

# Erstellen Sie einen Auslöser, um benutzerdefinierte Metriken für HTTP 404-Fehler zu sammeln

Veröffentlicht: 2024-09-26

Wenn Ihre Kunden und Klienten können aufgrund von Webseitenfehlern nicht auf die benötigten Informationen zugreifen. Sie können einen Auslöser schreiben, der Ihnen hilft, Antworten zu finden.

Zum Beispiel, wenn ein Kunde eine Seite anfordert, die Ihre HTTP Der Server kann nicht finden, der Server antwortet mit einem 404-Statuscode oder dem Fehler „Seite nicht gefunden“. Solche Antworten deuten in der Regel darauf hin, dass der Link, auf den der Kunde geklickt hat, zu einer ungültigen URI führt. Es wäre nützlich zu wissen, auf welche URI-Kunden zugreifen möchten, und den Referrer oder die Quelle des ungültigen Links zu finden.

In dieser exemplarischen Vorgehensweise erstellen Sie einen Auslöser, der die folgenden Fragen beantwortet:

- Erhalten meine Kunden die Fehlermeldung „Seite nicht gefunden“?
- Was ist der URI der Seite, die zu dem Fehler führt?
- Welcher Referrer hat den Fehler verursacht?


Sie erstellen einen Auslöser, der eine neue benutzerdefinierte Metrik generiert, die sowohl den ungültigen URI als auch den Referrer des ungültigen URI zurückgibt. Sie werden auch eine Anwendung erstellen, die einen maßgeschneiderten Überblick über Ihren Web-Traffic bietet, indem sie jedes Mal, wenn ein 404-Statuscode auf bestimmten Geräten auftritt, Webmetriken sammelt.

## Voraussetzungen

- Sie sollten sich mit den Konzepten in dieser Komplettlösung vertraut machen, indem Sie die [Erste Schritte mit Triggern](#) Abschnitt der [ExtraHop System-Benutzerhandbuch](#).
- Sie benötigen Zugriff auf ein ExtraHop-System mit einem Benutzerkonto mit eingeschränktem Schreib- oder [volle Schreibrechte](#).
- Ihr ExtraHop-System muss über Netzwerkdaten mit Webserver-Traffic verfügen.
- Es ist hilfreich, grundlegende JavaScript-Kenntnisse zu haben.

## Den Auslöser konfigurieren

In den folgenden Schritten benennen und beschreiben Sie den Auslöser, aktivieren das Debugging und konfigurieren den Auslöser für die Ausführung auf HTTP Reaktionsereignisse.

1. Klicken Sie auf das Symbol Systemeinstellungen , und klicken Sie dann auf **Auslöser**.
2. klicken **Erstellen**.
3. Geben Sie einen Namen für den Auslöser ein. Geben Sie für diese exemplarische Vorgehensweise ein `HTTP 404-Fehler`.
4. Geben Sie eine Beschreibung des Auslöser ein. Geben Sie für diese exemplarische Vorgehensweise ein `404-Fehler bis zur Quelle zurückverfolgen`.
5. klicken **Debug-Log aktivieren**.
6. Klicken Sie in den **Ereignisse** Feld und Auswahl **HTTP\_RESPONSE** aus der Liste.

Die folgende Abbildung zeigt die Trigger-Konfigurationsattribute, die Sie oben konfiguriert haben:

## Create Trigger

### Name

### Author

 ExtraHop 

### Description

### Assignments

### Options

- Enable trigger
- Enable debug log

### Events

[Show Advanced Options](#)

## Schreiben Sie eine Debug-Anweisung

Fügen Sie als Nächstes eine einfache Debug-Anweisung hinzu, um sich mit dem Layout und den Funktionen des Trigger-Editors vertraut zu machen.

Wenn der Auslöser ausgeführt wird, sucht er HTTP Antwortereignisse und prüft, ob der Statuscode in der Antwort 404 lautet. Wenn ein 404-Statuscode erkannt wird, fügt der Debug-Aufruf die URL, auf die der Benutzer zuzugreifen versucht hat, dem Debug-Log hinzu.

