
Trigger
Release latest

unknown

23.11.2022

1	Einführung	3
2	Systemvoraussetzungen	5
3	Installation	7
4	Einrichtung	9
5	Übersicht	13
6	HTTP-Requests	15
7	Shell-Skripte	17
8	Platzhalter	19
9	Buttons	23
10	Rechteverwaltung	25
11	Changelog	27
12	License	29

Mit dem Trigger Add-on können aus i-doit heraus manuell Webrequests oder Shellskripte abgeschickt (getriggert) werden. Über Platzhalter können die Requests automatisch mit Werten aus den markierten oder aktiven Objekten gefüllt werden.

Mit dem Trigger Add-on können aus i-doit heraus manuell Webrequests oder Shellskripte abgeschickt (getriggert) werden. Über Platzhalter können die Requests automatisch mit Werten aus den markierten oder aktiven Objekten gefüllt werden. Mögliche Usecases wären z.B.:

- Issues/Requests an ein Servicedesk/Ticketsystem schicken
- Objektdaten in andere Systeme syncen
- Webhooks in Opencelium aufrufen, um komplexere Datentransfers umzusetzen
- Eine Downtime/ein Wartungsfenster an das Monitoring schicken
- Ein Skript zum Deployen von Virtuellen Maschinen starten

Das Recht zum Aufrufen eines Triggers kann je Trigger einzeln konfiguriert werden. Über die Button-Konfiguration kann konfiguriert werden, wo die Buttons zum Aufrufen eines Triggers eingeblendet werden und welche Trigger unter einem gemeinsamen Aufrufbutton zusammengefasst werden.

Systemvoraussetzungen

Als Add-on für die i-doit CMDB setzt das Trigger-Add-on natürlich ein funktionsfähiges i-doit voraus.

Die aktuelle Version des Trigger-Add-on setzt mindestens die i-doit Version 1.17.2 voraus.

Neben den i-doit Standardvoraussetzungen muss die curl-Bibliothek in PHP zur Verfügung stehen.

Bemerkung: Diese Dokumentation bezieht sich -wenn nicht anders erwähnt- immer auf die aktuelle Version des Trigger-Add-on.

Die Installation des Trigger Add-ons entspricht dem Standardvorgehen für die Installation von i-doit Add-ons:

- Einloggen in des i-doit Admin-Center
- Auf den Reiter „Add-ons“ gehen
- Auf den Button „Install/update Add-on“ klicken
- Das ZIP-Paket des Add-ons auswählen
- Auf den Knopf „Upload und install“ klicken
- Fertig

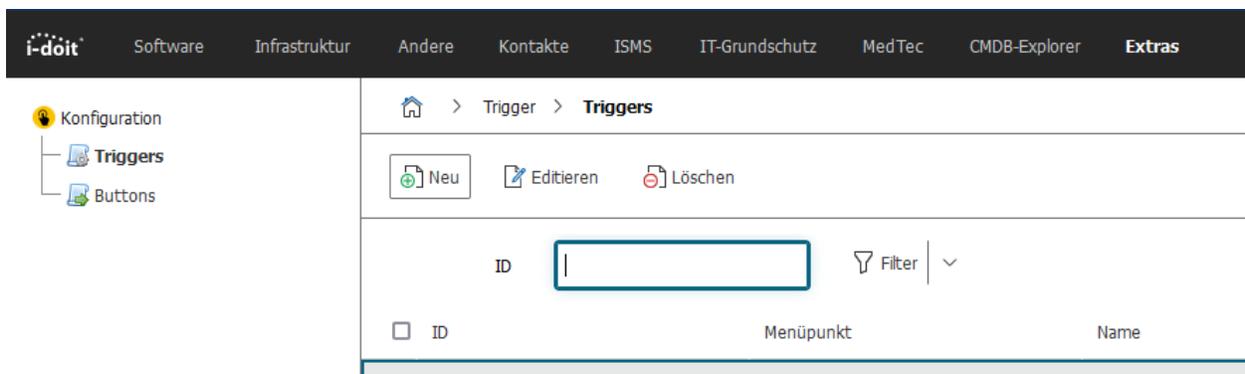
Nach der Installation muss das Trigger-Add-on noch konfiguriert werden. Informationen dazu finden Sie unter *Einrichtung*.

