],
    "hints": {},
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "script": "string"
}
```

`DELETE /triggers/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für den Auslöser.

`POST /triggers/externaldata`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Das Objekt, das die zu sendenden Daten enthält, wird durch das Ereignis EXTERNAL_DATA ausgelöst.

`type:` ***Schnur***

Ein Zeichenkettenbezeichner, der die im Body-Parameter enthaltenen Daten beschreibt. Sie könnten beispielsweise „Phantomdaten“ für Daten angeben, die von der Phantom SOAR-Plattform gesendet werden.

`body:` ***Objekt***

Die Daten, an die gesendet werden sollen, werden durch das Ereignis EXTERNAL_DATA ausgelöst. Auf diese Daten kann im Auslöser mit der Eigenschaft 'ExternalData.Body' zugegriffen werden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "body": {},
    "type": "string"
}
```

`POST /triggers`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Eigenschaftswerte für den neuen Auslöser.

`name:` ***Schnur***

Der freundliche Name für den Auslöser.

`description:` ***Schnur***

(Optional) Eine optionale Beschreibung des Auslöser.

`author:` ***Schnur***

Der Name des Erstellers des Auslöser.

`script:` ***Schnur***
Der JavaScript-Inhalt des Auslöser.

`event:` ***Schnur***
(Optional) Veraltet. Ersetzt durch das Feld Ereignisse.

`events:` ***Reihe von Zeichenketten***
Die Liste der Ereignisse, bei denen der Auslöser ausgeführt wird, ausgedrückt als JSON-Array.

`disabled:` ***Boolesch***
Gibt an, ob der Auslöser ausgeführt werden kann.

`debug:` ***Boolesch***
Gibt an, ob Debug-Anweisungen für den Auslöser gedruckt werden.

`apply_all:` ***Boolesch***
Gibt an, ob der Auslöser für alle relevanten Ressourcen gilt.

`hints:` ***Objekt***
Optionen, die auf ausgewählten Triggerereignissen basieren. Weitere Informationen zum Hints-Objekt finden Sie in der REST-API-Leitfaden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "apply_all": true,
    "author": "string",
    "debug": true,
    "description": "string",
    "disabled": true,
    "event": "string",
    "events": [
        "string"
    ],
    "hints": {},
    "name": "string",
    "script": "string"
}
```

`PATCH /triggers/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Der Eigenschaftswert wird für den Auslöser aktualisiert.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`GET /triggers/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "apply_all": true,
    "author": "string",
    "debug": true,
```

```
    "description": "string",
    "disabled": true,
    "event": "string",
    "events": [
        "string"
    ],
    "hints": {},
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "script": "string"
}
```

`GET /triggers/{id}/devicegroups`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`POST /triggers/{id}/devicegroups`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Eine Liste eindeutiger Identifikatoren für Gerätegruppen, die einem Auslöser zugewiesen sind oder nicht.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`POST /triggers/{id}/devicegroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`DELETE /triggers/{id}/devicegroups/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für die Gerätegruppe.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`GET /triggers/{id}/devices`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`POST /triggers/{id}/devices`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Eine Liste eindeutiger Identifikatoren für Geräte, die einem Auslöser zugewiesen sind oder nicht zugewiesen sind.

`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen

`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`POST /triggers/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Gerät.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

`DELETE /triggers/{id}/devices/{child-id}`

Geben Sie die folgenden Parameter an.

`child-id:` ***Zahl***
Die eindeutige Kennung für das Gerät.

`id:` ***Zahl***
Die eindeutige Kennung für den Auslöser.

Erweiterte Trigger-Optionen

Erweiterte Triggeroptionen sind Konfigurationsoptionen, die Sie abhängig von den mit dem Auslöser verknüpften Systemereignissen festlegen können. Sie können beispielsweise die Anzahl der Payload-Bytes konfigurieren, auf denen gepuffert werden soll HTTP Ereignisse anfragen.

Erweiterte Optionen sind enthalten in `hints` Objekt der Trigger-Ressource , wie im folgenden Beispiel gezeigt:

```
"hints": {
"flowClientPortMin": null,
"flowClientBytes": 16384,
"flowClientPortMax": null,
"flowServerBytes": 16384,
"flowPayloadTurn": true,
"flowServerPortMin": 135,
"flowServerPortMax": 49155
}
```

In der folgenden Tabelle werden die verfügbaren erweiterten Optionen und die entsprechenden Ereignisse beschrieben:

Wahl	Beschreibung	Anwendbare Ereignisse
`"snaplen": number`	Gibt die Anzahl der Byte an, die pro Paket erfasst werden sollen, bis zu einem Maximum von 65535. Die Erfassung beginnt mit dem ersten Byte im Paket. Geben Sie diese Option nur an, wenn das Triggerskript Pakete erfasst. Ein Wert von `0` konfiguriert den Auslöser so, dass er die maximale Anzahl von Byte für jedes Paket erfasst.	Alle Veranstaltungen außer: • ALERT_RECORD_COMMIT • METRIC_CYCLE_BEGIN • ENDE DES METRISCHEN ZYKLUS • FLOW_REPORT • NEUE_ANWENDUNG • NEUES_GERÄT • SITZUNG ABLAUFEN
`"payloadBytes": number`	Gibt die Mindestanzahl der zu puffernden Payload-Bytes an.	• CIFS_REQUEST • CIFS_RESPONSE • HTTP_REQUEST • HTTP_RESPONSE • ICA_TICK
`"clipboardBytes": number`	Gibt die Anzahl der Byte an, die bei einer Übertragung in die Citrix-Zwischenablage zwischengespeichert werden sollen.	• ICA_TICK
`"cycle": [30sec, 5min, 1hr, 24hr]`	Gibt die Länge des Metrik Zyklus in Sekunden an.	• METRIC_CYCLE_BEGIN • ENDE DES METRISCHEN ZYKLUS • METRIC_RECORD_COMMIT
`"metricTypes": string`	Gibt den Metriktyp anhand des Rohmetriknamens an, z. B. extrahop.device.http_server.	• ALERT_RECORD_COMMIT • METRIC_RECORD_COMMIT
`"flowPayloadTurn": boolean`	Aktiviert die PCAP bei jedem Flow-Turn. Die Per-Turn-Analyse analysiert kontinuierlich die Kommunikation zwischen zwei Endpunkten, um	• SSL_PAYLOAD • TCP_PAYLOAD

Wahl	Beschreibung	Anwendbare Ereignisse
	einen einzelnen Nutzdatenpunkt aus dem Fluss zu extrahieren. Wenn diese Option aktiviert ist, werden alle angegebenen Werte für `flowClientString` und `flowServerString` Optionen werden ignoriert.	
`"flowClientPortMin": number`	Gibt die Mindestportnummer des Client Portbereich. Gültige Werte sind `0` zu `65535`. Ein Wert von `0` spezifiziert die Übereinstimmung eines beliebigen Ports.	• SSL_PAYLOAD • TCP_PAYLOAD • UDP_PAYLOAD
`"flowClientPortMax": number`	Gibt die maximale Portnummer des Client Portbereich. Gültige Werte sind `0` zu `65535`. Jeder für diese Option angegebene Wert wird ignoriert, wenn der Wert von `flowClientPortMin` Option ist `0`.	• SSL_PAYLOAD • TCP_PAYLOAD • UDP_PAYLOAD
`"flowClientBytes": number`	Gibt die Anzahl der Client Bytes zum Puffer. Der Wert dieser Option kann nicht auf gesetzt werden `0` wenn der Wert der `flowServerBytes` Option ist auch gesetzt auf `0`.	• SSL_PAYLOAD • TCP_PAYLOAD
`"flowClientString": string`	Definiert die Formatzeichenfolge von Client zu verarbeitende Daten. Jeder für diese Option angegebene Wert wird ignoriert, wenn `flowPayloadTurn` Option ist aktiviert.	• SSL_PAYLOAD • TCP_PAYLOAD • UDP_PAYLOAD
`"flowServerPortMin": number`	Gibt die Mindestportnummer des Serverportbereichs an. Gültige Werte sind `0` nach `65535`. Ein Wert von `0` spezifiziert die Übereinstimmung eines beliebigen Ports.	• SSL_PAYLOAD • TCP_PAYLOAD • UDP_PAYLOAD
`"flowServerPortMax": number`	Gibt die maximale Portnummer des Serverportbereichs an. Gültige Werte sind `0` zu `65535`.	• SSL_PAYLOAD • TCP_PAYLOAD • UDP_PAYLOAD

Wahl	Beschreibung	Anwendbare Ereignisse
	Jeder für diese Option angegebene Wert wird ignoriert, wenn der Wert von `flowServerPortMin` Option ist `0`.	
`"flowServerBytes": number`	Gibt die Anzahl der Server-Bytes an, die gepuffert werden sollen. Der Wert dieser Option kann nicht auf gesetzt werden `0` wenn der Wert der `flowClientBytes` Option ist auch gesetzt auf `0`.	• SSL_PAYLOAD • TCP_PAYLOAD
`"flowServerString": string`	Gibt die Formatzeichenfolge der zu verarbeitenden Serverdaten an. Gibt bei einer übereinstimmenden Zeichenfolge das gesamte Paket zurück. Jeder für diese Option angegebene Wert wird ignoriert, wenn `flowPayloadTurn` Option ist aktiviert.	• SSL_PAYLOAD • TCP_PAYLOAD • UDP_PAYLOAD
`"flowUdpAll": boolean`	Ermöglicht die Erfassung aller UDP-Datagramme.	• UDP_PAYLOAD
`"fireClassifyOnExpiration": boolean`	Ermöglicht die Ausführung des Ereignis nach Ablauf, um Metriken für Flows zu sammeln, die vor Ablauf nicht klassifiziert wurden.	• FLOW_CLASSIFY

Nutzer

Mit der Benutzerressource können Sie die Liste der Benutzer, die Zugriff auf das ExtraHop-System haben, und die Berechtigungsstufen für diese Benutzer erstellen und verwalten.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /users	Ruft alle Benutzer ab.
POST /Benutzer	Erstellen Sie einen neuen Benutzer.
LÖSCHE /users/ {username}	Löscht einen bestimmten Benutzer.
GET /users/ {Nutzername}	Rufen Sie einen bestimmten Benutzer ab.
PATCH /users/ {Nutzername}	Aktualisieren Sie die Einstellungen für einen bestimmten Benutzer.
GET /users/ {username} /apikeys	Ruft alle API-Schlüssel für einen bestimmten Benutzer ab.
GET /users/ {username} /apikeys/ {keyid}	Rufen Sie Informationen über einen bestimmten API-Schlüssel und Benutzer ab.