1. Klicken Sie auf **Herausgeber** Tabulatur.
2. Fügen Sie dem Editor den folgenden Code hinzu:

```
if (HTTP.statusCode === 404) {
  debug (HTTP.uri);
}
```



**Hinweis:** Der Editor bietet einige Funktionen zur automatischen Vervollständigung bei der Codeeingabe. Wenn Sie beispielsweise nach der Auswahl eines Klassenobjekts einen Punkt (.) eingeben, erhalten Sie eine Liste von Methoden und Eigenschaften, die für das HTTP-Objekt gelten. Sie können das gewünschte Element aus der Liste auswählen, wie in der folgenden Abbildung gezeigt:

```

1  if (HTTP.statusCode) {
2    debug (HTTP.);
3  }

```

debug(message: string): void

- rspTimeToLastByte
- rspVersion
- rspZeroWnd
- sqli
- statusCode
- streamId
- thinkTime
- title
- tprocess
- uri (property) HTTP.uri: string ⓘ
- userAgent
- xss

Klicken Sie auf den **ich** Das Symbol zeigt eine Beschreibung des aktuellen Elements an, wie in der folgenden Abbildung dargestellt:

HTTP.stat

- statusCode (property) HTTP.statusCode: number ⓘ

The HTTP status code of the response.

Access only on `HTTP_RESPONSE` events or an error will occur.

View the complete description in the [Trigger API Reference](#).

3. klicken **Speichern und schließen**.

## Weisen Sie den Auslöser Geräten zu

Bevor der Auslöser die Ausgabe der Debug-Anweisung protokollieren kann, müssen Sie den Auslöser mindestens einem Gerät zuweisen. Für diese exemplarische Vorgehensweise weisen Sie den Auslöser dem zu HTTP Server Gerätegruppe.

Nachdem der Auslöser zugewiesen wurde, führt das System den Auslöser jedes Mal aus, wenn eine HTTP-Antwort auf einem Server in dieser Gruppe auftritt.

**Wichtig:** Wenn Sie Ihre eigenen Trigger erstellen, weisen Sie Trigger nur den Geräten zu, die Sie benötigen, um die Leistungseinbußen Ihrer Trigger auf das System zu minimieren.

1. klicken **Vermögenswerte** aus dem oberen Menü.
2. klicken **Gerätegruppen**, und wählen Sie dann **HTTP-Server** Checkbox.
3. klicken **Trigger zuweisen** vom Seitenanfang aus, um eine Liste der Auslöser zu öffnen, die für eine Zuweisung in Frage kommen.
4. Wählen Sie den **HTTP 404-Fehler** Auslöser und dann klicken **Trigger zuweisen**.
5. Klicken Sie auf **Systemeinstellungen** Symbol, zurück zum Auslöser Seite, und wählen Sie die **HTTP 404-Fehler** Auslöser.
6. In der Zuweisungen Feld, stellen Sie sicher, dass die ausgewählten HTTP-Server hinzugefügt wurden.

Das Feld sollte der folgenden Abbildung ähneln:

## Assignments

HTTP Servers



Add Source

## Debug-Ausgabe anzeigen

Nachdem Sie den Auslöser zugewiesen haben HTTP Auf Servern führt das System den Auslöser, wenn HTTP-Antwortverkehr auftritt, und wenn eine Antwort einen 404-Statuscode enthält, sendet das System die Ergebnisse an das Debug-Log.

Um die Ergebnisse der Debug-Anweisung anzuzeigen, klicken Sie auf **Debug-Protokoll** Registerkarte, die der folgenden Abbildung ähneln sollte:

PROBLEMS	DEBUG LOG
✖ 0 ⚠ 0	
[Wed Jun 12 15:25:42]	www.extrahop.com/
[Wed Jun 12 15:25:48]	www.bing.com/AS/Suggestions
[Wed Jun 12 17:25:23]	www.arena.altitudeplatform.com/vast
[Wed Jun 12 15:25:49]	www.bing.com/AS/Suggestions

Die Debug-Ausgabe beginnt, sobald der Auslöser zugewiesen und gespeichert wurde. Das Protokoll kann jedoch keine Daten von 404-Antworten anzeigen, die vor der Zuweisung und Speicherung des Auslöser aufgetreten sind.

