

## Handleiding XML-vraag

Datum	1 september 2017
Status	Definitief
Versie	3.2

## Inhoudsopgave

Handleiding XML-vraag	1
Inhoudsopgave	2
Inleiding	3
1 Standaard XML-functionaliteit in SmartDocuments	4
2 Standaard functionaliteit XML-vraag	6
2.1 Vraageigenschappen XML-vraag	7
2.1.1 Tabblad Algemeen	7
2.1.2 Tabblad XML	9
2.1.1.1 Grafisch	10
2.1.1.2 Attributen	13
3 XPath	14
3.1 Wat is XPath?	14
3.2 Waarom XPath?	14
4 XPath-expressies	16
Contactgegevens	20
Disclaimer	20

## Inleiding

SmartDocuments is bij uitstek geschikt om gekoppeld te worden aan andere applicaties. Deze koppelingen werken vaak op basis van XML. Dit document beschrijft de werking van het SmartDocuments vraagtype XML-vraag. De XML-vraag is een standaard SmartDocuments vraagtype voor het werken met repeterende xml-structuren. Met de XML-vraag heeft de SmartDocuments gebruiker een standaard oplossing waarmee deze gegevens op eenvoudige wijze in een sjabloon verwerkt kunnen worden.

## 1 Standaard XML-functionaliteit in SmartDocuments

Het komt vaak voor dat een XML-bestand bepaalde repeterende structuren bevat.  
Een voorbeeld:

```
<?xml version="1.0" encoding="UTF-8"?>
- <sdxml>
  + <klant>
  + <order>
  - <product>
    <Naam>Ultra - USB flash drive - 16 GB - USB 3.0</Naam>
    <merk>SanDisk</merk>
    <stuksprijs>8,11</stuksprijs>
    <aantalbesteld>2</aantalbesteld>
    <totaalprijs>16,22</totaalprijs>
    <BTW>3,41</BTW>
  </product>
  - <product>
    <Naam>Ultra Micro SDHC kaart 32 GB</Naam>
    <merk>SanDisk</merk>
    <stuksprijs>15,95</stuksprijs>
    <aantalbesteld>3</aantalbesteld>
    <totaalprijs>47,85</totaalprijs>
    <BTW>10,05</BTW>
  </product>
  - <product>
    <Naam>Bureaustoel NPR1813, zwart</Naam>
    <merk>ErgoStar</merk>
    <stuksprijs>458,59</stuksprijs>
    <aantalbesteld>2</aantalbesteld>
    <totaalprijs>917,18</totaalprijs>
    <BTW>192,61</BTW>
  </product>
  + <SmartDocument>
</sdxml>
```

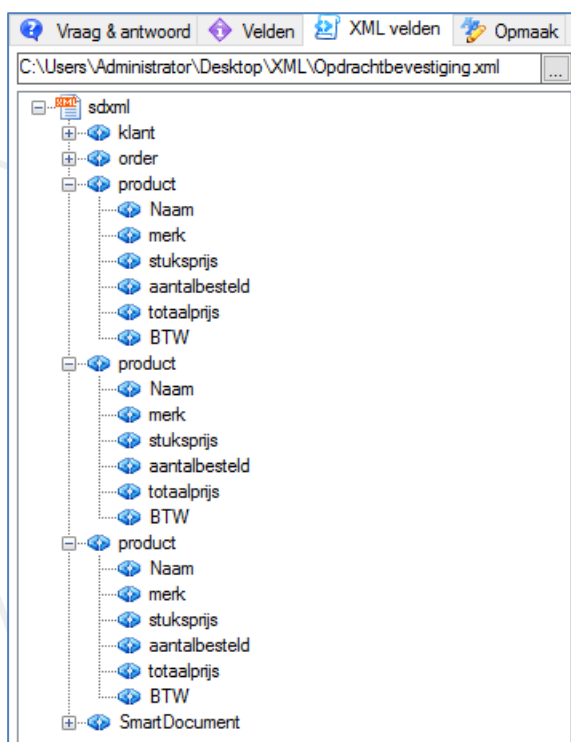
*Afbeelding 1: productinformatie in een XML-bestand.*

Het bovenstaande voorbeeld is afkomstig van een webwinkel. Zij gebruiken onder andere dit bestand om hun opdrachtbevestiging te genereren. Zoals op afbeelding 1 te zien is, worden de eigenschappen van de producten herhaald. Voor ieder product worden de naam, het merk, de stuksprijs, het aantal bestelde artikelen, de totaalprijs en het btw-bedrag getoond.

