

Functionele beschrijving Renvooiservice API

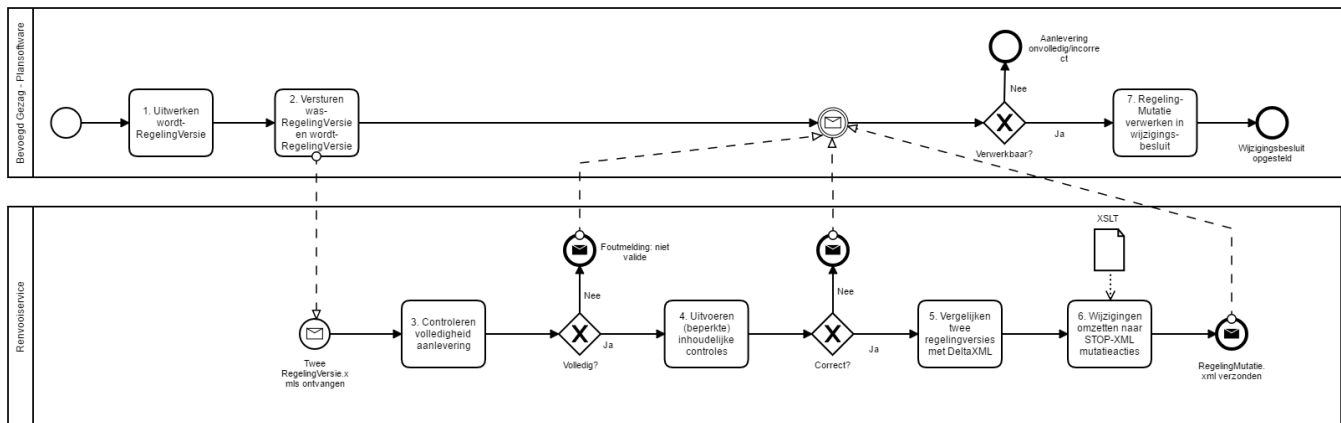
Introductie

STOP introduceert een nieuwe wijzigmethodiek: renvooi (zie de [STOP-standaard: presentatie van wijzigingsbesluiten](#)). Op basis van deze nieuwe wijzigmethodiek is de Renvooiservice ontwikkeld met als doel het bieden van een opzichzelfstaande service binnen het DSO-LV voor het bepalen van de renvooi tussen twee RegelingVersies. Een softwareleverancier kan er dus voor kiezen om de Renvooiservice in te zetten bij het bekendmakingsproces bij het bevoegd gezag, maar kan er ook voor kunnen kiezen om de renvooi in de eigen software te bepalen.

Het Renvooibepalingsproces (met de Renvooiservice)

Het aanspreken van de Renvooiservice maakt onderdeel uit van het Renvooibepalingsproces. Dit proces is hieronder uitgedrukt in een BPMN 2.0 diagram en kort beschreven.

BPMN



Procesbeschrijving

De trigger van het proces is het voornemen van het bevoegd gezag om wijzigingen door te voeren op een bestaande RegelingVersie.

1. Uitwerken wordt-RegelingVersie:

Het Bevoegd Gezag werkt in haar plansoftware een nieuwe versie van een Regeling uit (de "wordt-RegelingVersie").

2. Versturen was-RegelingVersie en wordt-RegelingVersie:

Op een bepaald moment wil het Bevoegd Gezag de wijzigingen tussen twee RegelingVersies vastleggen in een wijzigingsbesluit. Dit kan zowel een ontwerp als een definitief besluit zijn. Op dat moment kan de Renvooiservice gebruikt worden om de RegelingMutatie te bepalen. Hiervoor worden de was-RegelingVersie en de wordt-RegelingVersie aangeleverd.

3. Controleren volledigheid aanlevering

De Renvooiservice voert een aantal controles uit op de aangeleverde RegelingVersies. De was-RegelingVersie en de wordt-RegelingVersie moeten voldoen aan de relevante onderdelen van het lvbb-stop-aanlevering schema en de bijbehorende schematronregels.

Daarnaast zijn er controles specifiek voor de Renvooiservice:

- Het element [data:FRBRWork](#) dient aanwezig te zijn en in beide versies een identieke AKN-identificatie te bevatten;
- Het element [data:FRBRExpression](#) dient aanwezig te zijn en in beide versies een verschillende AKN-identificatie te bevatten.