## Erstellen Sie eine benutzerdefinierte Metrik

Die Ergebnisse der Debug-Anweisung haben bestätigt, dass es URIs gibt, die zu 404-Statuscodes führen. In diesem Abschnitt erstellen Sie eine benutzerdefinierte Metrik mit dem Namen „404UriandReferrer“, um die ungültigen URIs und die entsprechenden Referrer zu extrahieren.

Die benutzerdefinierte Metrik gibt sowohl den ungültigen URI als auch den Referrer des Links zurück, sodass Sie die beiden Datensätze in einem einzigen Ergebnis extrahieren können.

Um eine benutzerdefinierte Metrik zu erstellen, müssen Sie eine Metrik angeben Quelle aus dem die Daten extrahiert werden, z. B. eine Anwendung, ein Netzwerk oder ein Gerät. In dieser exemplarischen Vorgehensweise erstellen Sie eine Anwendung mit dem Namen „Datei nicht gefunden“ als Quelle für die benutzerdefinierte Metrik.

1. Fügen Sie im rechten Bereich den folgenden, grün hervorgehobenen Triggercode zum vorhandenen Skript hinzu:

```
if (HTTP.statusCode === 404){
  debug (HTTP.uri);
  var app = Application("File Not Found");
}
```

Dieser Code deklariert die Anwendung als Variable und gibt den Namen der Anwendung an. Es wird empfohlen, eine Variable für Methoden zu deklarieren, die Sie mehrmals aufrufen möchten, um den

Ressourcenverbrauch durch den Auslöser zu reduzieren. In dieser exemplarischen Vorgehensweise ruft der Code die Anwendung zweimal auf.

2. Fügen Sie dem vorhandenen Skript den folgenden grün hervorgehobenen Triggercode hinzu:

```
if (HTTP.statusCode === 404){
  debug (HTTP.uri);
  var app = Application("File Not Found");
  app.metricAddDetailCount(
    "404UriAndReferrer",
    "404:" + HTTP.uri + " | REFERRER:" + HTTP.referrer,
    1);
}
```

Die `metricAddDetailCount` Methode gibt den Metriknamen und die von der Metrik zurückgegebenen Ereignisseigenschaftendaten an. Die Methode spezifiziert auch das Format der extrahierten Daten als `404: <uri> | REFERRER: <uri>` wenn es in einem Diagramm angezeigt wird. Sehen Sie die [Bewerbung](#) Abschnitt der [ExtraHop Trigger API-Referenz](#) für weitere Informationen über benutzerdefinierte Metrik Methoden.


3. klicken **Speichern**.

Nachdem der Code gespeichert wurde, fügt das ExtraHop-System die neue benutzerdefinierte Metrik dem Metrikkatalog hinzu und erstellt die Anwendung. Zu diesem Zeitpunkt ist die Anwendung nur eine Quelle für die benutzerdefinierte Metrik; die Anwendung sammelt keine integrierten Metriken. Sehen Sie die [Metrischer Katalog](#) Abschnitt der [ExtraHop System-Benutzerhandbuch](#) und der [Bewerbung](#) Abschnitt der [ExtraHop Trigger API-Referenz](#) für weitere Informationen.

## Die benutzerdefinierte Metrik in einem Diagramm anzeigen

Um Ihre benutzerdefinierte Metrik anzuzeigen, müssen Sie sie einem Diagramm-Widget auf einem hinzufügen Dashboard. In der Tabelle zeigt die benutzerdefinierte Metrik, welche URIs ungültig sind und welchen Referrer die einzelnen ungültigen Links haben.

Weitere Informationen zum Anzeigen von Metriken auf einem Dashboard finden Sie in [Erste Schritte mit Dashboards](#) Abschnitt der [ExtraHop System-Benutzerhandbuch](#).

1. Auf dem Armaturenbrett Seite, klicken Sie auf das Befehlsmenü  in der oberen rechten Ecke und wählen Sie **Diagramm erstellen**.
2. Klicken Sie in Ihrem neu erstellten Dashboard auf das leere Diagramm-Widget, um das zu starten Metric Explorer.
3. klicken **Quelle hinzufügen**.
4. Geben Sie ein und wählen Sie **Datei wurde nicht gefunden**, das ist die Anwendung, die durch den Auslöser erstellt wurde.
5. In der Metriken Feld, Typ 404 und wählen Sie dann **404 Uri und Referrer**, das ist die benutzerdefinierte Metrik, die durch den Auslöser erstellt wurde.

