9.6
Leitfaden zur IDS-Sensor-REST-API

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

Veröffentlicht: 2024-04-10

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 Zugriffs-Einstellungen Abschnitt, klicken **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 Abschnitt API-Schlüssel 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. In der **Auf Einstellungen zugreifen** Abschnitt, klicken **API-Zugriff**.
2. 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.
3. Klicken Sie **Einstellungen speichern** und klicken Sie dann **Erledigt**.

Richten Sie ein SSL-Zertifikat ein

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

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

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

SSL über Windows PowerShell einrichten

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

SSL 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

Objekte auf dem ExtraHop-System können durch jeden eindeutigen Wert identifiziert werden, z. B. durch die IP-Adresse , die MAC-Adresse, den Namen oder die System-ID. Um API-Operationen für ein bestimmtes Objekt auszuführen, müssen Sie jedoch die Objekt-ID suchen. Sie können die Objekt-ID mit den folgenden Methoden im REST API Explorer leicht finden.

- Die Objekt-ID wird in den Headern bereitgestellt, die von einer POST-Anforderung 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
},
```

- Die Objekt-ID ist in der URL für die meisten Objekte angegeben. Klicken Sie beispielsweise im ExtraHop-System auf **Vermögenswerte**, und dann **Geräte**. Wählen Sie ein beliebiges Gerät aus und sehen Sie sich die URL an. Im folgenden Beispiel zeigt die URL für die Geräteseite Oid=10180.

```
https://10.10.10.205/extrahop/#/Devices?details=true&device
Oid=10180&from=6&interval_type=HR&until=0&view=l2stats
```

  Um spezifische Anfragen für dieses Gerät auszuführen, fügen Sie 10180 zur id Feld im REST API Explorer oder für den Body-Parameter in Ihrer Anfrage.

Die URL für Dashboards zeigt einen Short_Code an, der hinter /Dashboard erscheint. Wenn Sie den short_code zum REST API Explorer oder zu Ihrer Anfrage hinzufügen, müssen Sie dem Shortcode eine Tilde voranstellen.

Im folgenden Beispiel ist kmc9Y der short_code. Um Anfragen für dieses Dashboard auszuführen, fügen Sie `~kmC9Y` als Wert für das Feld short_code.

```
https://10.10.10.205/extrahop/#/Dashboard/kmC9Y/?from=6&interval_
type=HR&until=0
```

Sie finden den short_code und die Dashboard-ID auch in den Dashboard-Eigenschaften für jedes Dashboard, auf das Sie über das Befehlsmenü zugreifen können ⋮. Für einige API-Operationen, wie DELETE, ist die Dashboard-ID erforderlich.

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.

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.
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"
}
```

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"
}
```

Wolke

Mit dieser Ressource können Sie Ihre lokalen Geräte verbinden Sensoren to Reveal (x) 360 Weitere Informationen finden Sie unter Stellen Sie über selbstverwaltete Sensoren eine Verbindung zu Reveal (x) 360 her.

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

Betrieb	Beschreibung
POST /cloud/connect	Verbinden Sie das ExtraHop-System mit Reveal (x) 360.

Einzelheiten der Operation

`POST /cloud/connect`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Das Token, das Sie mit Reveal (x) 360 generiert haben.

`cloud_token:` ***Schnur***

Das Token, das Sie mit Reveal (x) 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"
}
```

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.
GET /Erkennungen/Formate	Ruft alle Erkennungstypen ab.
GET /detections/formats/ {id}	Ruft einen bestimmten benutzerdefinierten 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.

Bedienung	Beschreibung
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:` ***Zeichenfolge***

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 mit den angegebenen Tickets verknüpft sind. Geben Sie „.none“ an, um nach Entdeckungen zu suchen, die nicht mit Tickets verknüpft sind.

`status:` ***Reihe von Zeichenketten***

Gibt Erkennungen für Tickets mit dem angegebenen Status zurück. Geben Sie „.none“ an, um nach Funden ohne Ticketstatus zu suchen.