Indien de aangeleverde bestanden niet voldoen aan de controles dan voert de Renvooiservice geen vergelijking uit.

4. Uitvoeren inhoudelijke controles

Ook voert de Renvooiservice een aantal voorbereidingsstappen uit om de vergelijking met DeltaXML (zie volgende stap) te optimaliseren:

• Artikeltransitie:

In deze stap controleert de Renvooiservice voor de was-RegelingVersie en de wordt-RegelingVersie de inhoud van artikelen met de inhoud van leden (en vice versa) om te bepalen of content hergebruikt is. Zo kan de Renvooiservice fijnmaziger verschillen in content detecteren.

• Begrippen:

Wanneer er begrippenlijsten met 10 of meer begrippen zijn opgenomen in de was-RegelingVersie en de wordt-RegelingVersie dan vergelijkt de Renvooiservice per begrip het wld en de [tekst:Term](#) met elkaar. Als dit in 50% of meer van de gevallen hetzelfde is dan wordt DeltaXML geïnstrueerd om het wld te gebruiken om renvooi te bepalen. Op deze manier wordt de renvooi bepaling niet gehinderd door eventuele veranderingen in de volgorde van begrippen.

- **Verplaatsingen:**

Om verplaatsingen sneller en beter te kunnen detecteren slaat de Renvooiservice voor de was-RegelingVersie en de wordt-RegelingVersie de inhoud van:

- Artikelen
- Divisieteksten
- Leden
- en Koppen van structuurelementen

"plat geslagen" in een tijdelijke container op.

- **Functies:**

Wanneer er functies van [politieke ambtsdragers](#) (bijvoorbeeld een minister zoals in onderstaand voorbeeld) zijn opgenomen in de was-RegelingVersie en de wordt-RegelingVersie dan maakt de Renvooiservice daar één sequentie van letters van. Op deze manier wordt bij het bepalen van renvooi eventuele wijzigingen niet individueel maar als wijziging van de functie in zijn geheel behandeld.

De Minister van Binnenlandse Zaken en Koninkrijksrelaties,
Gelet op artikel 19, vijfde lid, van de Bekendmakingswet en de artikelen 2.1 en 3.8 van het Besluit elektronische publicaties;
Besluit:

- **Tabel:**

Als instructie voor de Renvooiservice kunnen softwareleveranciers in een tabel het attribuut `@renvooi:colnameAsRowIdentifier` (met als waarde de betreffende kolom naam) toevoegen aan `tekst:tgroup` om één van de kolommen van de tabel te gebruiken als "RowIdentifier." Hierdoor kan de Renvooiservice beter de volgorde van rijen binnen de tabel bepalen.

5. Vergelijken twee RegelingVersies met DeltaXML:

De Renvooiservice maakt gebruik van de XML-vergelijkingssoftware DeltaXML om tot een vergelijking te komen tussen de twee RegelingVersies. De Renvooiservice gaat hierbij uit van [de STOP-regels voor renvooibepaling](#) om de vergelijking door DeltaXML aan te sturen.

6. Wijzigingen omzetten naar STOP-XML mutatieacties:

De Renvooiservice zet de output van DeltaXML om naar een STOP-XML (imop-1.3.0) RegelingMutatie conform [de STOP-regels voor renvooibepaling](#). Alleen verschillen in de elementen die op [de lijst mutatie-eenheden volgens STOP 1.3.0](#) staan, worden door de Renvooiservice doorgegeven (bv. een wijziging in het RegelingOpschrift wordt niet doorgegeven omdat RegelingOpschrift volgens STOP 1.3.0 geen mutatie-eenheid is). De Renvooiservice controleert de gegenereerde RegelingMutatie tegen het imop-tekst schema, de bijbehorende [schematronregels](#) en de [tekstmutatie schematronregels](#). Alleen indien de gegenereerde RegelingMutatie voldoet, dan wordt deze aan het Bevoegd Gezag gestuurd.

Om onterechte foutcodes bij de verwerking van de gegenereerde RegelingMutatie in de LVBB te voorkomen is er momenteel een workaround in de Renvooiservice geïmplementeerd voor verplaatsingen. Hierbij gebruikt de Renvooiservice de gemeenschappelijke ouder voor de mutatieactie [tekst:Vervang](#) in plaats van het verplaatste element zelf.