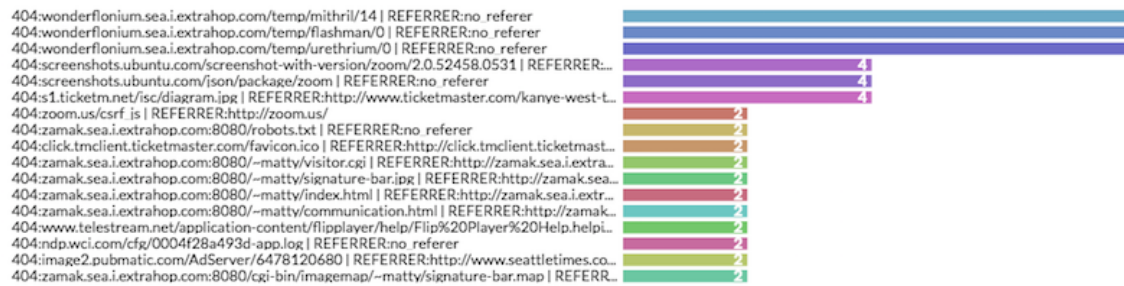


**Hinweis** Das ExtraHop-System konvertiert den Namen der benutzerdefinierten Metrik, den Sie im Code angegeben haben, in einen benutzerfreundlichen Anzeigenamen. In dieser exemplarischen Vorgehensweise wird „404UriandReferrer“ in „404 Uri And Referrer“ umgewandelt. Sie können den Anzeigenamen für die Anzeige in der Metrischer Katalog. Sehen Sie die [Metrischer Katalog](#) Abschnitt der [ExtraHop System-Benutzerhandbuch](#) für weitere Informationen.

6. Wählen Sie den **Bar** Diagrammtyp am unteren Rand des Metric Explorers.
7. Klicken Sie in der unteren rechten Ecke **Zum Dashboard hinzufügen** und wähle **Dashboard erstellen**.
8. Geben Sie einen Namen für Ihr Dashboard in der Titel Feld. Geben Sie für diese exemplarische Vorgehensweise ein `HTTP 404-Fehler`.
9. klicken **Erstellen**.

Das Diagramm mit Daten, die aus der benutzerdefinierten Metrik extrahiert wurden, wird automatisch zum neuen Dashboard hinzugefügt. Die Dashboard-Ausgabe sollte der folgenden Abbildung ähneln:

#### File Not Found 404 Uri And Referrer



## Sammeln Sie integrierte Metriken in der Anwendung

In den vorherigen Schritten haben Sie dem Skript Code hinzugefügt, um eine Anwendung als Quelle Ihrer neuen benutzerdefinierten Metrik. In diesem Abschnitt fügen Sie Code hinzu, der es der Anwendung ermöglicht, integrierte Daten zu sammeln HTTP Metriken, aber keine benutzerdefinierten Metriken, um jedes Mal, wenn ein 404-Statuscode auf den angegebenen Geräten auftritt, einen maßgeschneiderten Überblick über Ihren Web-Traffic zu erhalten.

1. In der Trigger konfigurieren Fenster, klicken Sie auf **Herausgeber** Tabulatur.
2. Fügen Sie dem vorhandenen Skript den folgenden grün hervorgehobenen Triggercode hinzu:

```
if (HTTP.statusCode === 404){
  debug (HTTP.uri);
  var app = Application("File Not Found");
  app.metricAddDetailCount(
    "404UriAndReferrer",
    "404:" + HTTP.uri + " | REFERRER:" + HTTP.referrer,
    1);
  app.commit();
}
```

3. klicken **Speichern und schließen**.