Het is de bedoeling dat er met behulp van de bovenstaande gegevens door SmartDocuments een opdrachtbevestiging gegenereerd wordt.




SmartDocuments biedt de gebruiker onder andere de mogelijkheid om XML-tags in een sjabloon te plaatsen.

In SmartEditor kan op het tabblad XML een XML-bestand in worden gelezen:



Afbeelding 2: het XML-bestand ingelezen in SmartEditor

Zoals op afbeelding 3 getoond wordt, kunnen de tags eenvoudig in het sjabloon gesleept worden:

Naam	Merk	Prijs per stuk incl. BTW	Aantal besteld	Totaalprijs incl. BTW	BTW 21 %
[  Naam ]	[  merk ]	[  stuksprijs ]			

Afbeelding 3: XML-tags in het sjabloon

Op deze manier kunnen alle tags van de 3 bestelde producten in het XML-bestand gesleept worden.

Het probleem is echter dat SmartDocuments geen onderscheid kan maken tussen XML-tags die dezelfde naam hebben en in hetzelfde bereik (in het XML-bestand onder de tag 'product') staan.

Ieder besteld product heeft in het XML-bestand dezelfde tags, alleen met een andere waarde. In de tabel staat in de 2<sup>e</sup> cel in de kolom 'Naam' de tag '<naam>'. SmartDocuments zoekt bij het opbouwen van het document in het XML-bestand naar de tag '<naam>'. SmartDocuments zoekt in de XML geen tags met dezelfde naam kan onderscheiden, zal de waarde van de 1<sup>e</sup> tag met die naam in het document worden geplaatst worden. In de 1<sup>e</sup> rij van de tabel zullen dus netjes de gegevens van het 1<sup>e</sup> bestelde product in het XML-bestand geplaatst worden. Maar in de 2<sup>e</sup> rij van de tabel zullen weer diezelfde gegevens geplaatst worden, omdat deze tags dezelfde naam hebben.

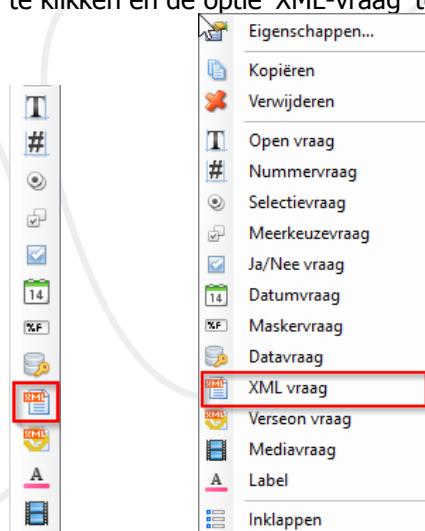
Verder is vooraf veelal niet bekend hoe vaak een repeterend element voorkomt.

## 2 Standaard functionaliteit XML-vraag

Zoals uit het voorgaande voorbeeld blijkt is de standaard XML-functionaliteit van SmartDocuments niet afdoende voor het werken met XML-bestanden met repeterende structuren. Dit hoofdstuk beschrijft hoe de XML-vraag daarvoor kan worden ingezet. Er zijn 2 grote verschillen met t.o.v. de andere vraagtypen van SmartDocuments:

- de XML-vraag is niet zichtbaar in het vragenscherf.
- SmartDocuments zal moeten worden doorgestart met een XML-bestand om de XML-vraag te activeren.

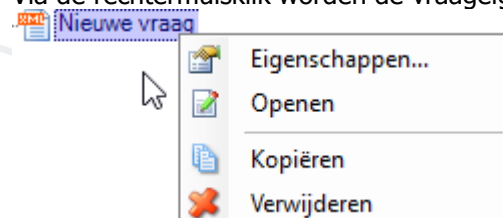
Allereerst zal het sjabloon geopend moeten worden. Vervolgens zal er een XML-vraag toe moeten worden gevoegd door te klikken op de knop 'XML-vraag', of door met de rechter muisknop op het vragenscherf te klikken en de optie 'XML-vraag' te selecteren.