Einzelheiten der Operation

`GET /users`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "date_joined": "string",
    "effective_roles": {},
    "eh_account_team": true,
    "enabled": true,
    "granted_roles": {},
    "last_ui_login_time": "string",
    "name": "string",
    "type": "string",
    "username": "string"
}
```

`POST /users`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Benutzerkontoeinstellungen.

`enabled:` ***Boolesch***

(Optional) Gibt an, ob sich der Benutzer beim ExtraHop-System anmelden kann.

`name:` ***Schnur***

Der freundliche Name für den Benutzer.

`username:` ***Schnur***

Der Anmeldename für den Benutzer.

`password:` ***Schnur***

Das Passwort für den Benutzer. Passwörter müssen die in den Administrationseinstellungen konfigurierten Anforderungen erfüllen.

`granted_roles:` ***Objekt***

(Optional) Die Rechte für den Benutzer. Unterstützte Berechtigungsstufen werden in der beschrieben REST-API-Leitfaden.

`create_apikey:` ***Boolesch***

(Optional) Generieren Sie einen neuen API-Schlüssel für den erstellten Benutzer und geben Sie ihn zurück.

`type:` ***Schnur***

(Optional) Die Authentifizierungsmethode, mit der sich dieser Benutzer anmeldet.

Die folgenden Werte sind gültig:

- `local`
- `remote`

`eh_account_team:` ***Boolesch***

Zeigt einen Benutzer des ExtraHop Account Teams an, der über ExtraHop Cloud Services auf das ExtraHop-System zugreift.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
```

```
    "create_apikey": true,
    "eh_account_team": true,
    "enabled": true,
    "granted_roles": {},
    "name": "string",
    "password": "string",
    "type": "string",
    "username": "string"
}
```

`GET /users/{username}`

Geben Sie die folgenden Parameter an.

`username:` ***Schnur***
Der Name des Benutzers.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "date_joined": "string",
    "effective_roles": {},
    "eh_account_team": true,
    "enabled": true,
    "granted_roles": {},
    "last_ui_login_time": "string",
    "name": "string",
    "type": "string",
    "username": "string"
}
```

`PATCH /users/{username}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***
Die Benutzerkontoeinstellungen.

`enabled:` ***Boolesch***
(Optional) Gibt an, ob sich der Benutzer beim ExtraHop-System anmelden kann.

`name:` ***Schnur***
(Optional) Der benutzerfreundliche Name für den Benutzer.

`password:` ***Schnur***
(Optional) Das Passwort für den Benutzer. Passwörter müssen die in den Administrationseinstellungen konfigurierten Anforderungen erfüllen.

`granted_roles:` ***Objekt***
(Optional) Die Rechte für den Benutzer. Unterstützte Berechtigungsstufen werden in der beschrieben REST-API-Leitfaden.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "enabled": true,
    "granted_roles": {},
    "name": "string",
    "password": "string"
}
```

`username:` ***Schnur***
Der Name des Benutzers.

`DELETE /users/{username}`

Geben Sie die folgenden Parameter an.

`username:` ***Schnur***
Der Name des Benutzers.
`dest_user:` ***Schnur***
(Optional) Der Benutzer, an den Anpassungen übertragen werden. Wenn dieser Parameter angegeben ist, werden alle Dashboards, Sammlungen und Aktivitätskarten, die dem gelöschten Benutzer gehören, an diesen Benutzer übertragen.

`GET /users/{username}/apikeys`

Geben Sie die folgenden Parameter an.

`username:` ***Schnur***
Der Name des Benutzers.

`GET /users/{username}/apikeys/{keyid}`

Geben Sie die folgenden Parameter an.

`keyid:` ***Schnur***
Die ID des API-Schlüssels.
`username:` ***Schnur***
Der Name des Benutzers.

Benutzergruppe

Mit der Benutzergruppenressource können Sie Benutzergruppen und ihre Dashboard-Freigabezuordnungen verwalten und aktualisieren.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
GET /usergroups	Ruft alle Benutzergruppen ab.
POST /Benutzergruppen	Erstellen Sie eine neue Benutzergruppe.
POST /usergroups/refresh	Fragen Sie LDAP nach den neuesten Benutzermitgliedschaften für alle Remote-Benutzergruppen ab.
/usergroups/ {id} LÖSCHEN	Löscht eine bestimmte Benutzergruppe.
GET /usergroups/ {id}	Rufen Sie eine bestimmte Benutzergruppe ab.
PATCH /Benutzergruppen/ {id}	Aktualisieren Sie eine bestimmte Benutzergruppe.
/usergroups/ {id} /associations LÖSCHEN	Löschen Sie alle Verknüpfungen zum Teilen von Dashboard mit einer bestimmten Benutzergruppe.
GET /usergroups/ {id} /members	Ruft alle Mitglieder einer bestimmten Benutzergruppe ab.

Betrieb	Beschreibung
PATCH /usergroups/ {id} /members	Weisen Sie einer Benutzergruppe Benutzer zu oder heben Sie deren Zuweisung auf.
PUT /usergroups/ {id} /members	Ersetzen Sie Benutzergruppenzuweisungen.
POST /usergroups/ {id} /refresh	Fragen Sie LDAP nach der letzten Benutzermitgliedschaft einer bestimmten Remote-Benutzergruppe ab.

Einzelheiten der Operation

`GET /usergroups`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "display_name": "string",
    "enabled": true,
    "id": "string",
    "is_remote": true,
    "last_sync_time": 0,
    "name": "string",
    "rights": []
}
```

`POST /usergroups`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Die Eigenschaften der Benutzergruppe.

`name:` ***Zeichenfolge***

Der Name der Benutzergruppe.

`enabled:` ***Boolesch***

Gibt an, ob die Benutzergruppe aktiviert ist.

Geben Sie den Body-Parameter im folgenden JSON-Format an.

```
{
    "enabled": true,
    "name": "string"
}
```

`PATCH /usergroups/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Der Eigenschaftswert wird für die spezifische Benutzergruppe aktualisiert.

`id:` ***Zeichenfolge***

Der eindeutige Bezeichner für die Benutzergruppe.

`GET /usergroups/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zeichenfolge***
Der eindeutige Bezeichner für die Benutzergruppe.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "display_name": "string",
    "enabled": true,
    "id": "string",
    "is_remote": true,
    "last_sync_time": 0,
    "name": "string",
    "rights": []
}
```

`DELETE /usergroups/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zeichenfolge***
Der eindeutige Bezeichner für die Benutzergruppe.

`DELETE /usergroups/{id}/associations`

Geben Sie die folgenden Parameter an.

`id:` ***Zeichenfolge***
Der eindeutige Bezeichner für die Benutzergruppe.

`GET /usergroups/{id}/members`

Geben Sie die folgenden Parameter an.

`id:` ***Zeichenfolge***
Der eindeutige Bezeichner für die Benutzergruppe.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "users": {}
}
```

`POST /usergroups/refresh`

Für diesen Vorgang gibt es keine Parameter.

`POST /usergroups/{id}/refresh`

Geben Sie die folgenden Parameter an.

`id:` ***Zeichenfolge***
Der eindeutige Bezeichner für die Benutzergruppe.

`PATCH /usergroups/{id}/members`

Geben Sie die folgenden Parameter an.

`id:` ***Zeichenfolge***

Der eindeutige Bezeichner für die Benutzergruppe.

`body:` ***Zeichenfolge***

Ein Objekt, das angibt, welche Benutzer zugewiesen oder deren Zuweisung aufgehoben werden soll. Jeder Schlüssel muss ein Benutzername sein und jeder Wert muss entweder „Mitglied“ oder Null sein. Zum Beispiel weist {"Alice“: „member“, „Bob“: null} Alice der Gruppe zu und hebt Bob von der Gruppe ab.

`PUT /usergroups/{id}/members`

Geben Sie die folgenden Parameter an.

`id:` ***Zeichenfolge***

Der eindeutige Bezeichner für die Benutzergruppe.

`body:` ***Zeichenfolge***

Ein Objekt, das angibt, welche Benutzer der Gruppe zugewiesen sind. Jeder Schlüssel muss ein Benutzername sein und jeder Wert muss „Mitglied“ sein. Zum Beispiel weist {"Alice“: „member“, „Bob“: „member"} Alice und Bob als einzige Mitglieder der Gruppe zu.

VLAN

Virtuelle LANs sind logische Gruppierungen von Datenverkehr oder Geräten im Netzwerk.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
Holen Sie sich /vlans	Alle VLANs abrufen
GET /vlans/ {id}	Rufen Sie ein bestimmtes VLAN ab.
PATCH /vlans/ {id}	Aktualisieren Sie ein bestimmtes VLAN.

Einzelheiten der Operation

`GET /vlans`

Für diesen Vorgang gibt es keine Parameter.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "network_id": 0,
    "node_id": 0,
    "vlanid": 0
}
```

`GET /vlans/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***

Die eindeutige Kennung für das VLAN.