Die Einrichtung eines Triggers besteht aus zwei Teilen:

- Anlegen eines Triggers
- Konfigurieren eines Buttons und Zuordnung eines Triggers zu einem Button

4.1 Anlegen eines Trigger:

Unter Extras Trigger kommt man in den Konfigurationsbereich des Trigger-Add-ons. Das Anlegen und Bearbeiten von Triggern findet man unter dem Punkt „Triggers“.



Ein Trigger hat sechs Attribute, bei HTTP-Requests kann außerdem noch eine Liste an cURL-Optionen konfiguriert werden:

Trigger-Konfiguration

Name	<input type="text"/>
Token	LFV4AVB2B4CK1ZGVOIVXB3
Aufruf-Art	<input type="text" value="HTTP/HTTPS"/>
Ziel	<input type="text"/>
Parameter	<input type="text"/>
Optionen <small>Beschreibung cURL-Optionen</small>	<input type="button" value="⊕ Neue Option"/>
Menü-Icon	<div> JDisc.png  OpenCelium.png  checkmk.png  i-doit.png</div>

- Name: Name des Triggers.
- Token: Der eindeutige Token wird beim Anlegen automatisch generiert und kann hinterher nicht mehr verändert werden.
- Aufruf-Art: Ein Trigger kann entweder einen [HTTP-Request](#) absetzen als auch ein [Shell-Skript](#) starten.
- Ziel: Das Ziel ist bei HTTP-Requests die Ziel-URL und bei Shell-Skripten der Pfad des aufzurufenden Skripts.
- Parameter: Bei HTTP-Requests können hier GET-Parameter eingegeben werden und bei Shellskripten können hier Parameter eingegeben werden, die dem Skript übergeben werden sollen.
- Optionen (nur bei HTTP-Requests): Hier können die Werte für eine oder mehrere cURL-Optionen gesetzt werden.
- Menü-Icon: Hier kann ein Icon ausgewählt werden, was auf dem Aufrufbutton angezeigt wird. Trigger mit gleichen Icons werden gruppiert. Wenn eigene (16x16) Icons hochgeladen werden, dann bitte beachten, dass der Dateiname des Icons (ohne Dateiendung) als Buttontext verwendet wird.

Bei den Attributen Ziel, Parameter und bei den cURL-Optionen können *Platzhalter* eingesetzt werden, die dann zum Zeitpunkt des Aufrufes mit den Werten des aktuellen/selektierten Objektes ersetzt werden.

In der i-doit *Rechteverwaltung* kann neben den Zugriffsrechten für die Trigger-Konfiguration selbst das Ausführrecht für jeden einzelnen Trigger eingerichtet werden.

4.2 Konfigurieren eines Buttons und Zuordnung eines Triggers zu einem Button

In der Triggerkonfiguration gibt es noch einen weiteren Punkt Buttons, in dem man konfigurieren kann, wann die Trigger-Aufrufbuttons angezeigt werden. Dazu hat man die Möglichkeit, Objekttypen, globale Kategorien, spezifischen Kategorien und benutzerdefinierte Kategorien auszuwählen.

Dabei gibt mehrere Kombinationsmöglichkeiten:

- Wenn man dabei nur einen oder mehrere Objekttypen angibt (aber keine Kategorien), dann wird der Button nur in der Objektlistenansicht der jeweiligen Objekttypen angezeigt, nicht jedoch, wenn man in einem Objekt eines der Objekttypen drin ist.
- Wenn man nur eine oder mehrere Kategorien angibt, dann wird der Button nur angezeigt, wenn man in der entsprechenden Kategorien drin ist, unabhängig vom Objekttypen
- Und wenn man einen oder mehrere Objekttypen und eine oder mehrere Kategorien ausgewählt hat, dann wird der Button nur angezeigt, wenn man in einer der ausgewählten Kategorien eines Objektes von einem der ausgewählten Typen drin ist.

