

Tussenresultaten Pilot RSGB-bevragingen nieuwe stijl

Op weg naar een nieuwe aanpak voor standaardisatie in
het gemeentelijk domein

Agenda

Ochtend

- Resultaten, leerpunten en discussiepunten (Cathy Dingemanse, Johan Boer)
- Presentatie REST-JSON API (Jeroen Prins)
- Beoordeling (Liam Blyth, SIG)

Middag

- Criteria waar een goede standaard aan moet voldoen (Niels van der Zwan, SIG)
- Veranderagenda (Theo Peters, Henri Korver, Johan Boer)

Projectdoelstelling

- Samenwerking van KING met Gemeente Den Haag
- Ontwikkelen en realiseren van een koppelvlak in twee varianten.
 - XML/SOAP
 - REST/JSON
- Uitwisseling van basisgegevens op basis van RSGB 2.01.
- Maximale bijdrage aan doelstellingen voor standaardisatie:
 - interoperabiliteit
 - kostenreductie
 - bevorderen van marktwerking
 - bevorderen van innovatie.

De koppelvlakstandaard worden aangeboden aan de regiegroep ter vaststelling.

Uitgangspunten

- RSGB 2.01
- Functionele koppelvlakspecificatie RSGB bevragingenservices v0.10.
- Moet werken met gangbare tooling zoals codegeneratie en serialisatie.
- Ontwikkelen van de koppelvlakstandaard vanuit implementatie perspectief. De baten van standaardisatie moeten in de gemeentelijke omgeving worden geoogst!

Bijstelling functionele uitgangspunten

Koppelvlakspecificaties RSGB bevragingenservices v0.10

- Soms niet correct (bijvoorbeeld het antwoordbericht voor historische bewoning).
- Onvolledig: In de StUF onderlaag gedefinieerde functionaliteit is niet expliciet gemaakt
- Soms ambigu
- Deze bevindingen worden aangeboden aan de werkgroep RSGB-bevragingen.

Wat hebben we gedaan?

- Gestart vanuit de concept berichten van RSGB-bevragingen op basis van StUF-BG 3.10.
- Scope voor het eerste deel beperkt tot berichten rond de Persoon.
- Complexiteit van de berichten gereduceerd om doelstellingen te bereiken.
- Na twee iteratie-slagen gekozen om vanuit implementatieperspectief nieuwe berichten te definiëren.
- Gewenste functionaliteit aangescherpt n.a.v. Agile werkwijze en nieuwe inzichten.
 - RSGB: Scope beperkt tot ingeschreven personen
 - Andere natuurlijke personen zijn niet te zoeken/raadpleegbaar in basisregistraties.
 - RSGB bevragingen document
 - Specs RSGB bevragingen zijn aangescherpt voor zoeken persoon (6x) en 1x raadplegen.

Wat hebben we gedaan (vervolg)?

- “Impliciete” functionele eisen en wensen die in de onderlaag van StUF geïmplementeerd zijn expliciet gemaakt. Deze zijn beoordeeld op functionaliteit en wenselijkheid.
- Hergebruik van de StUF-BG basistentiteiten losgelaten.
- Opleveren XML SOAP interface en JSON interface op basis van nieuwe schema's van RSGB bevestigingen:
 - RBS WSDL en .XSD
 - RBS REST JSON API +
 - .NET Consumer testcode API
 - .NET Consumer testcode XML SOAP interface (op basis van code generatie/ serialisatie)
 - .NET Provider testcode (op basis van code generatie / serialisatie)
- Alle (tussentijdse) opleveringen zijn beoordeeld door SIG. De beoordelingen en zijn besproken en de conclusies verwerkt in de volgende oplevering.

Leerpunt: ondubbelzinnige specificatie

Voorbeeld: zoeken op geslachtsnaam

- Geslachtsnaam met diakrieten
 - geslachtsnaam=Stöcker.
Alleen personen met geslachtsnaam Stöcker in zoekresultaat
 - geslachtsnaam=Stocker.
Bevat personen met geslachtsnaam Stocker en ook Stöcker in zoekresultaat
- Geslachtsnaam met wildcards
 - geslachtsnaam=Groen%.
Alle personen waarvan geslachtsnaam begint met groen
 - geslachtsnaam=%Groen.
Alle personen waarvan geslachtsnaam eindigt op groen
 - geslachtsnaam=%Groen%.
Alle personen met 'groen' in geslachtsnaam

Functionaliteit uit onderlaag is niet expliciet gemaakt in de specificatie

- Metagegeven Strijdigheid/nietigheid
 - Komt alleen voor bij formele historie
 - Formele historie is buiten scope voor RSGB bevragingen
 - Conclusie: maakt geen onderdeel uit van dit koppelvlak
- Metagegeven In onderzoek
 - Alle eigenschappen en relaties van een persoon kunnen onderwerp zijn van een onderzoek, m.u.v. van kiesrecht en reisdocument
 - Toegevoegd aan functionele specificatie

Leerpunt 1: koppelen kan sneller en eenvoudiger

- Met deze koppelvlakstandaard is het mogelijk om:
 - snel en eenvoudig
 - met beperkte kennis van het RSGB
 - met beperkte documentatie
 - zonder kennis van StUF-BGeen koppeling te maken.