Afbeelding 4 en 5: de knop 'XML-vraag' en de optie 'XML-vraag'

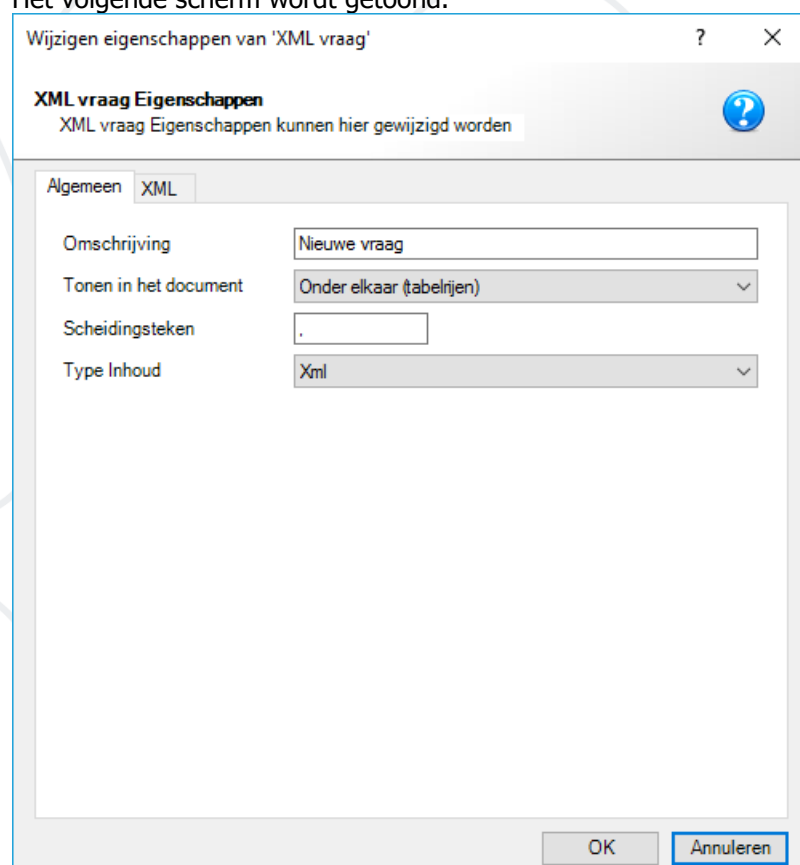
## 2.1 Vraageigenschappen XML-vraag

Via de rechtermuisklik worden de vraageigenschappen geselecteerd.



Afbeelding 6: Context-menu van de XML-vraag

Het volgende scherm wordt getoond.



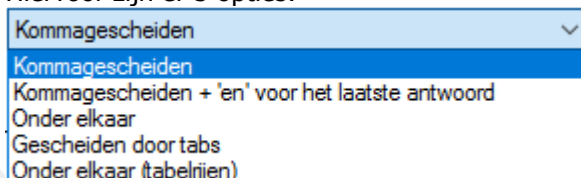
Afbeelding 7: de eigenschappen van de XML-vraag – Tabblad Algemeen

### 2.1.1 Tabblad Algemeen

De volgende items kunnen hier worden geconfigureerd.

- **Omschrijving:**  
Voer een duidelijke omschrijving van de vraag in.
- **Tonen in het document**  
Op welke wijze moeten de antwoorden in het opgebouwde document worden geplaatst.

Hiervoor zijn er 5 opties:



- **Komma gescheiden;** a, b, c, d
- **Komma gescheiden + 'en' voor het laatste antwoord;** a, b, c en d
- **Onder elkaar**
- **Gescheiden door tabs**
- **Onder elkaar in tabelrijen**

Bij de laatste optie is het noodzakelijk dat op de antwoordbladen een tabelrij wordt geplaatst, van gelijke opmaak. SmartDocuments zorgt er dan voor dat alle gekozen rijen netjes onder elkaar uitgelijnd in het opgebouwde document worden geplaatst.

#### ○ **Scheidingsteken**

In sommige gevallen, bijvoorbeeld bij een opsomming van waardes die al een komma bevatten, voldoet de komma niet als scheidingsteken.

In dat geval kan de sjabloonbeheerder een afwijkend scheidingsteken ingeven, bijvoorbeeld een ;  
De opsomming wordt dan bijvoorbeeld:

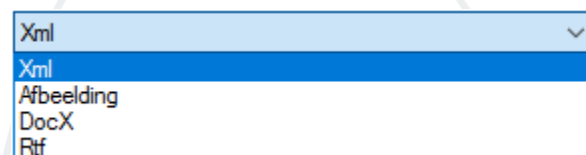
*De volgende personen zijn in juli 2017 in het geboorteregister opgenomen:*

**Joustra, Barend Peter; Hiemstra, Willemijn; Nanninga, Wilma en Ozdemir, Mohammed.**

#### ○ **Type inhoud**

De inhoud van de XML-node kan verschillend zijn van inhoud.