Zu der Objekttyp- und Kategoriekonfiguration kann man dann einen oder mehrere Trigger auswählen, die bei dieser Kombination angezeigt werden sollen (die entsprechenden Ausführrechte vorausgesetzt). Man hat außerdem noch die Möglichkeit, in dieser Kombination den Text auf dem Aufruflink anzugeben. Standardmäßig ist hier der Name des Triggers ausgefüllt.

KAPITEL 5

Übersicht

TBD

HTTP-Requests

Wenn als Aufrufart „HTTP/HTTPS“ ausgewählt ist, wird beim Aufrufen des Triggers ein HTTP-Request an die URL geschickt, die unter „Ziel“ angegeben ist.

GET-Parameter können entweder als Teil der Ziel-URL angegeben werden:

Aufruf-Art	<input type="text" value="HTTP/HTTPS"/>
Ziel	<input type="text" value="https://localhost/myApplication?param1=test&param2=test2"/>
Parameter	<input type="text"/>

oder aber im Feld „Parameter“ eingegeben werden:

Aufruf-Art	<input type="text" value="HTTP/HTTPS"/>
Ziel	<input type="text" value="https://localhost/myApplication"/>
Parameter	<input type="text" value="param1=test&param2=test2"/>

Standardmäßig werden Requests per GET aufgerufen, über die (cURL-)Optionen können aber auch andere HTTP-Methoden benutzt werden.

Um beispielsweise Formulare Daten per POST zu übertragen, muss die Option `CURLOPT_POST = 1` gesetzt werden. Über die Option `CURLOPT_POSTFIELDS` können die zu übertragenden Werte angegeben werden, entweder als URL-kodierter String:

Trigger, Release latest

Optionen

Beschreibung cURL-Optionen

CURLOPT_POST 1 

CURLOPT_POSTFIELDS param1=test¶m2=test 

oder als Array in JSON-Notation:

Optionen

Beschreibung cURL-Optionen

CURLOPT_POST 1 

CURLOPT_POSTFIELDS

```
{
  "param1": "test",
  "param2": "test2"
}
```



CURLOPT_HTTPHEADER ["Content-Type:application\json"] 

Bemerkung: Da die HTTP-Requests vom i-doit-Server abgeschickt werden, muss der i-doit-Server die DNS-Auflösung der Ziel-URL vornehmen können und natürlich die Zieladresse auch erreichen können.

KAPITEL 7

Shell-Skripte

TBD

Mit Platzhaltern können das Ziel, die Parameter und die cURL-Optionen eines Triggers mit Werten des aktuellen Objektes ersetzt werden. Platzhalter beginnen und Enden immer mit einem „%“. Wenn ein Platzhalter mehrere Werte zurückliefert, werden die einzelnen Werte kommasepariert angegeben. Einige Parameter unterstützen außerdem die Arraynotation, d.h. wenn der Platzhalter zusätzlich noch mit eckigen Klammern angegeben ist, dann wird der Platzhalter mit einem JSON-Array-String ausgetauscht (auch wenn nur ein Objekt ausgewählt ist). Einige Platzhalter können nur in bestimmten Kontexten ersetzt werden. Folgende Parameter werden derzeit angeboten:

8.1 %CATEGORY_CONSTANT.attribute_key%

Attribut einer Kategorie. Bei Multivaluekategorien wird der selektierte/aktive Wert genommen, wenn man sich in der Kategorie befindet, ansonsten der erste Eintrag. Die Kategoriekonstanten und Attribut-Keys findet man hier:

Verwaltung JSON-RPC API Kategorien und Attribute

Beispiel: Der Platzhalter für die Seriennummer aus der Kategorie Modell lautet: %C__CATG__MODEL.serial%

Array-Notation: Nicht unterstützt

8.2 %ipv4%

IPv4-Adresse des aktuellen Hostadressen- oder IP-Listen-Eintrags. Wenn man nicht in der Hostadressen-Einzelansicht ist, wird die primäre IP-Adresse benutzt.

