

Vomatec FLOW

In diesem Anleitungskartikel zeigen wir Ihnen wie Sie Einsatzdaten automatisiert von DIVERA 24/7 an FLOW übergeben können.



Bevor Sie beginnen

Für die Einrichtung der Anbindung ist es von erheblichem Vorteil in FLOW bereits eine vollständige Einrichtung durchgeführt zu haben, damit bei der Zuweisung der Einheiten keine Probleme entstehen.

Diese Anleitung geht hier von einer PRO-Struktur aus, es ist jedoch auch auf ALARM-Versionen anwendbar, hier entfällt dann nur die Mehrfachauswahl bei den Einheiten und die Anlage des Systembenutzers ist dann auf der selben Ebene, wie es die Einheit ist.

Zur Nutzung der Schnittstelle werden die [Lua-Skripte](#) benötigt, diese müssen Sie vorher durch unseren Support freischalten lassen.
Dies können Sie ganz mit Angabe Ihrer Einheit per [Kontakt-Formular](#) erledigen.

Damit die Alarmer auch in FLOW ankommen müssen die jeweiligen Einsätze lokal gespeichert sein, also in der jeweils gewünschten Einheit, da die LUA Skripte aktuell nicht auf der zentralen Ebene angelegt werden können. Dies betrifft nur PRO-Strukturen.

Wenn die Fahrzeuge übergeben werden sollen ist es aktuell zwingend nötig, dass man für die Fahrzeuge in DIVERA und FLOW jeweils den gleichen Fremdschlüssel UND den gleichen Fahrzeugnamen vergibt, damit hier ohne Probleme die Fahrzeuge zugewiesen werden können.

In FLOW heißt das zugehörige Feld "Identifikator" und in DIVERA "Fremdschlüssel".
Die Namen für die Fahrzeuge sind die Felder "Fahrzeugbezeichnung" in FLOW und "Funkrufname" in DIVERA.

Eine gute Übersicht seitens FLOW finden sie hier: [VOMATEC Helpcenter](#)



Hinweis zu Einschränkungen

Aktuell kann die API von FLOW keine Gruppen/Schleifen übernehmen.

Übersicht der Schritte

Folgende Schritte sind für die Anbindung nötig

1. Grundlagen einrichten
2. Anlegen der Anbindung in FLOW um Personaldaten mit DIVERA 24/7 zu synchronisieren
3. Lua-Skript in DIVERA 24/7 ablegen um Einsatzdaten mit FLOW zu synchronisieren

1. Grundlagen einrichten

1.) Navigieren Sie im Webbrowser in DIVERA 24/7 zu den Schnittstellen mit folgendem Weg: "Verwaltung > Einstellungen > Schnittstellen" (Bild 1)

2.) Klicken Sie dort dann auf "+ System-Benutzer", um einen neuen System-Benutzer anzulegen. (Bild 1)

3.) Im Reiter "Allgemeines" geben Sie alle nötigen Daten für das Konto ein. (Bild 2)

4.) Im Reiter "Berechtigungen" stellen Sie dann alle Rechte ein, die der Nutzer benötigt. Hier empfehlen wir für die Verwaltungsrechte die Vorgehensweise: "So wenig wie möglich, so viel wie nötig". Zur reinen Funktion werden hier nur die Rechte für den Bereich Personal benötigt, die anderen Verwaltungsrechte können deaktiviert werden. Nach der Einstellung der gewünschten Rechte klicken Sie auf "speichern". (Bild 3)

5.) Nun können Sie in der Übersicht den API-Key sehen, diesen können Sie später von hier kopieren. (Bild 4)

2. Anlegen der Anbindung in FLOW um Personaldaten mit DIVERA 24/7 zu synchronisieren

1.) Klicken Sie als Admin oben rechts in FLOW auf das Zahnrad. (Bild 1, Punkt 1)

2.) Klappen Sie durch einen Klick auf "Globale Einstellungen" das Menü aus und klicken dann auf "Anbindungen". (Bild 1, Punkt 2)

3.) Nun können Sie im nächsten Menü auf "+ Erstellen" klicken. (Bild 1, Punkt 3)

4.) Nun öffnet sich ein Popup zum hinzufügen einer Anbindung (Bild 1, Punkt 4). Hier richten Sie als erstes die "Schnittstelle Einsatzdaten" ein und klicken auf Weiter. (Bild 1, Punkt 5). Anschließend bestätigen Sie den Namen in dem Sie auf "Erstellen" klicken. (Bild 1, Punkt 6)

5.) Nun werden Ihnen Daten angezeigt, diese sind sehr wichtig und werden nur einmal angezeigt, daher sollten Sie diese wie im Fenster angeben sichern. (Bild 1, Punkt 7)