We onderscheiden 4 types:



- **XML**  
De inhoud van de node is numeriek of tekstueel
- **Afbeelding**  
Om een afbeelding te kunnen opslaan in een database of te kunnen verzenden via een XML moet de afbeelding worden gecodeerd/versleuteld. Je kunt een afbeelding bijvoorbeeld embedden in een XML door gebruik te maken van Base64 encoding.



Om ervoor te zorgen dat de afbeelding ook daadwerkelijk als Afbeelding wordt verwerkt in het op te bouwen document moet hier als inhoudstype voor '**Afbeelding**' worden gekozen.

- **DocX**

Het is ook mogelijk dat een XML-node een geheel document bevat.

Om de inhoud van dit document correct te verwerken in het op te bouwen document moet hier de optie '**DocX**' worden geselecteerd.

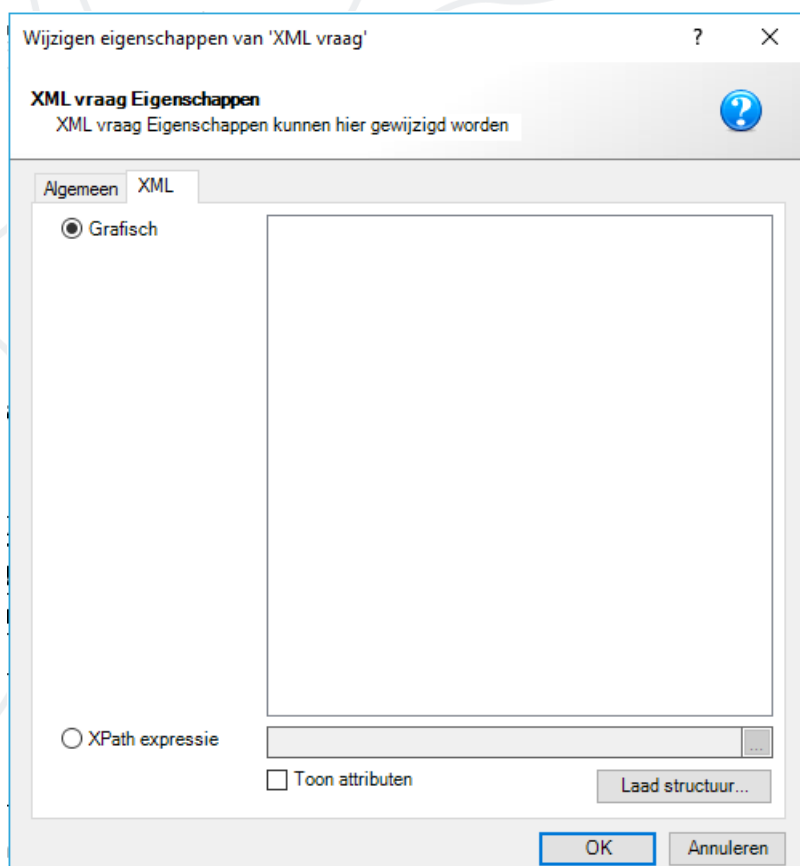
***Wist u dat?***

Sinds Microsoft Office 2007 wordt het bestandsformaat DocX gebruikt. Je kunt zien dat een DocX feitelijk zelf ook een verzameling van XML-bestanden is door de bestandsextensie DocX te wijzigen in .zip en vervolgens uit te pakken.

- **RTF**

Rich Tekst Format (RTF) is een bestandsindeling bedoeld om teksverwerkingbestanden platformonafhankelijk te kunnen uitwisselen. Op het moment dat de inhoud van een node uit rtf-tekst bestaat, dan dient deze optie te worden gebruikt.