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 Erkennungen zurück, die für die Triage empfohlen werden. 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.

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.

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"
        }
    ]
}
```

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.

Betrieb	Beschreibung
/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.

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/cloudressourcen	Aktualisieren Sie die Ressourcen auf dem ExtraHop-System manuell. Diese Ressourcen werden automatisch aktualisiert, wenn das System mit ExtraHop Cloud Services verbunden wird.
GET /extrahop/cluster	Rufen Sie die Cluster-Konfigurationseinstellungen von Explore ab.
PATCH /extrahop/cluster	Aktualisieren Sie die Cluster-Konfigurationseinstellungen von Explore.
GET /extrahop/detections/access	Rufen Sie die Einstellungen für die Zugriffskontrolle für Erkennungen ab.
PUT /extrahop/Erkennungen/Zugriff	Aktualisieren Sie die Einstellungen für die Zugriffskontrolle bei Erkennungen.
GET /extrahop/edition	Rufen Sie die Ausgabe 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.

Bedienung	Beschreibung
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	Ruft Informationen über eine Firmware-Version ab, auf die Sie das ExtraHop-System zurücksetzen können.
POST /extrahop/firmware/vorherig/rollback	Setzen Sie das ExtraHop-System auf die vorherige Firmware-Version zurück.
HOLEN SIE SICH /extrahop/flowlogs/secret	Rufen Sie das Flow-Log-Geheimnis ab.
POST /extrahop/flowlogs/secret	Generieren Sie ein neues Flow-Log-Geheimnis.
HOLEN SIE SICH /extrahop/idrac	Rufen Sie die iDRAC-IP-Adresse des ExtraHop-Systems ab.
GET /extrahop/platform	Ruft den Plattformnamen des ExtraHop-Systems ab. **Hinweis:** Für diesen Vorgang ist kein API-Schlüssel erforderlich.
GET /extrahop/prozesse	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	Rufen Sie 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 SSL-Zertifikat auf dem ExtraHop-System neu. Weitere Informationen finden Sie unter Erstellen Sie ein vertrauenswürdiges SSL-Zertifikat über die REST-API
AUS /extrahop/sslcert	Ersetzen Sie das SSL-Zertifikat auf dem ExtraHop-System.
POST /extrahop/sslcert/signingrequest	Erstellen Sie eine Anfrage zur Signierung eines SSL-Zertifikats. Weitere Informationen finden Sie unter Erstellen Sie ein vertrauenswürdiges SSL-Zertifikat über die REST-API.
HOLEN SIE SICH /extrahop/ticketing	Rufen Sie den Integrationsstatus des Ticketings ab.
PATCH /Extrahop/Ticketverkauf	Aktivieren oder deaktivieren Sie die Ticketing-Integration.

Bedienung	Beschreibung
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 Ermittlungen von einem externen Ticketsystem aus verfolgt werden.

`internal_ticketing_enabled:` ***Boolesch***

(Optional) Gibt an, ob Untersuchungen vom ExtraHop-System aus verfolgt werden.

`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 Erkennungsuntersuchungen 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"
}
```

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.

Bedienung	Beschreibung
GET /metrics/next/ {xid}	Wenn Sie Aktivitätsgruppenmetriken von einem anfordern Konsolemit dem `POST /metrics`, `POST /metrics/total`, oder `POST /metrics/totalbyobject` Operation und die Antwort enthält die `xid` Feld, das `GET /metrics/next/{xid}` Die Operation gibt Metriken von einem der an die Konsole angeschlossenen Sensoren zurück. Wiederhole das `GET /metrics/next/{xid}` Betrieb, um Metriken von zusätzlichen Sensoren zurückzugeben. Nachdem alle Metriken abgerufen wurden, gibt der Vorgang Null zurück.
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