7. RegelingMutatie verwerken in wijzigingsbesluit:

het Bevoegd Gezag neemt de RegelingMutatie op in een STOP-wijzigingsbesluit conform de eisen van het Bronhouderkoppelvlakschema. Het STOP-besluit kan zowel een definitief besluit als een ontwerpbesluit betreffen.

Opmerkingen

- Het Bevoegd Gezag is verantwoordelijk voor de aanlevering van lvba:RegelingVersies die voldoen aan het [lvba-1.2.0 schema](#).
- Het Bevoegd Gezag is verantwoordelijk voor adequaat regelingenversiebeheer. Dit betekent onder andere dat bij de aanlevering van een wijzigingsbesluit de was-RegelingVersie die gewijzigd wordt reeds bij de LVBB bekend dient te zijn.
- De Renvooiservice voert schema-, schematron- en eigen controles uit op de aangeleverde RegelingVersies en de gegenereerde RegelingMutatie. De LVBB voert meer controles uit (zie voorbeeld in voorgaande punt), daardoor is het mogelijk dat een volgens de Renvooiservice correcte RegelingMutatie toch door de LVBB afgekeurd wordt.

Kort overzicht Renvooiservice API

De Renvooiservice API is een opzichzelfstaande service (geen bronhouderkoppelvlakservice).

Om gebruik te kunnen maken van de Renvooiservice is een API-key benodigd, deze kan worden aangevraagd via renvooiservice@koop.overheid.nl. De API-key dient als header mee gestuurd te worden: key: x-api-key met de juiste waarde van de verkregen API-key.

Base Url's:

- Oefenomgeving (ETO): <https://renvooiservice-eto.overheid.nl/>
- Productieomgeving: <https://renvooiservice.overheid.nl/>

Endpoints:

- /
De openapi.json van deze api
- /regelingmutatie-maak
Maken RegelingMutatie xml uitgaande van twee RegelingVersie xml bestanden:
 - te gebruiken Method: POST
 - argumenten:
 - scenario: scenario waarmee renvooi bepaald moet worden (vooruitlopend op toekomstige nieuwe renvooibepalingen);
 - bhkv-schema-versie: BHKV versie waar Renvooiservice gebruik van moet maken (vooruitlopend op concurrent versioning);
 - return: reponse als xml of zip (in zip kunnen zowel de RegelingMutatie als de Validatiemeldingen verpakt zitten)
 - validate: al dan niet in en output tegen schema(tron) valideren. Er wordt daarnaast sowieso altijd een aantal minimale checks gedaan die voorwaardelijk zijn voor een juiste verwerking van de renvooiservice.
 - docA: was-RegelingVersie;
 - docB: wordt-RegelingVersie;
 - de RegelingVersie bestanden moeten voldoen aan [het lvbba schema 1.2.0](#).
 - de gegenereerde RegelingMutatie voldoet aan [het STOP schema 1.3.0](#).
- /documentatie
PDF met de functionele beschrijving van de Renvooiservice API.

Renvooiservice API controleert de aangeleverde bestanden (de was- en de wordt-versie) en het resultaatbestand (RegelingMutatie bestand) tegen resp:

- de relevante onderdelen van het lvbb-stop-aanlevering schema en de bijbehorende schematronregels (was- en wordt-versie: RegelingVersie);
- het imop-tekst schema, de bijbehorende schematronregels en de tekstmutatie schematronregels (RegelingMutatie).
- de Renvooiservice geeft bij return is xml alleen de errors uit de schematrons terug (sluit aan bij de LVBB).
- Daarnaast kent de Renvooiservice een aantal (fout)meldingen die specifiek voor de Renvooiservice zijn (bv. "De data:FRBRWork dient aanwezig te zijn en in beide versies identiek"). Voor de manier waarop de foutmeldingen door de Renvooiservice aan het aanroepende systeem worden terug gegeven, is aangesloten bij de wijze waarop dit in de LVBB plaats vindt.

De response van de Renvooiservice is één van de volgende opties:

- Als return xml:
 - RegelingMutatie.xml ([volgens STOP 1.3.0](#))
 - Meldingen.xml: de applicatie onderscheidt meerdere (fout)meldingen (zie OpenAPI specificatie).
- Als return zip: een zip bestand met daarin mogelijk zowel de RegelingMutatie als de Validatiemeldingen verpakt.