## 2.1.2 Tabblad XML



Afbeelding 8: de eigenschappen van de XML-vraag – Tabblad XML

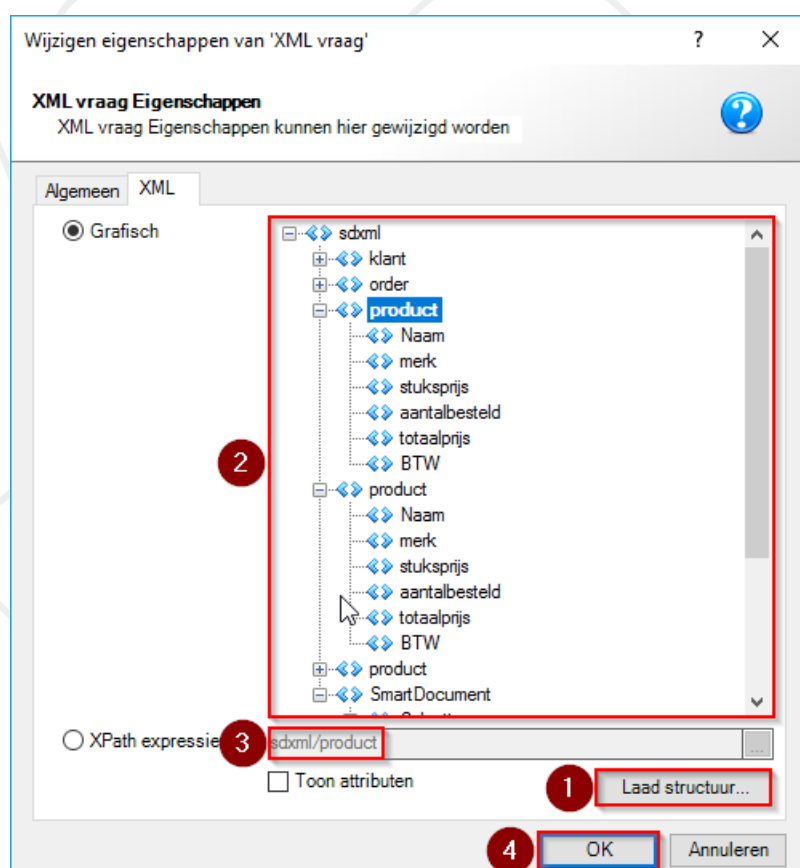
### 2.1.1.1 Grafisch

De optie 'Grafisch' geeft een grafische weergave van de structuur van het XML-bestand.

Deze optie moet worden gebruikt om repeterende XML-nodes correct te verwerken in het op te bouwen.

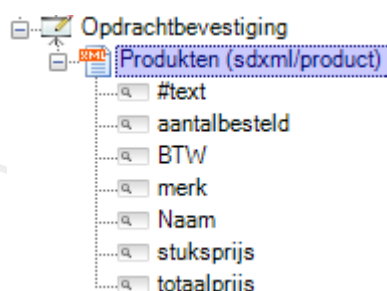
De werking wordt stapsgewijs verklaard:

1. Via 'Laad structuur' Open je de verkenner, selecteer je het te verwerken XML-bestand en wordt vervolgens de inhoud van het XML-bestand ingelezen.
2. De structuur van het XML is in dit venster vervolgens zichtbaar
3. Na selectie van het repeterende element (In dit geval 'Product') is de padverwijzing hier zichtbaar.
4. Vervolgens bevestigen via 'OK'



Afbeelding 9: de eigenschappen van de XML-vraag – Tabblad XML – Laad structuur

Vervolgens is de weergave in de vragenboom gewijzigd en ziet er als volgt uit:

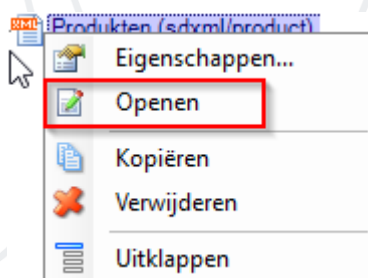


Afbeelding 10: Weergave in SmartEditor

De onderliggende XML-nodes (Childnodes) zijn zichtbaar.

Volgens de standaard werkwijze van SmartDocuments zouden de tags nu in het sjabloon gesleept kunnen worden.

Dit zorgt er echter voor dat alléén de gegevens van het 1<sup>e</sup> product uit het XML-bestand in zullen worden gevuld. Om ervoor te zorgen dat de gegevens van alle repeterende tags in het document komen te staan is er een andere werkwijze benodigd. Dubbelklik met de linker muisknop of klik met de rechtermuisknop op de XML-vraag en selecteer de optie 'Openen'.