Wenn die Anfrage erfolgreich ist, gibt das ExtraHop-System ein Objekt im folgenden Format zurück.

```
{
    "description": "string",
    "id": 0,
    "mod_time": 0,
    "name": "string",
    "network_id": 0,
    "node_id": 0,
    "vlanid": 0
}
```

`PATCH /vlans/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Wenden Sie die angegebenen Eigenschaftswertaktualisierungen auf das VLAN an.

`id:` ***Zahl***

Die eindeutige Kennung für das VLAN.

Beobachtungsliste

Um sicherzustellen, dass für ein Asset, z. B. ein wichtiger Server, eine Datenbank oder ein Laptop, die erweiterte Analyse garantiert ist, können Sie dieses Gerät zur Beobachtungsliste hinzufügen.

Hinweis: Wenn Sie der Beobachtungsliste mehrere Geräte hinzufügen möchten, sollten Sie in Erwägung ziehen, eine Gerätegruppe zu erstellen und diese Gruppe dann für Erweiterte Analyse zu priorisieren.

Hier sind wichtige Überlegungen zur Beobachtungsliste:

- Die Beobachtungsliste gilt nur für Erweiterte Analyse.
- Die Beobachtungsliste kann so viele Geräte enthalten, wie es die Erweiterte Analyse Analysis-Kapazität zulässt, die durch Ihre Lizenz bestimmt wird.
- Ein Gerät bleibt auf der Beobachtungsliste, unabhängig davon, ob es inaktiv oder aktiv ist. Damit das ExtraHop-System Erweiterte Analyse Analysis-Metriken erfassen kann, muss ein Gerät aktiv sein.

Weitere Informationen zu Erweiterte Analyse finden Sie unter Analysestufen.

In der folgenden Tabelle sind alle Operationen aufgeführt, die Sie mit dieser Ressource ausführen können:

Betrieb	Beschreibung
/watchlist/device/ {id} LÖSCHEN	Entferne ein Gerät von der Beobachtungsliste.
POST /watchlist/device/ {id}	Fügen Sie ein Gerät zur Beobachtungsliste.
GET /watchlist/devices	Rufen Sie alle Geräte ab, die sich in der Beobachtungsliste befinden.
POST /watchliste/devices	Geräte zur Beobachtungsliste hinzufügen oder daraus entfernen.

Einzelheiten der Operation

`GET /watchlist/devices`

Für diesen Vorgang gibt es keine Parameter.

`POST /watchlist/device/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät.

`DELETE /watchlist/device/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Zahl***
Die eindeutige Kennung für das Gerät.

`POST /watchlist/devices`

Geben Sie die folgenden Parameter an.

`assignments:` ***Objekt***
Eine Liste von Geräten, die zur Beobachtungsliste hinzugefügt oder daraus entfernt werden sollen.
`assign:` ***Reihe von Zahlen***
IDs der zuzuweisenden Ressourcen
`unassign:` ***Reihe von Zahlen***
IDs der Ressourcen, deren Zuweisung aufgehoben werden soll

Geben Sie den Zuweisungsparameter im folgenden JSON-Format an.

```
{
    "assign": [],
    "unassign": []
}
```

ExtraHop REST-API-Beispiele

Die folgenden Beispiele zeigen gängige REST-API-Operationen.

- Ändern Sie einen Dashboard-Besitzer über die REST-API
- Extrahieren Sie die Geräteliste über die REST-API
- Erstellen und Zuweisen eines Geräte-Tags über die REST-API
- Abfragen von Metriken zu einem bestimmten Gerät über die REST-API
- Ein Objekt über die REST-API erstellen, abrufen und löschen
- Das Datensatzprotokoll abfragen

Aktualisieren Sie die ExtraHop-Firmware über die REST-API

Sie können Upgrades der Firmware auf Ihren ExtraHop-Appliances über die ExtraHop REST API automatisieren. Dieses Handbuch enthält Anweisungen zum Upgrade über den REST API Explorer, einen cURL-Befehl und ein Python-Skript.

Hinweis: Wenn Ihr Gerät mit ExtraHop Cloud Services verbunden ist, können Sie den Upgrade-Prozess vereinfachen, indem Sie sich die verfügbaren Firmware-Versionen ansehen und Firmware direkt von ExtraHop Cloud Services auf das System herunterladen. Weitere Informationen finden Sie unter Aktualisieren Sie die ExtraHop-Firmware über die REST-API mit ExtraHop Cloud Services.

Der Firmware-Upgrade-Prozess ist zwar bei allen ExtraHop-Appliances ähnlich, bei einigen Appliances sind jedoch zusätzliche Überlegungen oder Schritte erforderlich, die Sie berücksichtigen müssen, bevor Sie die Firmware in Ihrer Umgebung installieren. Wenn Sie Hilfe bei Ihrem Upgrade benötigen, wenden Sie sich an den ExtraHop-Support.

Alle Geräte müssen die folgenden Anforderungen erfüllen:

- Die Firmware-Version muss mit Ihrem Gerätemodell kompatibel sein.
- Die Firmware-Version auf Ihrem Gerät muss von der Upgrade-Version unterstützt werden.
- Auf Befehlsgeräten muss eine Firmware ausgeführt werden, die größer oder gleich der Firmware der angeschlossenen Geräte ist.
- Auf Discover-Appliances muss eine Firmware ausgeführt werden, die größer oder gleich der Firmware der verbundenen Explore and Trace-Appliances ist.

Wenn Ihr Einsatz nur eine umfasst Sensor, weiter zum API-Explorer, cURL oder Python Upgrade-Anweisungen.

Wenn Ihre Bereitstellung zusätzliche Appliance-Typen umfasst, müssen Sie die folgenden Abhängigkeiten berücksichtigen, bevor Sie mit den Upgrade-Anweisungen fortfahren.

Wenn Ihr Einsatz beinhaltet...	Aufgaben vor dem Upgrade	Bestellung aktualisieren
Befehlsgeräte	Reservieren Sie ein Wartungsfenster von einer Stunde für Command-Appliances, die 50.000 Geräte oder mehr verwalten.	• Befehlsgerät • Geräte entdecken • Alle Explore-Appliances (Managerknoten, dann Datenknoten) • Appliances verfolgen
Entdecken Sie Geräte	siehe ExtraHop-Plattenspeicher aktualisieren.	
Appliances verfolgen	Keine	

Aktualisieren Sie die ExtraHop-Firmware über den REST API Explorer

Laden Sie die Firmware herunter und aktualisieren Sie die Appliance

1. klicken **BEITRAG /extrahop/firmware/download/url**.
2. klicken **Probiere es aus**.
3. Geben Sie im Feld die folgenden Felder an:

   • **Firmware-URL:** Die URL, von der die Firmware-.tar-Datei heruntergeladen werden kann.

   • **aufrüsten:** Gibt an, ob die Appliance nach Abschluss des Firmware-Downloads aktualisiert werden soll. Setze dieses Feld auf `true`.

   Das Textfeld sollte dem folgenden Beispieltext ähneln:

   ```
   {
     "upgrade": true,
     "firmware_url": "https://example.extrahop.com/eda/8.7.1.tar"
   }
   ```

4. klicken **Anfrage senden**.

   Notieren Sie sich in den Antwort-Headern den Wert nach dem letzten Schrägstrich in der `location` Kopfzeile. Sie benötigen diesen Wert, um den Fortschritt des Upgrade-Jobs zu überwachen. Die Job-ID im folgenden Beispiel lautet beispielsweise `ebbdbc9e-7113-448c-ab9b-cc0ec2307702`

   ```
   /api/v1/jobs/ebbdbc9e-7113-448c-ab9b-cc0ec2307702
   ```

Überwachen Sie den Fortschritt des Upgrade-Jobs

1. klicken **Jobs**.
2. klicken **GET /jobs/ {id}**.
3. Geben Sie im Feld id den Wert ein, den Sie aus dem `location` Kopfzeile in der vorherigen Aufgabe.
4. klicken **Anfrage senden**.
5. Sehen Sie sich im Antworttext Informationen zum Job an.

   Die `status` Feld ist `DONE` wenn der Job abgeschlossen ist.

Aktualisieren Sie die ExtraHop-Firmware mit cURL

Sie können die Firmware auf einer Appliance mit dem cURL-Befehl aktualisieren.

Bevor Sie beginnen

- Das cURL-Tool muss auf Ihrem Computer installiert sein.
- Die.tar-Datei der Systemfirmware muss auf Ihren Computer heruntergeladen werden.

1. Öffnen Sie eine Terminalanwendung.
2. Laden Sie die Firmware herunter und aktualisieren Sie die Appliance.

   Führen Sie den folgenden Befehl aus, wobei `YOUR_KEY` ist der API-Schlüssel für Ihr Benutzerkonto, `HOSTNAME` ist der Hostname Ihrer ExtraHop-Appliance und `FIRMWARE_URL` ist die URL, von der die Firmware-.tar-Datei heruntergeladen werden kann:

   ```
   curl -v -X POST https://HOSTNAME/api/v1/extrahop/firmware/download/url -H
    "Authorization: ExtraHop apikey=YOUR_KEY" -H "Content-Type: application/
   json" -d "{ \"upgrade\": true, \"firmware_url\": \"FIRMWARE_URL\"}"
   ```

Notieren Sie sich in der Befehlsausgabe die Job-ID im Location-Header. Die Job-ID im folgenden Beispiel lautet beispielsweise `ebbdbc9e-7113-448c-ab9b-cc0ec2307702`:

```
< Location: /api/v1/jobs/ebbdbc9e-7113-448c-ab9b-cc0ec2307702
```

3. Überwachen Sie den Fortschritt des Upgrade-Jobs.

