

WIE SIE SWITCH MIT IHREN BESTEHENDEN SYSTEMEN VERBINDEN KÖNNEN

Was sind die Alternativen zu Hotfoldern?

Switch World Tour 2022

Powered by Enfocus and Impressed

HOTFOLDER

- Sehr ineffizientes Verfahren, um festzustellen, ob neue zu verarbeitende Daten vorliegen
- Polling in einem festgelegten Intervall, dessen Frequenz eigentlich immer „falsch“ ist
- Besser wäre ein Event-basierter Ansatz, wie er von modernen Betriebssystemen (zumindest für lokale Speichermedien) unterstützt wird
- Bei überwachten Netzwerk-Ordern kann ein beträchtlicher Netzwerk-Traffic entstehen
- Neue Jobs müssen aus dem Ordner bewegt werden, weil ansonsten ein sehr aufwändiger Tracking-Mechanismus zum Einsatz kommen muss

HOTFOLDER

- Ein hybrider Hotfolder-Ansatz wäre, dass in den überwachten Ordner nur eine XML oder JSON (seit Switch 2022 Spring) Datei gelegt wird, aus der u.a. hervorgeht unter welchem Pfad (lokal oder Netzwerk) die eigentliche zu verarbeitende Datei zu finden ist.
- Vorteile:
 - Die zu verarbeitenden Daten können in einer beliebig komplexen Ordner-Struktur abgelegt sein können
 - Die Original-Dateien müssen nicht zwangsweise bewegt werden
- So haben wir z.B. häufig bei Kunden gearbeitet, die einen Helios Server mit ImageServer Erweiterung einsetzen
- Die eigentliche Produktionsdatei kann dann mit dem „XML aufgreifen“ oder „Job einschleusen“ Flow-Elementen an geeigneter Stelle (also so spät wie möglich) in den Flow geholt werden.

E-MAIL MIT ANHANG

- Problematisch, weil
 - Von Menschen geschriebene E-Mails nie einen standardisierten Aufbau haben und es deshalb kritisch ist, diese maschinell fehlerfrei auszuwerten
 - Oftmals ist die max. Größe von Attachment durch den Mailserver beschränkt
 - Attachments können durch falsche Mailserver Einstellungen beschädigt werden
 - Häufig werden inline Logos und Bilder aus HTML-formatierten E-Mails auch mit eingeschleust
 - Gefahr von Spam E-Mails

DATENBANK-SUCHE

- Über die in Switch verfügbare ODBC (Open Database Connectivity) Schnittstelle kann Switch in regelmäßigen Intervallen oder wenn ein geeigneter Job als Auslöser (trigger) vorhanden ist, eine Datenbank nach neuen, verarbeitungsbereiten Aufträgen durchsuchen und deren Dateipfad oder sonstige Metadaten als Datenset in den Switch Flow einschleusen.
- Sobald der Job eingeschleust wurde, findet eine Aktualisierung des zum Job gehörenden Datenbank-Datensatzes statt, so dass dieser Job beim nächsten Intervall nicht mehr gefunden wird.
- Unser Switch Skript „ODBC Timed query“, erledigt beide Arbeitsschritte in einem Arbeitsschritt!

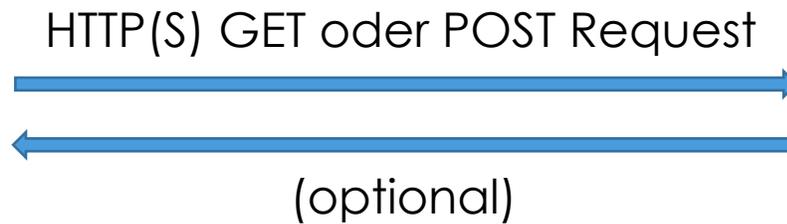
WEBHOOKS

- Was ist ein WebHook?
 - WebHooks ermöglichen es, einer Server-Software mitzuteilen, dass ein bestimmtes Ereignis eingetreten ist und eine Reaktion auf das Ereignis auszulösen.
 - Wenn eine Anwendung über ein eingetretenes Ereignis mittels WebHook informiert, müssen an dem Ereignis interessierte andere Anwendungen kein Polling betreiben, um von dem Ereignis Kenntnis zu erlangen. Das reduziert das Nachrichtenaufkommen zwischen den Anwendungen.
 - Technisch gesehen, wird eine HTTP-POST-Message an eine dafür vorbereitete URL gesendet, welche optional die angeforderten Daten zurückliefert oder einen externen Prozess auslöst.