Afbeelding 11: klik met de rechtermuisknop op de XML-vraag

De inhoud van de XML-vraag zal getoond worden in een 2<sup>e</sup> tabblad in SmartEditor. Alles wat in de inhoud van de XML-vraag aan tabellen/lay-out/kolommen/etcetera geplaatst wordt, zal voor ieder product waarvan de gegevens in het XML-bestand staan, herhaald worden.

De opdrachtbevestiging die gemaakt moet worden, zal de tabel bevatten die op afbeelding 3 getoond wordt. De tabel moet een opsomming bevatten van alle door de gebruiker aangeschafte producten met hun respectievelijke eigenschappen.

In de XML-vraag moet dus de rij van de tabel worden opgenomen die herhaald wordt.

Maak hiervoor een nieuwe tabelrij met 6 kolommen en sleep achtereenvolgens de tags 'naam', 'merk', 'stuksprijs', 'aantalbesteld', 'totaalprijs' en 'BTW' in de afzonderlijke kolommen:

[  <b>Produkten.Naam</b> ]	[  <b>Produkten.merk</b> ]	€ [  <b>Produkten.stuksprijs</b> ]	[  <b>Produkten.aantalbesteld</b> ]	€ [  <b>Produkten.totaalprijs</b> ]	€ [  <b>Produkten.BTW</b> ]
---	---	---	--	--	--

Afbeelding 12: de tabelrij die herhaald gaat worden

Ga hierna terug naar het sjabloon. Sleep de XML-vraag op de 1<sup>e</sup> regel onder deze rij:

Hartelijk dank voor uw bestelling bij Staples Online !

Hieronder een overzicht van de door u bestelde producten

Naam	Merk	Prijs per stuk incl. BTW	Aantal besteld	Totaalprijs incl. BTW	BTW 21 %
------	------	-----------------------------	----------------	--------------------------	----------

[? Producten ]

De bestelling zal binnen 1 werkdag bij u worden afgeleverd.

*Afbeelding 11: de rij met kolomnamen met daaronder de XML-vraag*

Sla de wijzigingen op en test vervolgens de werking.

### 2.1.1.2 Attributen

Attributen zijn elementen die gebruikt kunnen worden in een XML-tag om aanvullende gegevens op te nemen. Ze worden vaak gebruikt om metadata in op te slaan.

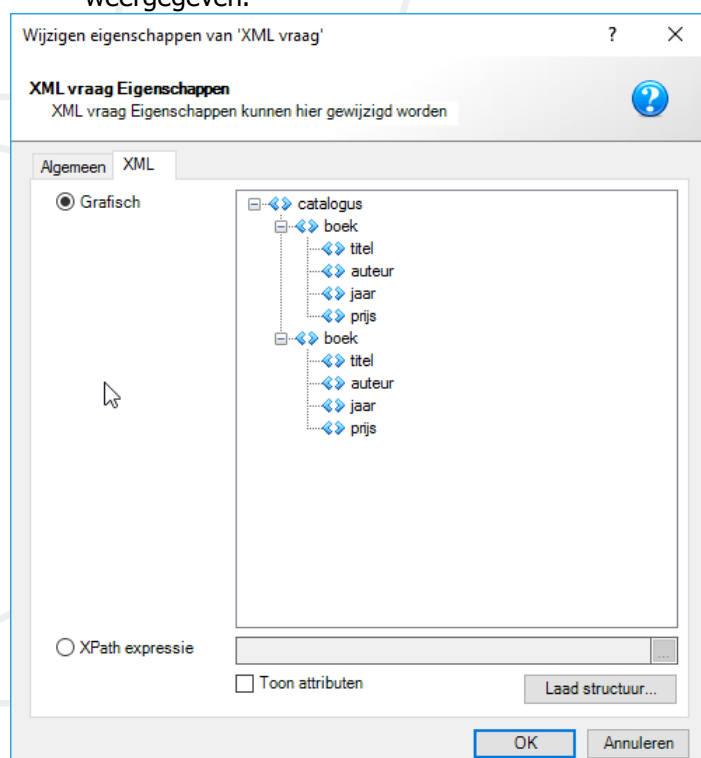
#### Een voorbeeld van het gebruik van attributen in een XML-tag.