   Führen Sie den folgenden Befehl aus, wobei `YOUR_KEY` ist der API-Schlüssel für Ihr Benutzerkonto `HOSTNAME` ist der Hostname Ihrer Appliance und `JOB_ID` ist die ID, die Sie im vorherigen Schritt aufgezeichnet haben:

   ```
   curl -v -X GET https://HOSTNAME/api/v1/jobs/JOB_ID -H "Authorization:
    ExtraHop apikey=API_KEY"
   ```

   Der Befehl zeigt ein Objekt an, das Informationen über den Upgrade-Job enthält. Das Upgrade ist abgeschlossen, wenn `status` Feld ist `DONE`. Wenn das Upgrade nicht abgeschlossen ist, warten Sie einige Minuten und führen Sie den Befehl erneut aus.

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

Das ExtraHop GitHub-Repository enthält ein Python-Skript, das mehrere Appliances aktualisiert, indem es URLs, API-Schlüssel und Firmware-Dateipfade aus einer CSV-Datei liest.

Wichtig: Das Beispiel-Python-Skript authentifiziert sich beim Sensor oder der Konsole über einen API-Schlüssel, der nicht mit der RevealX 360-REST-API kompatibel ist. Um dieses Skript mit RevealX 360 auszuführen, müssen Sie das Skript so ändern, dass es sich mit API-Token authentifiziert. Sehen Sie die py_rx360_auth.py Skript im ExtraHop GitHub-Repository für ein Beispiel für die Authentifizierung mit API-Token.

Hinweis: Das Skript deaktiviert die Aufnahme von Datensatz für ExtraHop-Plattenspeicher nicht automatisch. Du musst Datensatz manuell deaktivieren bevor Sie das Skript für einen ExtraHop-Recordstore ausführen.

1. Gehe zum GitHub-Repository mit ExtraHop-Codebeispielen und laden Sie den Inhalt des Verzeichnisses upgrade_system auf Ihren lokalen Computer herunter.
2. Öffnen Sie in einem Texteditor `systems.csv` archivieren und ersetzen Sie die Beispielwerte durch die Hostnamen und API-Schlüssel Ihrer Appliances.
3. Führen Sie den `upgrade_system_url.py` skript.

   Die folgenden Argumente sind optional:

   --max-threads {int}

   Gibt die maximale Anzahl gleichzeitiger Threads an. Der Standardwert ist 2.

   --warte {float}

   Gibt an, wie viele Minuten gewartet werden soll, bevor der Status eines Upgrade-Jobs überprüft wird. Der Standardwert ist 0,5.

   Mit dem folgenden Befehl werden beispielsweise maximal 3 Appliances gleichzeitig aktualisiert:

   ```
   python3 upgrade_system_url.py --max-threads 3
   ```

   Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die TLS-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)
   ```

ExtraHop-Plattenspeicher aktualisieren

Aufgaben vor dem Upgrade

Bevor Sie einen ExtraHop-Recordstore aktualisieren, müssen Sie die Aufnahme von Datensätzen stoppen. Sie können die Aufnahme von Datensatz für alle Knoten in einem Cluster von einem einzelnen Knoten aus stoppen.

Hinweis: Die Botschaft `Could not determine ingest status on some nodes` und `Error` wird möglicherweise auf der Seite Cluster-Datenverwaltung in den Verwaltungseinstellungen der aktualisierten Knoten angezeigt, bis alle Knoten im Cluster aktualisiert sind. Diese Fehler werden erwartet und können ignoriert werden.

1. Öffnen Sie eine Terminal-Anwendung.
2. Führen Sie den folgenden Befehl aus, wobei `YOUR_KEY` ist die API für Ihr Benutzerkonto und `HOSTNAME` ist der Hostname Ihres ExtraHop-Recordstores:

```
curl -X PATCH "https://HOST/api/v1/extrahop/cluster" -H "accept:
 application/json" -H "Authorization: ExtraHop apikey=YOUR_KEY" -H
 "Content-Type: application/json" -d "{ \"ingest_enabled\": false}"
```

Aufgaben nach dem Upgrade

Nachdem Sie alle Knoten im Recordstore-Cluster aktualisiert haben, aktivieren Sie die Datensatzaufnahme.

1. Öffnen Sie eine Terminal-Anwendung.
2. Führen Sie den folgenden Befehl aus, wobei `YOUR_KEY` ist die API für Ihr Benutzerkonto und `HOSTNAME` ist der Hostname Ihres ExtraHop-Recordstores:

```
curl -X PATCH "https://HOST/api/v1/extrahop/cluster" -H "accept:
 application/json" -H "Authorization: ExtraHop apikey=YOUR_KEY" -H
 "Content-Type: application/json" -d "{ \"ingest_enabled\": true}"
```

Ändern Sie einen Dashboard-Besitzer über die REST-API

Dashboards gehören dem angemeldeten Benutzer, der sie erstellt hat. Wenn ein Benutzer nicht mehr in Ihrem Unternehmen ist, müssen Sie möglicherweise den Besitzer des Dashboards ändern, um das Dashboard zu verwalten.

Um die Inhaberschaft an einem Dashboard zu übertragen, benötigen Sie die Dashboard-ID und den Benutzernamen des Dashboard-Inhabers. Sie können den Benutzernamen des Besitzers eines Dashboard nur über die REST-API anzeigen.

Bevor Sie beginnen

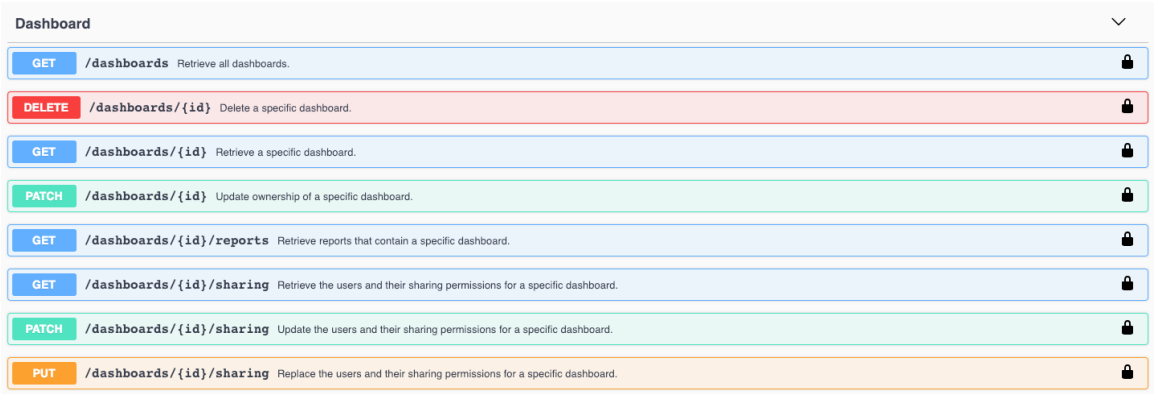
- Für Sensoren und ECA-VMs benötigen Sie einen gültigen API-Schlüssel mit System- und Zugriffsverwaltung Privilegien oder höher. (siehe Generieren Sie einen API-Schlüssel.)
- Für RevealX 360 benötigen Sie gültige REST-API-Anmeldeinformationen mit System- und Zugriffsverwaltung Privilegien oder höher. (siehe REST-API-Anmeldeinformationen erstellen.)
- Machen Sie sich mit dem vertraut ExtraHop REST-API-Leitfaden um zu erfahren, wie Sie im ExtraHop REST API Explorer navigieren.

Rufen Sie die Dashboard-IDs ab

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

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

2. Geben Sie Ihre REST-API-Anmeldeinformationen Anmeldedaten.
   - Für Sensoren und ECA-VMs klicken Sie auf **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
   - Klicken Sie für RevealX 360 auf **Geben Sie die API-Anmeldeinformationen ein** und fügen Sie dann die ID und das Geheimnis Ihrer API-Anmeldeinformationen ein oder geben Sie sie in das **ID** und **Geheim** Felder.
3. klicken **Autorisieren** und klicken Sie dann **Schliessen**.
4. klicken **Armaturenbrett** um Dashboard-Operationen anzuzeigen.



5. klicken **GET /dashboards**.
6. klicken **Probiere es aus** und klicken Sie dann **Anfrage senden** um die Anfrage an Ihren Sensor oder Ihre Konsole zu senden.
7. Suchen Sie nach den Dashboards anhand des Dashboard-Namens oder anhand des Benutzerkonto, das in der `"owner"` Feld. Wenn Ihre Liste von Dashboards lang ist, können Sie Strg-F drücken und den Antworttext durchsuchen.
   Für unser Beispiel wollen wir das ändern `"LDAP Server Health"` Dashboard erstellt vom Benutzerkonto für `"marksmith"`:

```
{
    "id": 1876,
    "comment": null,
    "mod_time": 1507576983922,
    "author": "Mark Smith",
    "name": "LDAP Server Health",
    "owner": "marksmith",
    "built-in": false,
    "short_code": "MpXgk",
    "rights": [
      "transfer",
      "view",
      "edit",
      "share",
      "delete"
    ]
}
```

8. Notieren Sie sich die Zahl in der `"id"` Feld für jedes Dashboard, das Sie ändern möchten.

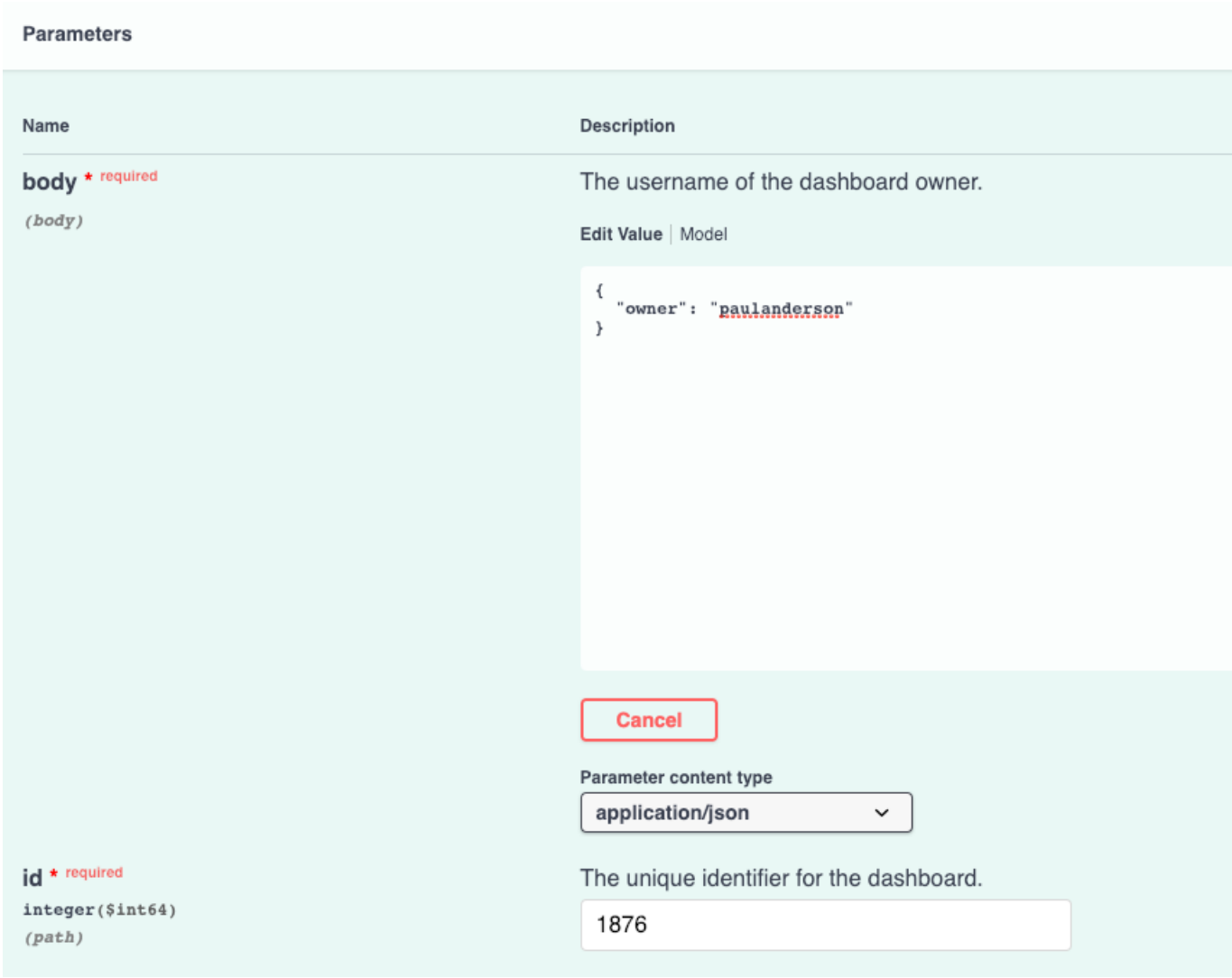
Den Besitzer des Dashboard ändern

1. Scrollen Sie auf der Seite mit den Dashboard-Vorgängen zum Abschnitt /dashboards/ {id}.
2. klicken **PATCH /dashboards/ {id}**.
3. klicken **Probiere es aus**.
   Das JSON-Schema wird automatisch zum Textfeld für den Body-Parameter hinzugefügt.

4. Im Textfeld Textkörper, in der `"owner"` Feld, ersetzen `string` mit dem Benutzernamen des neuen Besitzers.
5. In der **id** Feld, geben Sie die Nummer ein, die Sie zuvor für das Dashboard notiert haben.

   In unserem Beispiel ist dieser Wert `1876`. (Sie können jeweils nur ein Dashboard über den REST API Explorer ändern.)

   In der folgenden Abbildung haben wir das JSON hinzugefügt `"string"` für die `"owner"` Parameter zum Körper Parameter-Textfeld, geändert `"string"` zu `"paulanderson"`, und getippt `"1876"` in der id Feld.



6. klicken **Anfrage senden** um die Anfrage an Ihren Sensor oder Ihre Konsole zu senden.
   Unter Antwort des Servers, das Kode Spaltenanzeigen `204` wenn die Operation erfolgreich ist. Du kannst klicken **GET /dashboards** noch einmal, um zu überprüfen, ob `"owner"` Feld hat sich geändert. Beachten Sie, dass Sie nur den Dashboard-Besitzer ändern können. Sie können den Dashboard-Namen oder die Autorenfelder nicht über die REST-API ändern.

   Das Dashboard ist jetzt verfügbar unter **Meine Dashboards** im ExtraHop-System für den neuen Benutzer. Als neuer Besitzer können Sie sich jetzt in Ihr ExtraHop-System einloggen und andere Dashboard-Eigenschaften wie den Dashboard-Namen oder den Autor ändern.

Hinweis: Nachdem Sie geklickt haben **Anfrage senden**, der REST API Explorer stellt Skripte für die Operation in Curl, Python 2.7 oder Ruby bereit.

Python-Skriptbeispiel

Das ExtraHop GitHub-Repository enthält ein Python-Beispielskript, das nach allen Dashboards sucht, die einem Benutzerkonto auf einem Sensor oder Konsole und ändert dann den Besitzer für all diese Dashboards in ein anderes Benutzerkonto.

Wichtig: Das Beispiel-Python-Skript authentifiziert sich beim Sensor oder der Konsole über einen API-Schlüssel, der nicht mit der RevealX 360-REST-API kompatibel ist. Um dieses Skript mit RevealX 360 auszuführen, müssen Sie das Skript so ändern, dass es sich mit API-Token authentifiziert. Sehen Sie die py_rx360_auth.py Skript im ExtraHop GitHub-Repository für ein Beispiel für die Authentifizierung mit API-Token.

1. Gehe zum GitHub-Repository mit ExtraHop-Codebeispielen und laden Sie die `change_dashboard_owner/change_dashboard_owner.py` Datei auf Ihrem lokalen Computer.
2. Öffnen Sie in einem Texteditor den `change_dashboard_owner.py` archivieren und ersetzen Sie die folgenden Konfigurationsvariablen durch Informationen aus Ihrer Umgebung:
   - **GASTGEBER:** Die IP-Adresse oder der Hostname des Sensor oder der Konsole.
   - **API_KEY:** Der API-Schlüssel.
   - **AKTUELL:** Der Benutzername des aktuellen Dashboard-Besitzers.
   - **NEU:** Der Benutzername des neuen Dashboard-Besitzers.
3. Führen Sie den folgenden Befehl aus:

```
python3 change_dashboard_owner.py
```

Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die TLS-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)
```

Extrahieren Sie die Geräteliste über die REST-API

Mit der ExtraHop REST-API können Sie die Liste der Geräte extrahieren, die vom Sensor oder Konsole. Indem Sie die Liste mit einem REST-API-Skript extrahieren, können Sie die Liste in ein Format exportieren, das von Drittanbieteranwendungen gelesen werden kann, z. B. in einer Configuration Management Datenbank (CMDB). In diesem Thema zeigen wir Methoden zum Extrahieren einer Liste sowohl mit dem cURL-Befehl als auch mit einem Python-Skript.

Bevor Sie beginnen

- Für Sensoren und ECA-VMs benötigen Sie einen gültigen API-Schlüssel, um Änderungen über die REST-API vorzunehmen und die folgenden Verfahren durchzuführen. (siehe Generieren Sie einen API-Schlüssel.)
- Für RevealX 360 benötigen Sie gültige REST-API-Anmeldeinformationen, um Änderungen über die REST-API vorzunehmen und die folgenden Verfahren durchzuführen. (siehe REST-API-Anmeldeinformationen erstellen.)

Rufen Sie die Geräteliste mit dem Befehl cURL ab

Die Geräteliste enthält alle Geräte-Metadaten wie MAC-Adressen und Geräte-IDs. Sie können die Geräteliste jedoch mit einem JSON-Parser filtern, um die spezifischen Informationen zu extrahieren, die Sie exportieren möchten. In diesem Beispiel wird die Geräteliste abgerufen und dann mit dem jq-Parser gefiltert, um nur den Anzeigenamen jedes Gerät zu extrahieren.

Hinweis: Das folgende Verfahren ist nicht mit der RevealX 360-REST-API kompatibel. Informationen zum Abrufen der Geräteliste von RevealX 360 finden Sie unter Rufen Sie die Geräteliste von RevealX 360 mit dem Befehl cURL ab.

Bevor Sie beginnen

- Das cURL-Tool muss auf Ihrem Computer installiert sein.
- Der JQ-Parser muss auf Ihrem Computer installiert sein. Weitere Informationen finden Sie unter https://stedolan.github.io/jq/.

