

Kennisgeving van bron aan GegevensMagazijn

Een kennisgeving bericht van een bron (hier PIV4All) aan het GegevensMagazijn wordt verwerkt in een aantal series van stappen. De eerste serie stappen zorgt er voor dat de kennisgeving berichten van PIV4All naar DDS4All worden afgetapt zodat ze ook beschikbaar zijn voor het GegevensMagazijn.

1. Het StUF kennisgeving bericht van PIV4All wordt over HTTP aangeboden aan een endpoint dat is geconfigureerd in Mule.
2. Het bericht wordt in eerste instantie verwerkt door de WSDL-Proxy service. Als het hier een vraag betreft voor de WSDL van de service dan wordt de WSDL response van het DDS4All systeem terug gegeven als antwoord en is de verwerking afgerond. Als het geen WSDL vraag betreft dan worden de volgende stappen uitgevoerd.
3. De inhoud van het kennisgeving bericht wordt opgeslagen in de aftap bericht tabel van het GUC_QUEUE schema.
4. De kennisgeving wordt over HTTP aangeboden aan het endpoint voor de DDS4All server dat is geconfigureerd in Mule.
5. De DDS4All server verwerkt het bericht en bouwt als response een StUF bevestigingsbericht op.
6. De inhoud van dit response bericht worden opgeslagen in de aftap response tabel van het GUC_QUEUE schema.
7. Tenslotte wordt dit StUF response bericht over de HTTP verbinding teruggegeven aan de PIV4All server.

Na deze stappen is de verwerking wat betreft PIV4All en DDS4All afgerond. Het afgetapte bericht is nu ook beschikbaar voor verwerking door het GegevensMagazijn.

8. Een polling service leest het kennisgeving bericht uit de aftap bericht tabel van het GUC_QUEUE schema.
9. Via transformers worden een aantal propertjes aan het bericht toegevoegd zoals een file naam op basis van het unieke ID van het bericht (correlation ID).
10. Via een wiretap wordt een kopie van het bericht opgeslagen in een file in de backup directory.
11. De karakteristieken van het bericht worden opgeslagen in de bericht_in tabel van het GUC_LOG schema.
12. Via een wiretap wordt een kopie van het bericht ook beschikbaar gemaakt voor andere afnemers die eventueel interesse kunnen hebben in het

bericht.

13. Via een transformator wordt de naam van de bronhouder als message property toegevoegd aan deze kopie van het bericht.
14. Vervolgens wordt de kopie van het bericht op een persistent queue gezet voor verwerking door de afnemer filters (dat wordt verder uitgewerkt in een ander plaatje).
15. Voor de routing van het oorspronkelijke bericht naar verdere verwerking door het GegevensMagazijn wordt eerst bepaald of het een StUF 2.04 of een StUF 3.01 bericht betreft.

StUF 204 berichten worden verder verwerkt via de volgende serie stappen.

16. Het bericht wordt getransformeerd naar een formaat dat geschikt is voor de LoaderService: de SOAP envelop en de namespaces worden verwijderd en het wordt van versie 2.04 omgezet naar versie 3.00.
17. De Stuf300 LoaderService breekt het bericht op in afzonderlijke elementen zoals aangegeven in een configuratie file.
18. De afzonderlijke elementen worden opgeslagen in de tabellen van het STUF300_IN schema.
19. Vervolgens wordt een stored procedure geactiveerd die de gegevens uit de tabellen van het STUF300_IN schema verwerkt in de DataVault tabellen van het DVGM schema (het eigenlijke GegevensMagazijn).
20. Tenslotte wordt het verwerkte bericht bewaard in een processed directory.

StUF 301 berichten worden verwerkt via een vergelijkbare serie stappen.

21. Het bericht wordt getransformeerd naar een formaat dat geschikt is voor de LoaderService: de namespaces worden verwijderd.
22. De Stuf301 LoaderService breekt het bericht op in afzonderlijke elementen zoals aangegeven in een configuratie file.
23. De afzonderlijke elementen worden opgeslagen in de tabellen van het STUF301_IN schema.
24. Vervolgens wordt een stored procedure geactiveerd die de gegevens uit de tabellen van het STUF301_IN schema verwerkt in de DataVault tabellen van het DVGM schema (het eigenlijke GegevensMagazijn).
25. Tenslotte wordt het verwerkte bericht bewaard in een processed directory.