XML elementen kunnen attributen bevatten. Die geven extra inhoudelijke informatie die op zich niet in de waarde van het element vervat zit. Attributen bestaan uit een naam, een gelijkheidsteken(=) en een corresponderende waarde. De waarde van elk attribuut staat steeds tussen "dubbele aanhalingstekens". Attributen bevinden zich steeds in de begintag van het element waarover ze informatie geven:

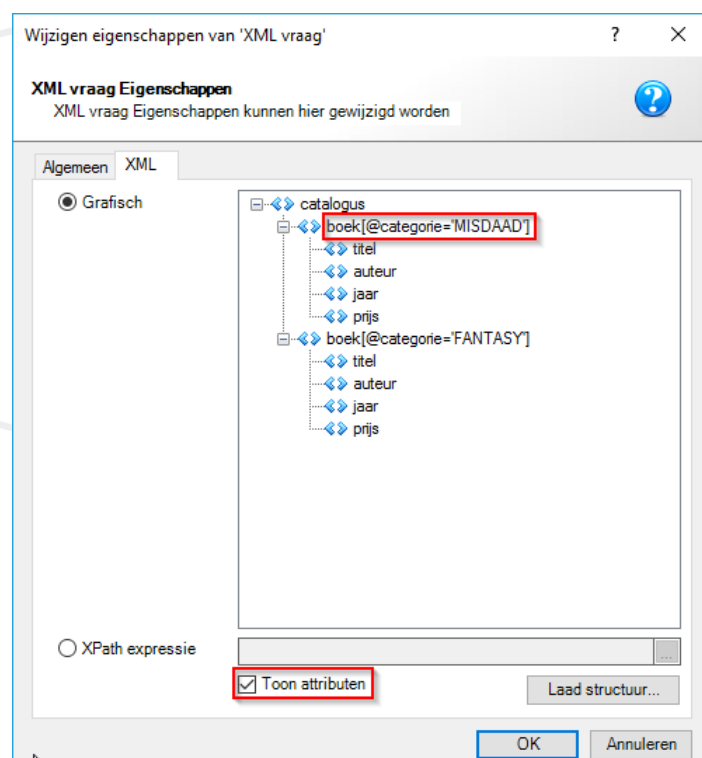
```
<catalogus>
  <boek categorie="MISDAAD">
    <titel>Wallander</titel>
    <auteur>Henning Mankell</auteur>
    <jaar>2001</jaar>
    <prijs>15,95</prijs>
  </boek>
  <boek categorie="FANTASY">
    <titel>Harry Potter en de Steen der Wijzen</titel>
    <auteur>J.K. Rowling</auteur>
    <jaar>1998</jaar>
    <prijs>19,95</prijs>
  </boek>
</catalogus>
```

Afbeelding 12: een XML-node met een attribuut

In de onderstaande afbeeldingen wordt bovenstaande XML **zonder** en **met** de optie 'Toon attributen' weergegeven.



Afbeelding 13: De optie 'Toon attributen'



### 3 XPath

#### 3.1 Wat is XPath?

XPath is een query-taal waarmee de inhoud van een XML-bestand doorzocht kan worden. Er kan dus een commando op een XML-bestand worden afgevuurd. In het geval van de XML-vraag kan het resultaat van de query door SmartDocuments in een document geplaatst worden.

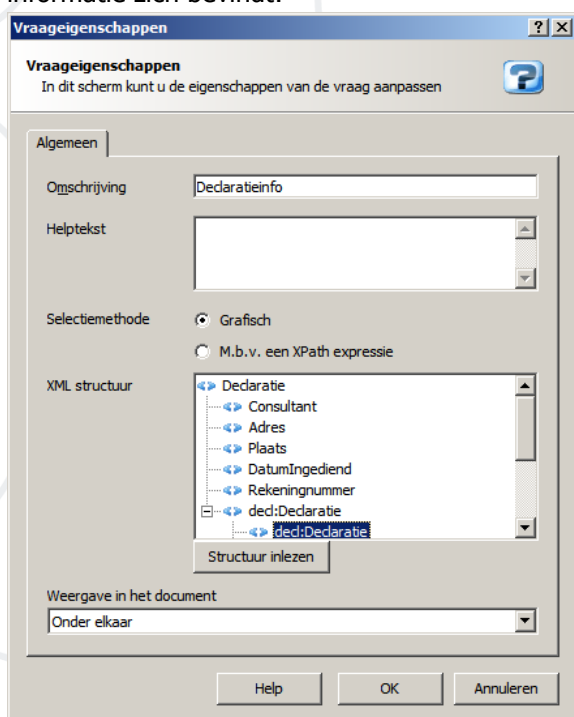
#### 3.2 Waarom XPath?