Öffnen Sie eine Terminal-Anwendung und führen Sie den folgenden Befehl aus, wobei `YOUR_KEY` ist die API für Ihr Benutzerkonto, `HOSTNAME` ist der Hostname Ihres Sensor oder Ihrer Konsole und `MAX_DEVICES` ist eine Zahl, die groß genug ist, um mehr als die Gesamtzahl der von Ihrem System erkannten Geräte zu sein:

```
curl -s -X POST "https://HOSTNAME/api/v1/devices/search" --header
 "accept: application/json" --header "Authorization: ExtraHop
 apikey=YOUR_KEY" --header "Content-Type: application/json" -d
 "{ \"active_from\": 1, \"active_until\": 0, \"limit\": MAX_DEVICES}" |
 jq -r '.[] | .display_name'
```

Hinweis: Wenn der Befehl keine Ergebnisse zurückgibt, stellen Sie sicher, dass Ihrem ExtraHop-System wurde ein vertrauenswürdiges Zertifikat hinzugefügt. Alternativ können Sie das hinzufügen `--insecure` Option zum Abrufen der Geräteliste von einem ExtraHop-System ohne vertrauenswürdiges Zertifikat; diese Methode ist jedoch nicht sicher und wird nicht empfohlen.

Hinweis: Sie können das anhängen `select(.analysis == "LEVEL")` Option zum Filtern der Ergebnisse nach Analyseebene. Beispielsweise schränkt der folgende Befehl die Ergebnisse so ein, dass nur Geräte berücksichtigt werden, die für die erweiterte Analyse ausgewählt wurden:

```
curl -s -X POST "https://HOSTNAME/api/v1/devices/search" --
header "accept: application/json" --header "Authorization:
 ExtraHop apikey=YOUR_KEY" --header "Content-Type: application/
json" -d "{ \"active_from\": 1, \"active_until\": 0, \"limit\":
 1000000000}" | jq -r '.[] | select(.analysis == "advanced")
 | .display_name'
```

Hinweis: Sie können das anhängen `select(.critical == BOOLEAN)` Option zum Filtern der Ergebnisse nach dem kritischen Feld. Mit dem folgenden Befehl werden die Ergebnisse beispielsweise so begrenzt, dass sie nur Geräte enthalten, die vom ExtraHop-System als kritisch eingestuft wurden:

```
curl -s -X POST "https://HOSTNAME/api/v1/devices/search" --
header "accept: application/json" --header "Authorization:
 ExtraHop apikey=YOUR_KEY" --header "Content-Type: application/
json" -d "{ \"active_from\": 1, \"active_until\": 0, \"limit
\": 1000000000}" | jq -r '.[] | select(.critical == true)
 | .display_name'
```

Hinweis: Sie können das anhängen `select(.cloud_instance_name != null)` Option zum Filtern der Ergebnisse nach dem Feld mit dem Namen der Cloud-Instanz. Beispielsweise schränkt der folgende Befehl die Ergebnisse so ein, dass sie nur Geräte mit einem Cloud-Instanznamen enthalten:

```
curl -s -X POST "https://HOSTNAME/api/v1/devices/search" --
header "accept: application/json" --header "Authorization:
 ExtraHop apikey=YOUR_KEY" --header "Content-Type: application/
json" -d "{ \"active_from\": 1, \"active_until\": 0, \"limit
```

```
\": 1000000000}" | jq -r '.[] | select(.cloud_instance_name !=
 null) | .cloud_instance_name'
```

Rufen Sie die Geräteliste von RevealX 360 mit dem Befehl cURL ab

Die Geräteliste enthält alle Geräte-Metadaten wie MAC-Adressen und Geräte-IDs. Sie können die Geräteliste jedoch mit einem JSON-Parser filtern, um die spezifischen Informationen zu extrahieren, die Sie exportieren möchten. In diesem Beispiel wird die Geräteliste abgerufen und dann mit dem jq-Parser gefiltert, um nur den Anzeigenamen jedes Gerät zu extrahieren.

Hinweis: Das folgende Verfahren ist nur mit der RevealX 360 REST-API kompatibel. Informationen zum Abrufen der Geräteliste von Sensoren und ECA-VMs finden Sie unter Rufen Sie die Geräteliste mit dem Befehl cURL ab.

Bevor Sie beginnen

- Das cURL-Tool muss auf Ihrem Computer installiert sein.
- Der JQ-Parser muss auf Ihrem Computer installiert sein. Weitere Informationen finden Sie unter https://stedolan.github.io/jq/.

1. Öffnen Sie eine Terminal-Anwendung und führen Sie den folgenden Befehl aus, wobei `REVEAL_X_360_REST_API` ist der Hostname der RevealX 360-API. Dieser Hostname wird angezeigt in RevealX 360 auf der API Access Seite unter API-Endpunkt. Der Hostname beinhaltet nicht `/oauth2/token`:

   ```
   HOST="https://REVEAL_X_360_REST_API"
   ```

2. Führen Sie den folgenden Befehl aus, wobei `YOUR_ID` ist die ID der REST-API-Anmeldeinformationen:

   ```
   ID="YOUR_ID"
   ```

3. Führen Sie den folgenden Befehl aus, wobei `YOUR_SECRET` ist das Geheimnis der REST-API-Anmeldeinformationen:

   ```
   SECRET="YOUR_SECRET"
   ```

4. Führen Sie den folgenden Befehl aus:

   ```
   AUTH=$(printf "$ID:$SECRET" | base64 --wrap=0)
   ```

5. Führen Sie den folgenden Befehl aus:

   ```
   ACCESS_TOKEN=$(curl -s \
       -H "Authorization: Basic ${AUTH}" \
       -H "Content-Type: application/x-www-form-urlencoded" \
       --request POST \
        ${HOST}/oauth2/token \
       -d "grant_type=client_credentials" \
       | jq -r '.access_token')
   ```

6. Führen Sie den folgenden Befehl aus, wobei `MAX_DEVICES` ist eine Zahl, die groß genug ist, um mehr als die Gesamtzahl der von Ihrem System erkannten Geräte zu sein:

   ```
   curl -s -X GET -H "Authorization: Bearer ${ACCESS_TOKEN}"  "$HOST/api/
   v1/devices?active_from=1&active_until=0&limit=MAX_DEVICES" | jq -r '.[]
    | .display_name'
   ```

   Hinweis: Sie können das anhängen `select(.analysis == "LEVEL")` Option zum Filtern der Ergebnisse nach Analyseebene. Beispielsweise schränkt der folgende Befehl die

Ergebnisse so ein, dass nur Geräte berücksichtigt werden, die für die erweiterte Analyse ausgewählt wurden:

```
curl -s -X GET -H "Authorization: Bearer
 ${ACCESS_TOKEN}"  "$HOST/api/v1/devices?
active_from=1&active_until=0&limit=10000000000" | jq -r '.[] |
 select(.analysis == "advanced") | .display_name'
```

Hinweis: Sie können das anhängen `select(.critical == BOOLEAN)` Option zum Filtern der Ergebnisse nach dem kritischen Feld. Mit dem folgenden Befehl werden die Ergebnisse beispielsweise so begrenzt , dass sie nur Geräte enthalten, die vom ExtraHop-System als kritisch eingestuft wurden:

```
curl -s -X GET -H "Authorization: Bearer
 ${ACCESS_TOKEN}"  "$HOST/api/v1/devices?
active_from=1&active_until=0&limit=10000000000"  | jq -r '.[] |
 select(.critical == true) | .display_name'
```

Hinweis: Sie können das anhängen `select(.cloud_instance_name != null)` Option zum Filtern der Ergebnisse nach dem Feld mit dem Namen der Cloud-Instanz. Beispielsweise schränkt der folgende Befehl die Ergebnisse so ein, dass sie nur Geräte mit einem Cloud-Instanznamen enthalten:

```
curl -s -X GET -H "Authorization: Bearer
 ${ACCESS_TOKEN}"  "$HOST/api/v1/devices?
active_from=1&active_until=0&limit=10000000000"  | jq -r '.[] |
 select(.cloud_instance_name != null) | .cloud_instance_name'
```

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

Das ExtraHop GitHub-Repository enthält ein Python-Skript, das die Geräteliste einschließlich aller Geräte-Metadaten extrahiert und die Liste in eine CSV-Datei im selben Verzeichnis wie das Skript schreibt.

1. Gehe zum ExtraHop Codebeispiele GitHub-Repository und laden Sie das herunter `extract_device_list/extract_device_list.py` Datei auf Ihrem lokalen Computer.
2. Öffnen Sie in einem Texteditor den `extract_device_list.py` archivieren und ersetzen Sie die Konfigurationsvariablen durch Informationen aus Ihrer Umgebung.
   - Geben Sie für Sensoren und ECA-VMs die folgenden Konfigurationsvariablen an:
     - **GASTGEBER:** Die IP-Adresse oder der Hostname des Sensor oder der ECA-VM.
     - **API-SCHLÜSSEL:** Der API-Schlüssel.
     - **CSV_DATEI:** Die Datei, die die Liste der Gerätegruppen enthält.
     - **DATEINAME:** Die Datei, in die die Ausgabe geschrieben wird
     - **GRENZE:** Die maximale Anzahl von Geräten, die mit jeder GET-Anfrage abgerufen werden sollen
     - **SAVEL 2:** Ruft übergeordnete L2-Geräte ab. Diese Variable ist nur gültig, wenn Sie das ExtraHop-System aktiviert haben, Geräte anhand der IP-Adresse zu erkennen.
     - **NUR FÜR FORTGESCHRITTENE:** Ruft nur Geräte ab, die derzeit einer erweiterten Analyse unterzogen werden
     - **NUR HOHER WERT:** Ruft nur Geräte ab, die als hoher Wert eingestuft werden
   - Geben Sie für RevealX 360 die folgenden Konfigurationsvariablen an:
     - **GASTGEBER:** Der Hostname der RevealX 360-API. Dieser Hostname wird auf der RevealX 360 API Access-Seite unter API-Endpunkt angezeigt. Der Hostname enthält nicht die `/oauth2/token`.
     - **ID:** Die ID der RevealX 360 REST-API-Anmeldeinformationen.