Zeiteinheit	Einheitensuffix
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"]
}
```

Eingabe der Netzwerklokalität

Sie können eine Liste verwalten, die die Netzwerklokalität von IP-Adressen angibt.

Sie können beispielsweise einen Eintrag in der Netzwerklokalisierungsliste 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:

Betrieb	Beschreibung
GET /networkalities	Ruft alle Netzwerk-Lokalitätseinträge ab.
POST /Netzwerklocations	Erstellen Sie einen Eintrag für die Netzwerklokalität.
/networklocalities/ {id} LÖSCHEN	Löscht einen Eintrag für die Netzwerklokalität. **Hinweis** Dieser Vorgang ist bei Sensoren, die an Reveal (x) 360 angeschlossen sind, nicht verfügbar. Diese Operation ist jedoch verfügbar in Reveal (x) 360 REST-API.
GET /networklocalities/ {id}	Ruft einen bestimmten Eintrag für die Netzwerklokalität ab.
PATCH /networklocalities/ {id}	Wenden Sie Aktualisierungen auf einen bestimmten Netzwerklokalitätseintrag an. **Hinweis** Dieser Vorgang ist bei Sensoren, die an Reveal (x) 360 angeschlossen sind, nicht

Betrieb	Beschreibung
	verfügbar. Diese Operation ist jedoch verfügbar in Reveal (x) 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.

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.

Betrieb	Beschreibung
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.
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:` ***Boolescher Wert***

(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:` ***Boolescher Wert***

Gibt an, ob mehrere gleichzeitige HTTP-Verbindungen aktiviert sind, was die Durchsatzgeschwindigkeit verbessern 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` 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` 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 Azure AD-Authentifizierung erforderlich.

Die folgenden Werte sind gültig:

- `client`
- `resource_owner`

`client_id:` ***Schnur***

(Optional) Die Client-ID. Diese Option ist für die Azure AD- und Crowdstrike-Authentifizierung erforderlich.

`client_secret:` ***Schnur***

(Optional) Der geheime Schlüssel des Client. Diese Option ist für die Azure AD- und Crowdstrike-Authentifizierung erforderlich.

`resource:` ***Schnur***

(Optional) Der Azure AD-Ressourcen-URI. Diese Option ist für die Azure AD-Authentifizierung erforderlich.

`token_endpoint:` ***Schnur***

(Optional) Der Azure AD /Token-Endpunkt. Zum Beispiel: "https://login.microsoftonline.com/ <tenant_id>/oauth2/token“. Diese Option ist für die Azure AD-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***

Eine Reihe 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:` ***Boolescher Wert***

(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 anhand 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 Hash-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:` ***Boolescher Wert***

(Optional) Gibt an, ob Daten mit TLS verschlüsselt sind.

`skip_cert_verification:` ***Boolescher Wert***

(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) Eine Reihe 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:` ***Boolescher Wert***

(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 die Erstellung 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 umfangreich 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:` ***Boolescher Wert***

(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 Byte, die gleichzeitig an den Syslog-Server gesendet werden sollen.

`concurrent_connections:` ***Zahl***

(Optional) Die Anzahl der gleichzeitigen Verbindungen, über die Nachrichten gesendet werden sollen.

`localtime:` ***Boolescher Wert***

(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:` ***Boolescher Wert***

(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 anhand 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.

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 Plattenspeicher 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:

Betrieb	Beschreibung
GET /records/cursor/ {cursor}	Veraltet. Ersetzt durch `POST /records/cursor`.
POST /Datensätze/Cursor	Ruft Datensätze ab einem bestimmten Cursor ab.
POST /records/search	Führen Sie eine Protokollabfrage 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`

Zeiteinheit	Einheitensuffix
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"]
}
```

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.
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.