6.) Nun Klicken Sie erneut auf "+Erstellen" um eine weitere Anbindung anzulegen, dieses mal die "DIVERA 24 /7" Anbindung und klicken dann auf "weiter". (Bild 1, Punkt 8)

7.) Im nun geöffneten Fenster klicken Sie den Schieberegler auf "ein", wenn Ihre Einheit nicht auf unseren Servern abgelegt ist, damit man eigene Daten eingeben kann (falls Sie z.B. SERVER-Kunde sind, zur Veranschaulichung wird dies so weitergeführt, auch wenn die Eintragung in dem Fall nicht benötigt wird). (Bild 1, Punkt 9)

8.) Gebe Sie nun bei Bedarf die Server-Adresse und den API-Key des System-Benutzers von Schritt 1.5 ein und klicken auf "erstellen". (Bild 2, Punkt 10)

9.) Nun Können Sie die gewünschten Adressaten auswählen, die mit DIVERA synchronisiert werden sollen und klicken dann auf "übernehmen". (Bild 2, Punkt 11)

10.) Sie können nun in der Tabelle die FLOW-Einheiten (links) mit den DIVERA-Einheiten (rechts) verbinden. Und klicken dann auf "speichern". (Bild 3, Punkt 12)

11.) Nun wird Ihnen folgende Tabelle angezeigt in der bei "Status" bei beiden "läuft" und beide Schieberegler auf "aktiv" stehen sollten. Ist dies nicht der Fall wie z.B. "Wird gestartet" muss einmal der jeweilige Schieberegler deaktiviert und dann wieder aktiviert werden. (Bild 3, Punkt 13)

Anbindungskonfiguration

Pflichtangaben

...

DIVERA 24/7

Eigene API-URL verwenden

API-URL*

https://app.divera247.com/

API-Key

Bleibt dieses Feld leer, wird der bisherige API-Key verwendet.

Organisationszuordnung

DIVERA 24/7 Einheit

LZ1

Löscheinheit Nord

LZ2

Löscheinheit Drausental

+ Hinzufügen

Auf der linken Seite sehen Sie die Organisationseinheiten aus FLOW, auf der rechten Seite die entsprechenden Einheiten aus DIVERA 24/7. Bitte wählen Sie aus, welche Organisationsknoten aus FLOW mit welcher Einheit aus DIVERA 24/7 synchronisiert werden soll.

Schließen

Abbrechen

Speichern

12

Bezeichnung

↑

Anbindung

Status

Letzter Fehler

Stand

Aktiv

DIVERA

DIVERA 24/7

Läuft

-

11.06.2026, 10:06

Aktiviert

Schnittstelle

Schnittstelle Einsatzdaten

Läuft

-

11.06.2026, 09:33

Aktiviert

13

3. Lua-Skript in DIVERA 24 /7 anlegen um Einsatzdaten mit FLOW zu synchronisieren

2.) Klicken Sie dort auf "+Vorlage" und geben dann in der Konfiguration einen gewünschten Namen ein und bei Bedarf eine kurze Beschreibung. Als Event wählen Sie "Erstellung" und "Aktualisierung". (Bild 2)

3.) Gehen Sie auf den Reiter "Eingabeparameter" und löschen Sie die bisherigen Zeichen in dem Feld. Fügen Sie nun folgenden Code in das Feld ein. (Bild 3)

```
[
  {
    "type": "text",
    "key": "clientId",
    "label": "Client ID",
    "description": "Die
Client-ID aus der
Anleitung"
  },
  {
    "type": "text",
    "key": "clientSecret",
    "label": "Client
Secret",
    "description": "Das
Client-Secret aus der
Anleitung"
  }
]
```