- **GEHEIM:** Das Geheimnis der RevealX 360 REST-API-Anmeldeinformationen.
- **CSV_DATEI:** Die Datei, die die Liste der Gerätegruppen enthält.
- **DATEINAME:** Die Datei, in die die Ausgabe geschrieben wird
- **GRENZE:** Die maximale Anzahl von Geräten, die mit jeder GET-Anfrage abgerufen werden sollen
- **SAVEL 2:** Ruft übergeordnete L2-Geräte ab. Diese Variable ist nur gültig, wenn Sie das ExtraHop-System aktiviert haben, Geräte anhand der IP-Adresse zu erkennen.
- **NUR FÜR FORTGESCHRITTENE:** Ruft nur Geräte ab, die derzeit einer erweiterten Analyse unterzogen werden
- **NUR HOHER WERT:** Ruft nur Geräte ab, die als hoher Wert eingestuft werden

3. Führen Sie den folgenden Befehl aus:

```
python3 extract_device_list.py
```

Hinweis: Wenn das Skript eine Fehlermeldung zurückgibt, dass die TLS-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)
```

Erstellen Sie ein vertrauenswürdiges TLS-Zertifikat über die REST-API

In der Standardeinstellung Sensoren und Konsolen fügen Sie ein selbstsigniertes TLS-Zertifikat hinzu. Sie können jedoch die Sicherheit und Leistung Ihres Systems verbessern, indem Sie ein vertrauenswürdiges Zertifikat hinzufügen, das von einer Zertifizierungsstelle (CA) signiert wurde. Sie können die Anfrage zur Zertifikatsignierung erstellen, um sie über die ExtraHop REST-API an Ihre CA zu senden. Nachdem Sie das signierte Zertifikat erhalten haben, können Sie es auch Ihrem hinzufügen Sensor oder Konsole über die REST-API.

Bevor Sie beginnen

- Du musst dich einloggen im Sensor oder Konsole mit einem Konto, das System- und Zugriffsadministrationsrechte um einen API-Schlüssel zu generieren.
- Sie benötigen einen gültigen API-Schlüssel, um Änderungen über die REST-API vorzunehmen und die folgenden Verfahren durchzuführen. (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.

Hinweis: Sie können die Verfahren in diesem Thema auch über die Administrationseinstellungen ausführen. Weitere Informationen finden Sie in den folgenden Themen:

- Erstellen Sie eine Anfrage zur Zertifikatsignierung von Ihrem ExtraHop-System
- TLS-Zertifikat

Erstellen Sie eine Anfrage zum Signieren eines TLS-Zertifikats

Um ein signiertes TLS-Zertifikat zu erstellen, müssen Sie eine Anfrage zur Zertifikatsignierung an eine vertrauenswürdige CA senden.

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

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

2. Klicken Sie **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
3. Klicken Sie **Autorisieren** und klicken Sie dann **Schliessen**.
4. Klicken Sie **ExtraHop** und klicken Sie dann **Anfrage für Post/Extrahop/SSLCert/Signierung**.
5. Klicken Sie **Probiere es aus**.
   Das JSON-Schema wird automatisch dem hinzugefügt Parameter für die Signierung von SSL-Zertifikaten Parameter-Textfeld.
6. In der Parameter für die Signierung von SSL-Zertifikaten Parameter-Textfeld, geben Sie die Felder für die Zertifikatsignierungsanforderung an.
   a) In der `common_name` Feld, ersetzen `string` mit dem vollqualifizierten Domänenname Ihres Sensor oder Ihrer Konsole.
   b) In der `subject_alternative_names` Feld, fügen Sie einen oder mehrere alternative Domainnamen oder IP-Adressen für Ihren Sensor oder Ihre Konsole hinzu.

      Hinweis: Das `subject_alternative_names` Feld ist erforderlich. Wenn Ihr System nur einen Domänenname hat, duplizieren Sie den Wert aus dem `common_name` Feld. Sie müssen mindestens einen alternativen Betreff mit dem folgenden Typ angeben `dns`, aber für zusätzliche alternative Namen kann der Typ auf gesetzt werden `ip` oder `dns`.

   c) Optional: In der `email_address` Feld, ersetzen `string` mit der E-Mail-Adresse des Zertifikatsinhabers.
   d) Optional: In der `organization_name` Feld, ersetzen `string` mit dem eingetragenen Firmennamen Ihrer Organisation.
   e) Optional: In der `country_code` Feld, ersetzen `string` mit dem 2-stelligen ISO-Ländercode des Landes, in dem sich Ihre Organisation befindet.
   f) Optional: In der `state_or_province_name` Feld, ersetzen `string` mit dem Namen des Bundesstaates oder in dem sich Ihre Organisation befindet.
   g) Optional: In der `locality_name` Feld, ersetzen `string` mit dem Namen der Stadt, in der sich Ihre Organisation befindet.
   h) Optional: In der `organizational_unit_name` Feld, ersetzen `string` mit dem Namen Ihrer Abteilung innerhalb Ihrer Organisation.

   Das Wert Der Abschnitt sollte dem folgenden Beispiel ähneln:

```
{
  "subject": {
    "common_name": "example.com",
    "email_address": "admin@example.com",
    "organization_name": "Example",
    "country_code": "US"
  },
  "subject_alternative_names": [
    {
      "name": "www.example.com",
      "type": "dns"
    }
  ]
}
```

7. Klicken Sie **Anfrage senden** um die Signieranforderung zu erstellen.
   In der Antwort des Servers Abschnitt, der Antworttext zeigt die Signieranforderung in der `pem` Feld.

Nächste Schritte
Senden Sie die Signaturanfrage an Ihre CA, um Ihr signiertes TLS-Zertifikat zu erstellen.

Wichtig: Die Signieranforderung enthält Escape-Sequenzen, die Zeilenumbrüche (\n) darstellen. Ersetzen Sie jede Instanz von\ndurch einen Zeilenumbruch, bevor Sie die Anfrage an Ihre CA senden. Sie können die PEM-Anfrage manuell in einem Texteditor oder automatisch über ein JSON-Analyseprogramm ändern, wie im folgenden Beispielbefehl gezeigt:

```
echo '<json_output>' | python -c 'import sys, json; print
 json.load(sys.stdin)["pem"]'
```

Ersetzen Sie die `<json_output>` Variable mit der gesamten JSON-Zeichenfolge, die im Abschnitt Response Body zurückgegeben wird.

Fügen Sie Ihrem Sensor oder Ihrer Konsole ein vertrauenswürdiges TLS-Zertifikat hinzu

Sie können Ihrem ein TLS-Zertifikat hinzufügen, das von einer vertrauenswürdigen CA signiert wurde Sensor oder Konsole über den REST API Explorer.

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

   Die URL ist der Hostname oder die IP-Adresse Ihres Sensor oder Konsole, gefolgt von `/api/v1/explore/`. Wenn Ihr Hostname beispielsweise seattle-eda ist, lautet die URL `https://seattle-eda/api/v1/explore/`.
2. klicken **API-Schlüssel eingeben** und fügen Sie dann Ihren API-Schlüssel ein oder geben Sie ihn in das **API-Schlüssel** Feld.
3. klicken **Autorisieren** und klicken Sie dann **Schliessen**.
4. klicken **ExtraHop** und klicken Sie dann **PUT/ExtraHop/SSLCERT**.
5. klicken **Probiere es aus**.
6. In der **Zertifikat und Schlüssel** Feld, fügen Sie das TLS-Zertifikat ein.

   Das Zertifikat sollte dem folgenden Text ähneln:

```
-----BEGIN CERTIFICATE-----
a0O8zvV4MlDhWX4e0VyvGAJx+9d4AqQB4Czy/P7z36CmHe2Y7PPdVSeWHNCQoJ0g
CnO42u2V9YKNFYRQejIJv8CxGVJKsdfV0iP0WnCvpZXkaBOYIrDvE5xn010WPUls
6qe3mCXsUK87i++mYuVDA1U0A5YVXRO2OOWIWy7P+MCU/cR/op3Jpekng2cxN4qD
FqGbtRpLdCuJ/xGWL1FFRHBg76+TbO+pxgZhiCtHYXfMKIaoPmDwsAqEtLbizz1W
mbMig9hs4QNcJ+aMNSnTZpkbeBR4a2nkGnQoYvnFOXV/nWzvfHmI4ydSH9g4I8qt
4ArqFepInvm70n07FYAKL6Mdd1i+7ieo9AqckltVzzKFzkakHm04214wtsYmle94
4HqIJ7p7NH5maXxttXMzHFlArbnjHWCl0gIv8lAu+IvLJ8aiGAb3zqveNz6ZAZ5j
PGAUsP+dVYV/8VjvqhkiP/1jWzUHwzpdlHbcD8qOkAF41fnbv+2EXqFJ096JSSiU
rqeJpgNuH3LbkT0KORAiLoGLMZKEKxF+3OpLVD7ox7NQh9pMdZlB8tcTbTmsvD8T
3L2tMVZssqYOANcidtd17t72VW4hzQURT1me5tGWxpN6od/q6B+FIvRq/7Vq0UE1
c2AG/om5UN/Vj3pUjXzq/B1IWUS9TicRcKdl5wrKEkPUGjK4w1R/87bj5HSn8nyd
lMCcOpLTokHj0B5+8O1ylNhVXNPlj3eY0n6OQOdClBqTDM0/4sB3XgeC/pjpleU3
3uot+wM/GoN/Dqb1LPt3BNpUQuCzSfmGSSOXiWELsEhz3ix/36a9eUWjfhmtPsW5
dne5Lf+G7cf+ebsRTb7R89GmgKzTpUl1KAzKINAebkT6WrWWljugpA0BcfANjS6o
mik4ZbY8d54UtA17evprr2+8UotIgVIrCbfLgA2DY8QOTCBYIFKJ3GZAedqRK9Sm
I2qdaB6QBczYNaVYSeCsBdHHw1+h7dBeqdUUwYKtmPW96/djj/6vJSXh9/UX/3c0
eqXG36w/lqJAYu8QtAydJsVC85IzqzikkX0f0KE315Doginpg59yix9dHD2sxLb1
X39BRpLkZ9nvW6ke2YHU/VKBVIxqSslukGoTUIcUtPJrtMQOwCi/EQQXbPK9a2pW
K51938h6OuLjNbDTFuxfhE4zITWHTgyAs2MNVR9+uDUiVJclX+CIPjhZzjyPqmD6
6uh8Sr3zndOMabqDquo69rMQyvclF0xOUMVgUw1Rb8Y=
-----END CERTIFICATE-----
```

Hinweis: Wenn Sie möchten, dass das Zertifikat mit Ihrem eigenen privaten Schlüssel signiert wird, können Sie Ihren Schlüssel nach dem TLS-Zertifikat einfügen, getrennt durch einen Zeilenumbruch. Wir empfehlen jedoch, keinen eigenen Schlüssel anzugeben. Standardmäßig signiert der Sensor oder die Konsole das Zertifikat mit dem privaten Schlüssel auf dem System.

7. klicken **Anfrage senden** um das Zertifikat hinzuzufügen.

Erstellen Sie benutzerdefinierte Geräte über die REST-API

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 TLS-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)
```

