

Lua-Scripts - Tipps & Tricks (Cheat Sheet)

Inhalt

Sammlung von Hilfestellungen und häufig gestellter Fragen für das Schreiben von Scripts.

- [Inhalt](#)
- [Häufig gestellte Fragen](#)
 - [Ich weiß nicht, wie die Event-Daten aussehen \(JSON-Daten loggen\)](#)
 - [Wie greife ich auf ausgefüllte Formular-Felder zu?](#)
 - [Ich weiß nicht, wie der versendete Request aussieht \(webhook.site\)](#)
 - [Wie prüfe ich auf leere / null Felder?](#)
 - [Verwandte Artikel](#)



Datenschutz: Testeinheiten verwenden

Wir weisen darauf hin, dass zum Entwickeln von Scripts stets eine separate Testeinheit und Testdaten verwendet werden sollte.

Zögern Sie nicht hierzu eine neue Einheit zu registrieren und als Notiz: "Testeinheit zur Entwicklung von User-Scripts" zu verwenden.

Übertragen Sie niemals produktive Einsatzdaten an ungeprüfte Drittsysteme!

Häufig gestellte Fragen

Ich weiß nicht, wie die Event-Daten aussehen (JSON-Daten loggen)

Nutze in der "Script testen"-Funktion `print(json.encode(event))` um alle Daten als json in der Ausgabe zu haben.

Wie greife ich auf ausgefüllte Formular-Felder zu?

Auf ausgefüllte Felder, die im Tab "Fomular" definiert wurden, können über "configuration.key" zugegriffen werden. Bei nachfolgender Konfiguration also über `print(configuration.url)`

```
[
  {
    "type": "text",
    "key": "url",
    "label": "URL",
    "description": "URL des Servers an den Sie Einsätze übertragen möchten."
  }
]
```

Ich weiß nicht, wie der versendete Request aussieht (webhook.site)

Um die alle Header und den Body von durch `fetch` erzeugte http(s)-Anfrage zu inspizieren, kann <https://webhook.site> (ext. US-amerikanischer Anbieter) genutzt werden. So ist es insbesondere zum Testen/Entwickeln nicht notwendig einen eigenen Endpunkt zum Testen aufzusetzen.

```
local json = require "json"
local fetch = require "fetch"

fetch.post(
  "Deine URL von webhook.site hier!",
  {
    contentType = "application/json",
    data = json.encode(event)
  }
)
```

Wie prüfe ich auf leere / null Felder?

Metadaten die im Event-Payload (`event.payload`) übertragen werden, können auch leer sein, und werden dann als *null* dargestellt. JSONs "null" ist allerdings nicht identisch mit dem *nil* von Lua bzw. sind sogar *truthy*.

Zum Gegenprüfen muss daher *json.null* berücksichtigt werden.

```
local json = require "json"

local function isEmpty(string)
    return string == nil or string == '' or string == json.null
end

if isEmpty(event.payload.alarm.foreign_id) then
    print("Einsatznummer ist erforderlich, aber nicht vorhanden. " .. json.encode(event.alarm.id))
    return
end
```

Verwandte Artikel

- [Monitor-App – Download](#)
- [Anbindung X-Sense Rauchmelder an DIVERA 24/7](#)
- [Community-Projekte](#)
- [Tipps und Tricks](#)
- [Newsletter 2026](#)