Lua-Script

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

local function isEmpty
(string)
    return string == nil or
string == '' or string ==
```

Vorführung • Feuerwehren Kreis Wupperstadt • Löschinheit Nord

Einheiten	Personal	Einstellungen	Konto
infokanal (Info)  Löschinheit Drausental (01)  Löschinheit Nord (02)  Vorgaben Personal Einstellungen Setup Meldungen Automatisierung Schnittstellen Konto	Benutzer Gruppen	Meldungen Einheitsübergreifende Monitore und Monitor- Benutzer Einheitsvorgaben  Automatisierung Schnittstellen Leitstelle	Support Kontakt- und Vertrags Guthaben und Verbrau Sponsoring Datenschutz

Schnittstellen für den Datenaustausch ?

API RICS ALARMSERVER SMS-/FAX-INBOUND DATENÜBERGABE SYSTEM-BENUTZER

Typ	Einstellungen
Lua-Scripts (Beta)	Vorlagen: 0/5, Aktive Instanzen: 0/5

Zurück

Vorlage: FLOW

Speichern

KONFIGURATION

EINGABEPARAMETER

SCRIPT

Formular Editor

Die Ausführung eines Scripts erfordert teilweise (sensitive) Eingabeparameter, bspw. API-Schlüssel. Hier können solche in JSON definiert werden. Diese werden dann als Felder in Script-Instanzen dargestellt, deren Werte an das Script bei der Ausführung übergeben werden. Mehr über die Formular-Konfiguration finden Sie in unserer FAQ.

1

Formular Vorschau

Dieses Formular dient nur als Vorschau. Hier eingegebenen Werte werden **nicht** gespeichert.

Client ID

Die Client-ID aus der Anleitung

Client Secret

Das Client-Secret aus der Anleitung

Lua-Script Vorlagen

Scripts erlauben die flexible Datenübertragung übertragende Datenformat individuell angepasst es das Datenformat von DIVERA 24/7 bereits unterstützt. Scripts werden in dem Lua 5.1 Dialekt Luau gealarmierungen ausgeführt.

Da Scripts nicht nutzerbezogen sind, sondern auf greifen auch keine gesetzeten Einstellungen für daher immer alle Informationen (z.B. zu einem Alarmierung).

Script-Vorlagen (0)

Bevor ein Script angelegt werden kann, muss z.B. eine Vorlage erstellt werden.

Vorlagen definieren das eigentliche Script, die Eingabeparameter und Ereignisse bei denen es ausgeführt werden kann. Aus einer Vorlage können mehrere Script-Instanzen erstellt werden.

Zurück

Vorlage: FLOW

CONFIGURATION

EINGABEPARAMETER

SCRIPT

Titel*

FLOW

Beschreibung

Übergabe der Einsatzdaten zu Vornamete

Unterstützte Events

Wählen Sie hier Ereignisse aus, bei denen dieses Script erstellt werden, können dann aus den hier ausgewählten Ereignissen finden Sie in [unserer FAQ](#).

Alarm	Warnung
☒ Erstellung (alarm.create) ?	☐ Änderung
☒ Aktualisierung (alarm.update) ?	☐ Erfüllung
☐ Benachrichtigung (alarm.notify) ?	☐ Nichterfüllung
☐ Rückmeldung (alarm.response) ?	☐ Rückmeldung
☐ Gelesen (alarm.read) ?	☐ Gelesen

Fahrzeugstatus/FMS

- ☐ Statusänderung (vehicle:fmsstatus-change)

Zurück

Vorlage: FLOW

KONFIGURATION

EINGABEPARAMETER

SCRIPTEN

Cheat Sheet

Das ausgeführte Event steht unter der Variable `event`. Die Eingabeparameter stehen unter der Variable `input`. Mit `print()` können Log-Ausgaben erzeugt werden. Weitere Beispiele und Details haben wir in unserer [Dokumentation](#).

Script-Editor

```

1 -- [[ In Script hier

```

Script Ausführen

Lua-Script Vorlagen

Scripts erlauben die flexible Datenübertragung übertragende Datenformat individuell angepasst es das Datenformat von DIVERA 24/7 bereits unterstützt. Scripts werden in dem Lua 5.1 Dialekt [Lua](#) geschrieben und Alarmierungen ausgeführt.

Da Scripts nicht nutzerbezogen sind, sondern global greifen auch keine gesetzten Einstellungen für den Nutzer daher immer alle Informationen (z.B. zu einem Alarm).

Script-Vorlagen (1)

Vorlagen definieren das eigentliche Script, die möglichen Eingabeparameter und Ereignisse, die es ausgeführt werden kann. Aus einer Vorlage werden dann mehrere Script-Instanzen erstellt werden können.

FLOW

Übergabe der Einsatzdaten zu Vomatics FLOW

Unterstützte Events: alarm.create, alarm.update, alarm.delete

✓ Vorlage bearbeiten

Instanzen

+

Neue Instanz

```

json.null
end

local response = fetch({
    url = "https://id.
vomatec.app/connect/token",
    method = "POST",
    headers = {
        ["Content-Type"] =
"application/x-www-form-
urlencoded;charset=UTF-8"
    },
    body =
"client_id=IHRE_ID&client_s
ecret=IHR_SECRET&grant_type
=client_credentials"
})

print("Alarm: " .. json.
encode(event.payload))

local decoded = json.decode
(response.body)
local token = decoded.
access_token

local einsatzmittelArray =
{}

if event.payload.vehicles
then
    for _, vehicle in
ipairs(event.payload.
vehicles) do
        local newVehicle =
{

fahrzeugBezeichnung =
tostring(vehicle.name)
        }

        if not isEmpty
(vehicle.foreignid) then
            newVehicle.
fremdsystemId = tostring
(vehicle.foreignid)
        else
            newVehicle.
fremdsystemId = tostring
(vehicle.name)
        end