Array-Notation: [%ipv4%]

8.3 %hostname%

Hostname des aktuell besuchten Hostadressen- oder IP-Listen-Eintrags. Wenn man nicht in der Hostadressen-Einzelansicht ist, wird der Hostname des primären Eintrags benutzt.

Array-Notation: [%hostname%]

8.4 %fqdn%

FQDN des aktuell besuchten Hostadressen-Eintrags. Wenn man nicht in der Hostadressen-Einzelansicht ist, wird der FQDN des primären Eintrags benutzt.

Array-Notation: [%fqdn%]

8.5 %objid_sel%

Objekt ID des ausgewählten IP-Listen-Eintrags. Steht nur in der Kategorie „IP-Liste“ zur Verfügung.

Array-Notation: [%objid_sel%]

8.6 %objtitle_sel%

Objekt Titel des ausgewählten IP-Listen-Eintrags. Steht nur in der Kategorie „IP-Liste“ zur Verfügung.

Array-Notation: [%objtitle_sel%]

8.7 %catgid%

Kategorie ID der aktuellen globalen Kategorie. Wird nur in globalen Kategorien unterstützt.

Array-Notation: Nicht unterstützt

8.8 %catsid%

Kategorie ID der aktuellen spezifischen Kategorie. Wird nur in spezifischen Kategorien unterstützt.

Array-Notation: Nicht unterstützt

8.9 %catid%

Kategorie ID der aktuellen benutzerdefinierten Kategorie. Wird nur in benutzerdefinierten Kategorien unterstützt.

Array-Notation: Nicht unterstützt

8.10 %cateid%

Daten-ID des aktuellen Kategorieeintrags. Wird nur bei Multivalue-Kategorien unterstützt.

Array-Notation: Nicht unterstützt

8.11 %username%

Username des aktuell eingeloggten Users

Array-Notation: Nicht unterstützt

8.12 %usermail%

Email-Adresse des aktuell eingeloggten Users

Array-Notation: Nicht unterstützt

KAPITEL 9

Buttons

TBD

KAPITEL 10

Rechteverwaltung

In der i-doit Rechteverwaltung können unter dem Punkt Trigger folgende Rechte erteilt werden:

      	Bedingung	Parameter	Aktion
<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	bezieht sich auf <input type="text" value="Trigger konfigurieren"/> 		
<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	bezieht sich auf <input type="text" value="Buttons konfigurieren"/> 		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	bezieht sich auf <input type="text" value="Trigger aufrufen"/> 	<input checked="" type="checkbox"/> Alle <input type="text" value="Bitte Trigger auswählen ..."/>	

Das Ausführrecht für konfigurierte Trigger kann sowohl pauschal für alle Trigger eingerichtet werden oder es können explizit die Trigger ausgewählt werden, die der Benutzer oder die Benutzergruppe ausführen darf.

Bemerkung: Die Personengruppe „admin“ erhält bei Installation des Trigger Add-on automatisch die oben abgebildeten Rechte.

Warnung: Die Vergabe von Rechten zur Konfiguration von Triggern sollte wohl überlegt sein, da Trigger ein mächtiges Werkzeug darstellen, mit dem ggf. sensible Informationen vom Dateisystem ausgelesen werden können.

KAPITEL 11

Changelog

11.1 Version 1.0

11.1.1 Deutsch

Initiales Release

11.1.2 English

Initial release

KAPITEL 12

License

becon © 2013-2022 becon GmbH