## Anwendung- und Metrikseiten anzeigen

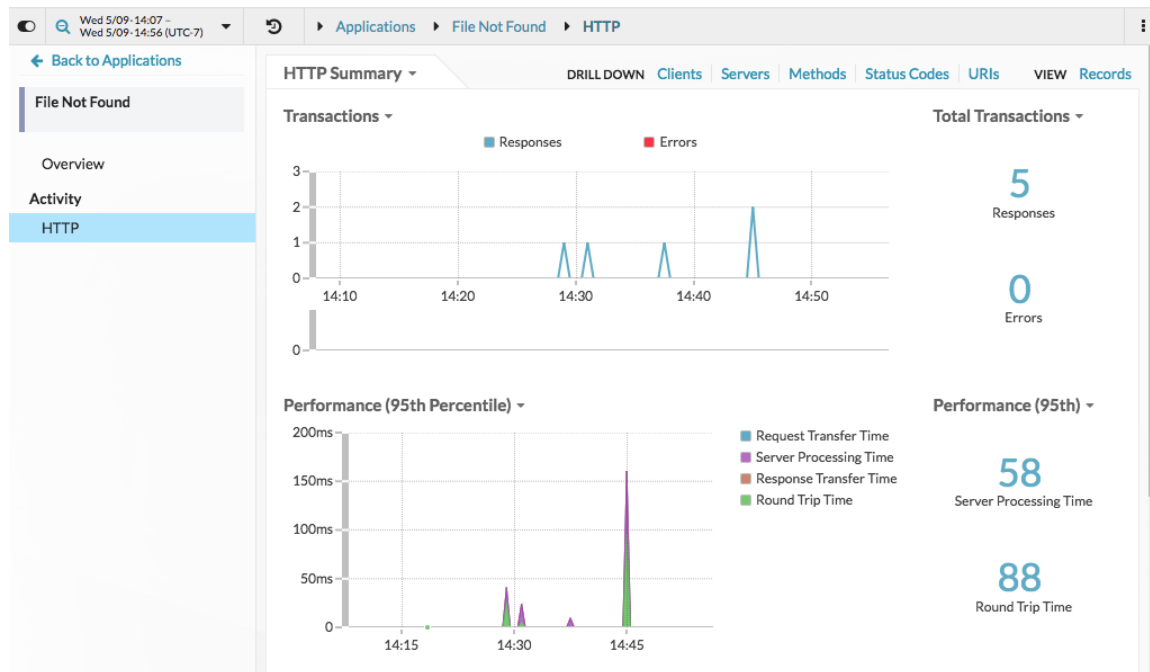
Die neue Anwendung „Datei nicht gefunden“ wird erstellt, wenn der Auslöser zum ersten Mal ausgeführt wird und die vom Triggerskript angegebenen Bedingungen erfüllt sind. Nachdem Sie die Anwendung festgeschrieben haben, kann es mehrere Minuten dauern, bis der Auslöser ausgeführt wird und Daten in der Anwendung verfügbar sind.

Die festgeschriebene Anwendung fügt bei jeder Ausführung des Triggers integrierte Metrikdaten hinzu, die für die von Ihrem Trigger definierten Bedingungen relevant sind. In dieser exemplarischen Vorgehensweise ist integriert HTTP Metriken werden der Anwendung jedes Mal hinzugefügt, wenn eine HTTP-Antwort auf den ausgewählten Geräten zu einem 404-Statuscode führt

Die Anwendung zeigt die gesammelten Daten auf einer oder mehreren Protokollseiten an. Protokollseiten werden automatisch in Abhängigkeit von den Protokollobjekten erstellt, die in Ihrem Auslöser enthalten sind. In dieser exemplarischen Vorgehensweise enthält die Anwendung „Datei nicht gefunden“ eine Protokollseite namens Web, auf der integrierte HTTP-Metriken angezeigt werden, da der Auslöser mit HTTP-Antworten ausgeführt wird. Wenn Sie einen Auslöser erstellen, der für die Ausführung bei TLS-Ereignissen konfiguriert ist, zeigt die Anwendung eine Seite mit integrierten TLS-Metriken an, und ein Auslöser, der bei Flow-Ereignissen ausgeführt wird, zeigt eine Seite mit integrierten L4-Metriken an.