Erstellen und Zuweisen eines Geräte-Tags über die REST-API

Das folgende Python-Skript erstellt ein Geräte-Tag und weist dieses Tag dann allen Geräten in einem angegebenen Subnetz zu.

```
#!/usr/bin/env python

import httplib
import urllib
import json
import sys

# Configuration Options:
host = "{HOST}"
apikey = "{API KEY}"
tag_name = "MyTestTag"
subnet = "10.20.0.[0-9]+"
batch_limit = 100
headers = {'Accept': 'application/json',
           'Authorization': "ExtraHop apikey=%s" % apikey}
conn = httplib.HTTPSConnection(host)
  def execute_req(method, path, expected_code, failure_message, body=None):
"""
   Returns the body of a successful request,
   otherwise prints error and terminates
"""

conn.request(method, "/api/v1" + path, headers=headers, body=body)
resp = conn.getresponse()
```

```
if resp.status is not expected_code:
    print(failure_message)
    print(resp.read())
    sys.exit(1)
return resp

def execute_get(path, expected_code, failure_message):
    resp = execute_req("GET", path, expected_code, failure_message)
    return json.loads(resp.read())

def execute_create(path, body, expected_code, failure_message):
    """Returns ID of newly created resource"""
    resp = execute_req("POST", path, expected_code, failure_message, body)
    resp.read() # drain the response
    return int(resp.getheader("location").split("/")[-1])

# First, search for the specified tag, by name
resp = execute_get("/tags", 200, "Unable to retrieve tags from ExtraHop")
tags = [tag for tag in resp if tag["name"] == tag_name]

if not tags:
    # tag is not found, create it
    body = json.dumps({"name": tag_name})
    tag_id = execute_create('/tags', body, 201, "Unable to create tag")
else:
    tag_id = tags[0]["id"]

query_params = {'limit': batch_limit,
                'search_type': 'ip address',
                'value': subnet}
query_string = urllib.urlencode(query_params)

# Paginate device results, building up a list of all devices to assign
device_ids = []
offset = 0

while True:
    path = "/devices?" + query_string + ("&offset=%d" % offset)
    resp = execute_get(path, 200, "Unable to retrieve devices")
    if not resp:
        break

    device_ids += [device["id"] for device in resp]
    offset += batch_limit

# Perform the assignments
resp = execute_req("POST", "/tags/%d/devices" % tag_id,
                   204, "Unable to perform assignments",
                   body=json.dumps({"assign": device_ids}))
resp.read() # drain the response

# Check that assignments were successful
resp = execute_get("/tags/%d/devices" % tag_id,
                   200, "Unable to retrieve tag assignments")
assigned_device_ids = [device["id"] for device in resp]

successful = set(device_ids).issubset(set(assigned_device_ids))
if successful:
    print("%d devices assigned to tag" % len(device_ids))
else:
    print("Unable to assign all devices to tag")
```

Abfragen von Metriken zu einem bestimmten Gerät über die REST-API

Das folgende Python-Skript fragt nach Metriken von einem HTTP Client Gerät mit der ID 9363 und druckt die Antwort aus.

```
import httplib

headers = {'Content-Type': 'application/json',
       'Accept': 'application/json',
       'Authorization': 'ExtraHop apikey={API KEY}'
body = r"""{
  "cycle": "auto",
  "from": -1800000,
  "until": 0,
  "metric_category": "http_client",
  "metric_specs": [
    {
       "name": "req"
    }
   ],
   "object_ids": [
      9363
   ],
   "object_type": "device"
}"""
conn = httplib.HTTPSConnection('{HOST}')
conn.request('POST', '/api/v1/metrics', headers=headers, body=body)
resp = conn.getresponse()
print resp.status, resp.reason
print resp.read()
```

Die folgende Antwort zeigt Einträge für das Gerät mit der ID 9363:

```
{
 "date": "Thu, 19 Nov 2015 23:20:07 GMT",
 "via": "1.1 localhost",
 "server": "Apache",
 "vary": "Accept-Encoding",
 "content-type": "application/json; charset=utf-8",
 "cache-control": "private, max-age=0",
 "connection": "Keep-Alive",
 "content-encoding": "gzip",
 "keep-alive": "timeout=45, max=44",
 "content-length": "277"
}

{
 "stats": [
   {
    "oid": 9363,
    "time": 1447973460000,
    "duration": 30000,
    "values": [
     2
    ]
   },
   {
    "oid": 9363,
    "time": 1447973490000,
    "duration": 30000,
    "values": [
     0
```

```
    ]
   },
   {
    "oid": 9363,
    "time": 1447973520000,
    "duration": 30000,
    "values": [
     1
    ]
   },
   {
    "oid": 9363,
    "time": 1447973550000,
    "duration": 30000,
    "values": [
     2
    ]
   }
```

Ein Objekt über die REST-API erstellen, abrufen und löschen

Dieses Beispiel zeigt, wie Sie Informationen zu einem Geräte-Tag erstellen und erfolgreich abrufen können. Nach dem Löschen der Geräte-Tags zeigt das Beispiel, dass ein Versuch, Informationen abzurufen, anschließend fehlschlägt.

Das folgende Beispiel zeigt, wie ein Geräte-Tag namens my_test_tag erstellt wird.

```
curl -i -X POST --header "Content-Type: application/json" \
--header "Accept: application/json" \
--header "Authorization: ExtraHop apikey={API KEY}" \
-d "{
\"name\": \"my_test_tag\"
}" "https://{HOST}/api/v1/tags"
```

Ein 201-Status kehrt bei Erfolg mit den folgenden Antwort-Headern zurück, die anzeigen, dass das Tag erstellt wurde, und geben den Standort und die ID des Device-Tags von /api/v1/tags/1 an.

```
{
 "date": "Wed, 18 Nov 2015 20:24:13 GMT",
 "via": "1.1 localhost",
 "server": "Apache",
 "content-type": "text/plain; charset=utf-8",
 "location": "/api/v1/tags/1",
 "cache-control": "private, max-age=0",
 "connection": "Keep-Alive",
 "keep-alive": "timeout=45, max=88",
 "content-length": "0"
}
```

Als Nächstes wird die ID (1) zur folgenden GET-Anfrage hinzugefügt, die bei Erfolg den Status 200 und die JSON-Darstellung des abgerufenen Tags zurückgibt:

```
curl -i -X GET --header "Accept: application/json" \
--header "Authorization: ExtraHop apikey={API KEY}" \
"https://{HOST}/api/v1/tags/1"
{
 "mod_time": 1447878253953,
 "id": 1,
 "name": "my_test_tag"
}
```

Als Nächstes zeigt das folgende Beispiel eine DELETE-Anfrage zum Entfernen des Geräte-Tags aus dem System, die bei Erfolg den Status 204 zurückgibt:

```
curl -i -X DELETE --header "Accept: application/json" \
--header "Authorization: ExtraHop apikey={API KEY}" \
"https://{HOST}/api/v1/tags/1"
```

Wenn schließlich eine weitere GET-Anfrage für dieses gelöschte Geräte-Tag gesendet wird, schlägt der Vorgang fehl und bei einem Fehler wird ein 404-Status zurückgegeben, der darauf hinweist, dass das Tag nicht mehr verfügbar ist.

```
curl -i -X GET --header "Accept: application/json" \
--header "Authorization: ExtraHop apikey={API KEY}" \
"https://{HOST}/api/v1/tags/1"
```

Das Datensatzprotokoll abfragen

Der folgende Anforderungstext fragt das Datensatzprotokoll ab, um 100 abzurufen HTTP zeichnet auf, wo die Methode GET ist und der Statuscode 404 ist.

```
{
  "filter": {
    "operator": "and",
    "rules": [
        {
            "field": "method",
            "operand": "GET",
            "operator": "="
        },
        {
            "field": "statusCode",
            "operand": "404",
            "operator": "="
        }
   ]
 },
 "from": -900000,
 "limit": 100,
 "types": [
   "~http"
 ]
}
```