        table.insert
(einsatzmittelArray,
newVehicle)
    end
end

local ts = event.payload.
ts_create
local beginnDatum = if not
isEmpty(ts) then string.sub
(ts, 1, 10) else ""
local beginnUhrzeit = if
not isEmpty(ts) then
string.sub(ts, 12, 19)
else ""

local
ermittelteEinsatznummer =
if not isEmpty(event.

```

Zurück
Instanz: FLOW
Speichern
Löschen

Name*

FLOW

☒ Ausführung aktiv

Aktive Events

Alarm

☒ Erstellung (alarm.create) ?
☒ Aktualisierung (alarm.update) ?

Konfiguration

Client ID

Die Client-ID aus der Anleitung

Client Secret

Das Client-Secret aus der Anleitung

Lua-Script Vorlagen
Vorlage importieren ?

Scripts erlauben die flexible Datenübertragung zu externen Systemen. Im Unterschied zu Webhooks kann hier das zu übertragende Datenformat individuell angepasst werden – das Drittsystem kann also angebunden werden, ohne dass es das Datenformat von DIVERA 24/7 bereits unterstützt.

Scripts werden in dem Lua 5.1 Dialekt Luau geschrieben und werden bei bestimmten Ereignissen z.B. neuen Alarmierungen ausgeführt.

Da Scripts nicht nutzerbezogen sind, sondern gemeinsam von der Verwaltung dieser Einheit konfiguriert werden, greifen auch keine gesetzten Einstellungen für die Berechtigung "Felder beschränken". In den Event-Payloads stehen daher immer alle Informationen (z.B. zu einem Einsatz) zur Verfügung.

Script-Vorlagen (1)

Vorlagen definieren das eigentliche Script, die möglichen Eingabeparameter und Ereignisse bei denen es ausgeführt werden kann. Aus einer Vorlage können dann mehrere Script-Instanzen erstellt werden.

+ Vorlage

FLOW

Übergabe der Einsatzdaten zu Vomatec FLOW
Unterstützte Events: alarm.create, alarm.update

☒ Vorlage bearbeiten

Instanzen

Übergabe zu FLOW (Aktiv)

+ Neue Instanz

Do., 11.06.
10:44

BMA Einle

Adresse: Vohw
Meldung: Geb
Melder: 24/7
Etage: 2
Raum: 06

Mit Priorität/S
Alarmierte Fal

nicht ve

☐ Einsatznummer
☐ 370258

Beginn ↓
11.06.2026, 10:4

```

payload.foreign_id) then
event.payload.foreign_id
else event.payload.id

local payloadEinsatzdaten
= {
  einsatznummer =
ermittelteEinsatznummer,
  stichwort = event.
payload.title,
  meldebild = event.
payload.text,
  einsatzleiter =
"DIVERA247",
  istBeendet = if (event.
payload.closed == "1")
then true else false,
  istAktiv = if (event.
payload.closed == "1")
then false else true,
  beginnDatum =
beginnDatum,
  beginnUhrzeit =
beginnUhrzeit,

  einsatzort = {
    adresszusatz = event.
payload.address,
    latitude = 0,
    longitude = 0
  },

  kurzbericht = event.
payload.report,
  einsatzmittel =
einsatzmittelArray
}

if not isEmpty(event.
payload.lat) and not
isEmpty(event.payload.lng)
then
  payloadEinsatzdaten.
einsatzort.latitude =
tonumber(event.payload.
lat) or 0
  payloadEinsatzdaten.
einsatzort.longitude =
tonumber(event.payload.
lng) or 0
end

print
("payloadEinsatzdaten: "
.. json.encode
(payloadEinsatzdaten))

local apiUrl =
`https://api.flow.arigon.
vomatec.app/ext/einsaetze/
{ermittelteEinsatznummer}?
api-version=1.0-extern`

local resultEinsatzdaten =
fetch({
  url = apiUrl,
  method = "PUT",
  headers = {
    ["Content-Type"] =
"application/json",

```

```
["Authorization"] =  
`Bearer {token}`  
},  
body = json.encode  
(payloadEinsatzdaten)  
})
```

5.) In der Instanz vergeben Sie einen Namen und geben in den jeweiligen Feldern die Daten ID und Secret, welche Sie sich bei Punkt 2.5 notiert haben, ein. (Bild 5)

6.) Zum Testen kann man dann einen Testeinsatz auslösen, in diesem Beispiel nur mit dem persönlichen Benutzer als einzigen Adressaten, dieser wird dann automatisch an FLOW übertragen. (Bild 6)

Verwandte Artikel

- [Vomatec FLOW](#)
- [Leitstellenanbindungen nach Bundesland](#)
- [Web-Schnittstelle \(API Dokumentation\)](#)
- [Einsatzbericht PDF](#)
- [Einsatz-Rückmeldungen konfigurieren](#)