1. klicken **Vermögenswerte** aus dem oberen Menü.
2. Klicken Sie im linken Bereich auf **Anwendungen**, und wählen Sie dann den **Datei wurde nicht gefunden** Anwendung.
3. Klicken Sie im linken Bereich auf **HTTP** um die in der Anwendung gesammelten integrierten HTTP-Metriken anzuzeigen.

Die Webseite sollte der folgenden Abbildung ähneln:



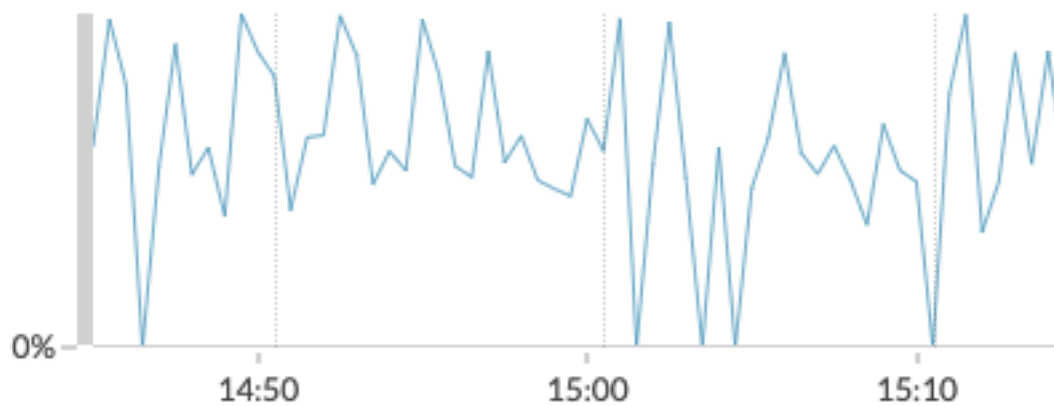
## Überprüfen Sie die Leistung des Auslöser

Trigger sind ein leistungsstarkes Tool, das detaillierte Einblicke in Ihre Umgebung bieten kann. Trigger verbrauchen jedoch Ressourcen und beeinträchtigen die Systemleistung. In diesem Abschnitt überprüfen Sie, wie sich der Auslöser auf die Leistung auswirkt, und erfahren, welche kleinen Änderungen Sie vornehmen können, um die Leistung zu verbessern.

Das Trigger-Leistungsdigramm gibt die Anzahl der Zyklen an, die der Auslöser innerhalb des angegebenen Zeitintervalls verbraucht hat. Sie können den Mauszeiger über einen Datenpunkt bewegen, um Details zur Triggerleistung zu einem bestimmten Zeitpunkt anzuzeigen.

In der Trigger bearbeiten Fenster, betrachten Sie die **Triggerlast erfassen** Das Diagramm sieht der folgenden Abbildung ähnlich:

## Capture Trigger Load ?



**Hinweis** Auf der Seite Systemstatus in den Systemeinstellungen finden Sie zusätzliche Leistungsdiagramme für Auslöser, mit denen Sie die kumulative Wirkung all Ihrer Trigger auf das System überwachen können. Sehen Sie die [Systemstatus-Dashboard](#) Thema.

Versuchen Sie, die folgenden Änderungen vorzunehmen, und überprüfen Sie dann das Leistungsdiagramm, um alle Änderungen zu sehen:

- Kommentieren Sie die Debug-Anweisung aus. Sie haben überprüft, ob der Auslöser funktioniert und die gewünschten benutzerdefinierten und integrierten Metriken sammelt. Sie benötigen die Debug-Ausgabe nicht mehr.
- Weisen Sie den Auslöser bestimmten Geräten statt allen neu zu HTTP Server in der Gerätegruppe.

Weitere Tipps und Tricks zur Trigger-Optimierung finden Sie in [ExtraHop Trigger API-Referenz](#) und die folgenden Blogbeiträge zu Trigger Optimization 101 in der [ExtraHop Community-Foren](#):

- [Trigger-Optimierung 101: Zugriff auf Metriken](#)
- [Trigger-Optimierung 101: Schnell zurückkehren](#)
- [Trigger-Optimierung 101: Ausnahmebehandlung](#)
- [Trigger-Optimierung 101: Return oder Exit \(\)?](#)