WEBHOOKS



Sender (MIS, ERP,
WebShop, etc.)



Enfocus Switch Flow
mit WebHook Flow-
Element

WEBHOOKS

- Vorteile
 - Kein Polling – Push statt Pull
 - Kommunikation erfolgt über Standard HTTP(S) Protokoll
 - Keine (komplizierte) Transportschicht wie z.B. bei SOAP
 - Payload-Format (bei POST Requests) kann frei gewählt werden
 - Optional kann der Zugriff auf die WebHook URL via eines frei wählbaren Authentifizierungs-Verfahrens geschützt werden
- Nachteile
 - Zwischen beiden Systemen (Sender und Empfänger) muss eine Internet-Verbindung bestehen
 - Nur geeignet für kleine Payload-Größen (Enfocus begrenzt z.B. auf max. 1 Mb)
 - Was passiert, wenn der Request verpasst wurde? Switch cached requests wenn der Empfänger-Flow nicht gestartet ist

HTTP-ANFORDERUNG

- Seit Switch 2018 existiert das sog. **HTTP-Anforderung** Flow-Element
- Dies erlaubt dem Switch-Anwender beliebige HTTP(s) basierte POST, GET, PUT, DELETE und HEAD zu versenden
- Damit können beliebige Webservice, APIs, WebHooks oder sonstige via HTTP(S) Protokoll erreichbare Dienste bedient oder genutzt werden
- Im Falle von GET Requests, kann die Response des Dienstes als Job oder als Datenset in den Flow eingeschleust werden

REMOTE PROCESSING

- Seit Switch Version 2019 existiert ein Flow-Element namens „Remote-Prozess“, welches getriggert durch einen Job einen externen Prozess ansteuern kann
- Vergleichbar mit dem WebHook Ansatz, nur ist hier Enfocus Switch der sendende Client und der externe Prozess der Empfänger
- Außerdem ist der Aufbau der Payload die von Switch mit dem Request versendet wird, durch eine API standardisiert.
- Es existiert hierzu eigens eine umfangreiche Dokumentation der sog. „**Remote Processing REST API**“ und eine Node.js basierte **Referenz-Implementierung** einer Remote Processing App.

REMOTE PROCESSING



Enfocus Switch Flow
mit Remote-Prozess
Flow-Element

HTTP(S) POST Request



```
{  
  "token": "xxx...xxx",  
  "jobId": "xxxxx",  
  "outConnections": [  
    {  
      "name": "xxx",  
      "id": "xxx"  
    },  
    ...  
    {  
      "name": "xxx",  
      "id": "xxx"  
    }  
  ]  
}
```



Remote Prozess App

REMOTE PROCESSING



Enfocus Switch Flow
mit Remote-Prozess
Flow-Element

HTTP(S) GET Job Download

← http://<switch_ip>:51120/api/v1/job/:jobId

HTTP(S) GET List Datasets

← http://<switch_ip>:51120/api/v1/job/:id/metadata

Beliebige Form der Job-Bearbeitung oder -Erzeugung



Remote Prozess App

HTTP(S) POST Create Child

← http://<switch_ip>:51120/api/v1/job/:id/child
(Payload besitzt multipart/form-data Encoding)

HTTP(S) POST Finish Processing

← http://<switch_ip>:51120/api/v1/processingFinished



FRAGEN?

Danke für Ihre Aufmerksamkeit