SSL-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:

Betrieb	Beschreibung
Holen Sie sich /ssldecryptkeys	Rufen Sie alle SSL-Entschlüsselungsschlüssel ab.
POST /ssldecryptkeys	Erstellen Sie einen neuen SSL-Entschlüsselungsschlüssel.
LÖSCHE /ssldecryptkeys/ {id}	Entfernen Sie einen SSL-Schlüssel aus dem ExtraHop-System.
GET /ssldecryptkeys/ {id}	Rufen Sie ein SSL-PEM und Metadaten ab.
PATCH /ssldecryptkeys/ {id}	Aktualisieren Sie einen vorhandenen SSL-Entschlüsselungsschlüssel.
GET /ssldecryptkeys/ {id} /protokolle	Alles abrufen Protokolle einem SSL-Entschlüsselungsschlüssel zugewiesen.
POST /ssldecryptkeys/ {id} /protokolle	Erstellen Sie ein neues Protokoll für einen SSL-Entschlüsselungsschlüssel.
LÖSCHEN Sie /ssldecryptkeys/ {id} /protocols/ {Protokoll}	Löscht ein Protokoll aus einem SSL-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 Kennzeichnen 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.
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 Bedrohungssammlungen wird von der Sicherheits-Community für Ihr Reveal (x) -System angeboten.

- Sie müssen Bedrohungssammlungen einzeln auf Ihre Command-Appliance oder Reveal (x) 360 und auf alle verbundenen Geräte hochladen Sensoren.
- Benutzerdefinierte Bedrohungssammlungen müssen in Structured Threat Information eXpression (STIX) als TAR.GZ -Dateien formatiert werden. Reveal (x) unterstützt derzeit STIX Version 1.0 — 1.2.
- Sie können Bedrohungssammlungen direkt auf Reveal (x) 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 von Observables, die eine Bedrohungssammlung enthalten kann, hängt von Ihrer Plattform und Lizenz ab. Weitere Informationen erhalten Sie von Ihrem ExtraHop-Vertreter.

Hinweis: Dieses Thema gilt nur für ExtraHop Reveal (x) 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:

Betrieb	Beschreibung
GET /threatcollections	Rufen Sie alle Bedrohungssammlungen ab.

Betrieb	Beschreibung
POST/Bedrohungssammlungen	Erstellen Sie eine neue Bedrohungssammlung.
/threatcollections/ {id} LÖSCHEN	Löscht 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 vorhanden ist, wird die bestehende Bedrohungssammlung überschrieben.
GET /threatcollections/ {id} /observables	Ruft die Anzahl der aus einer Bedrohungssammlung geladenen STIX-Observables ab, 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.

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.
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:` ***Schnur***

Der Name der Benutzergruppe.

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

Zeigt an, ob die Benutzergruppe aktiviert ist.

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

```
{
    "enabled": true,
    "name": "string"
}
```

`POST /usergroups/refresh`

Für diesen Vorgang gibt es keine Parameter.

`PATCH /usergroups/{id}`

Geben Sie die folgenden Parameter an.

`body:` ***Objekt***

Der Eigenschaftswert wird für die spezifische Benutzergruppe aktualisiert.

`id:` ***Schnur***

Die eindeutige Kennung für die Benutzergruppe.

`GET /usergroups/{id}`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***

Die eindeutige Kennung 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:` ***Schnur***
Die eindeutige Kennung für die Benutzergruppe.

`DELETE /usergroups/{id}/associations`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Benutzergruppe.

`POST /usergroups/{id}/refresh`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Benutzergruppe.

`GET /usergroups/{id}/members`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Benutzergruppe.

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

```
{
    "users": {}
}
```

`PATCH /usergroups/{id}/members`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Benutzergruppe.

`body:` ***Schnur***
Ein Objekt, das angibt, welche Benutzer zugewiesen oder welche Zuweisung aufgehoben werden sollen. 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 trennt Bob von der Gruppe.

`PUT /usergroups/{id}/members`

Geben Sie die folgenden Parameter an.

`id:` ***Schnur***
Die eindeutige Kennung für die Benutzergruppe.

`body`: ***Schnur***

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.