Leerpunt 2: platslaan is prima!

- Noodzakelijk
- Positief effect op de bruikbaarheid
- Herleidbaarheid naar RSGB is een belangrijk criterium
- Leidt tot simpele mappings naar het RSGB
- Voorbeeld: adres (samenvoegen van objecten die onderdeel zijn van een adres)

Leerpunt 3: valideer met gangbare tooling tijdens het opstellen van de koppelspecificatie

Vermijden van constructies die gebruik tooling belemmeren

- Gebruik geen choices
 - .NET kent geen choice constructie. In de gegenereerde code verlies je hierdoor strong typing.
- Gebruik geen XML-attributen
 - Die worden vertaald naar kenmerken, gebruik is niet nodig.
- Gebruik geen StUF:NoValue
 - Bestaande webservice frameworks kennen StUF:NoValue niet, waardoor een verplicht element zonder waarde altijd leidt tot serialisatie problemen.
- Gebruik geen namen als "Object" en "Tag" als elementnaam
 - Conflicten met code generatoren

Leerpunt 4: herkenbaarheid is verbeterd door gebruik natuurlijke taal

Vraagbericht:

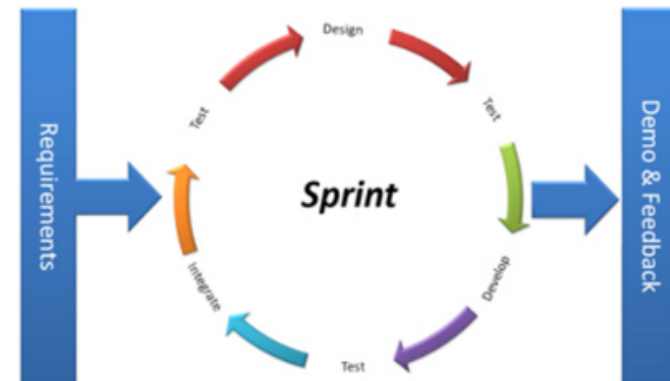
- Niet: NPS-vraag-naam-Di02
- Maar: ZoekIngeschrevenPersoonOpGeslachtsnaam

Bij platslaan:

- Niet: Inp.bsn of
verblijfsadres.gor.openbareRuimteNaam
- Maar: Burgerservicenummer of OpenbareRuimteNaam

Leerpunt 5: volgorde ontwerp koppelvlakstandaard

1. Functionele specificatie
2. Specificatie objectmodel conform en ten dienste van functionaliteit
3. Ontwerp REST en SOAP API
4. Code generatie



Leerpunt 6: Er hoeft geen relatie te zijn tussen het zoekfilter en het te vinden kenmerk

- De definitie van een zoekveld wordt niet meer afgeleid van het simpletype waarop dat zoekveld is gebaseerd. Het zoekveld geslachtsnaam-wildcard wordt dus een simpletype met een eigen definitie :
- ```
<simpleType name="Geslachtsnaam-wildcard">
 <annotation>
 <documentation>Met behulp van de % teken wordt
aangegeven dat een wildcard search moet worden
uitgevoerd</documentation>
 </annotation>
 <restriction base="string">
 <pattern value="[%]{0,1}[a-zA-z]{2,200}[%]{0,1}" />
 </restriction>
</simpleType>
```

## Het filter geslachtsnaam

Zoekfilter "geslachtsnaam" IS GEEN geslachtsnaam

- Kan % teken bevatten
  - Moet minimaal 2 karakters lang zijn (excl % teken)
  - Mag alleen a-z karakters en diakrieten bevatten
- 
- Waarom moeten deze simple types een relatie hebben?
  - Bij een inheritance relatie overerft het kind (filter) de restricties van de ouder (geslachtsnaam)
  - Dit is geen functionele eis maar een beperking bij het realiseren van functionele eisen die wel zijn gedefinieerd

## Het kenmerk geslachtsnaam

- Bevat geen % teken
- Mag max 200 karakters lang zijn
- Mag alle karakters bevatten



## Leerpunt 7 : Ter hergebruiken functionaliteit in een generiek document beschrijven

- Herbruikbare functionaliteit in een generieke voorziening vastleggen en in alle koppelvlakken op éénduidige wijze implementeren.
- Noot: Benoem hergebruikte functionaliteit ook in een koppelvlak. (Op zijn minst een verwijzing, indien gewenst een aanscherping.)
- Klein discussiepunt: wordt reeds bekende generieke functionaliteit op voorhand in een generiek document gevat of vatten we dat pas in een “Best-practice”
- Voorbeelden: “inOnderzoek”, “StrijdigheidNietigheid”, Sortering van de objecten in het antwoordbericht.

# Discussiepunt: Wanneer Platslaan?

## Standpunt A

- A-priori in een Uitwisselings Gegevens Model (UGM) als basis voor alle koppelvlakken. (Basis-entiteiten)

## Standpunt B

- Alleen op het moment dat het nodig is voor de functionaliteit van een koppelvlak.
- Altijd in dienst van of ondergeschikt aan de functionaliteit.
- Bouw zo een library op voor hergebruik.