Zoals op de voorgaande pagina verteld is, kan de XML-vraag nu omgaan met attributen. Op afbeelding 13 is een voorbeeld te zien van een XML-bestand met attributen:

```
<Declaratie>
  <Consultant>S. van Doornspeijk</Consultant>
  <Adres>Dorpsstraat 1</Adres>
  <Plaats>Ons Dorp</Plaats>
  <DatumIngediend>01-31-07</DatumIngediend>
  <Rekeningnummer>166052049</Rekeningnummer>
</decl:Declaratie>
<decl:Declaratie bonnr="1" datum="01-13-07" omschrijving="Kosten internet januari 2007" Naambedrijf="Demon Internet" tbvrelatie="" goedgekeurd="Ja">25,00</decl:Declaratie>
<decl:Declaratie bonnr="2" datum="01-16-07" omschrijving="Onderhoud auto" Naambedrijf="Wasstraat Blinkend Blik" tbvrelatie="" goedgekeurd="Nee" reden="Schoonmaak auto is voor eigen rekening">7,50</decl:Declaratie>
<decl:Declaratie bonnr="3" datum="01-17-07" omschrijving="Parkeren" Naambedrijf="Cityparking" tbvrelatie="gemeente Breed" goedgekeurd="Ja">12,50</decl:Declaratie>
<decl:Declaratie bonnr="4" datum="01-19-07" omschrijving="Reiniging pak" Naambedrijf="Stomerij Speedy Clean" tbvrelatie="" goedgekeurd="Nee" reden="Reiniging werkkleding is voor eigen rekening">15,00</decl:Declaratie>
<decl:Declaratie bonnr="5" datum="01-23-07" omschrijving="Parkeren" Naambedrijf="Parking Lokaal" tbvrelatie="gemeente Amsterdam" goedgekeurd="Ja">75,00</decl:Declaratie>
<decl:Declaratie bonnr="6" datum="01-24-07" omschrijving="Parkeren" Naambedrijf="Wielklem BV" tbvrelatie="gemeente Enschede" goedgekeurd="Ja">7,50</decl:Declaratie>
</decl:Declaratie>
</Declaratie>
```

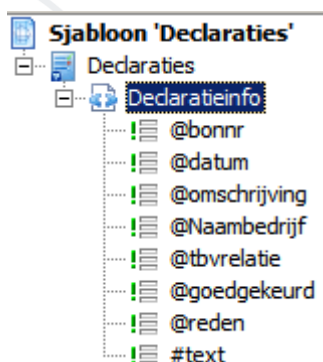
Afbeelding 13: een XML-bestand met attributen

Als dit bestand ingelezen wordt in de XML-vraag, kan grafisch aangegeven worden waar de herhalende informatie zich bevindt:



*Afbeelding 14: grafische selectie van een herhalende tag*

Als er bij afbeelding 14 op 'OK' geklikt wordt, zullen de attributen en de waarde van de herhalende tags onder elkaar worden opgesomd. Attributen onderscheiden zich door een '@' voor de naam. De waarde van de tag wordt opgesomd in het '#text' veld:

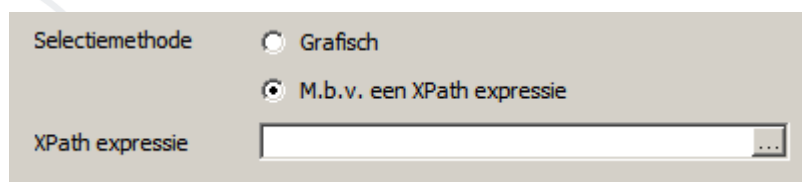


*Afbeelding 15: attributen en waarde van de repeterende tags*

Met behulp van de op pagina 8 beschreven functionaliteit kan er nu een tabel worden gemaakt waarin de declaraties onder elkaar worden opgesomd. Maar wat als er nu 2 opsommingen gemaakt moeten worden, 1 van de goedgekeurde en 1 van de afgekeurde declaraties? Oftewel, als het attribuut 'goedgekeurd' de waarde 'Ja' bevat, dan moet de declaratie in het document worden geplaatst. Dit soort zaken kunnen met de gebruikelijke grafische wijze niet gerealiseerd worden, hiervoor zal een XPath-expressie gemaakt moeten worden.

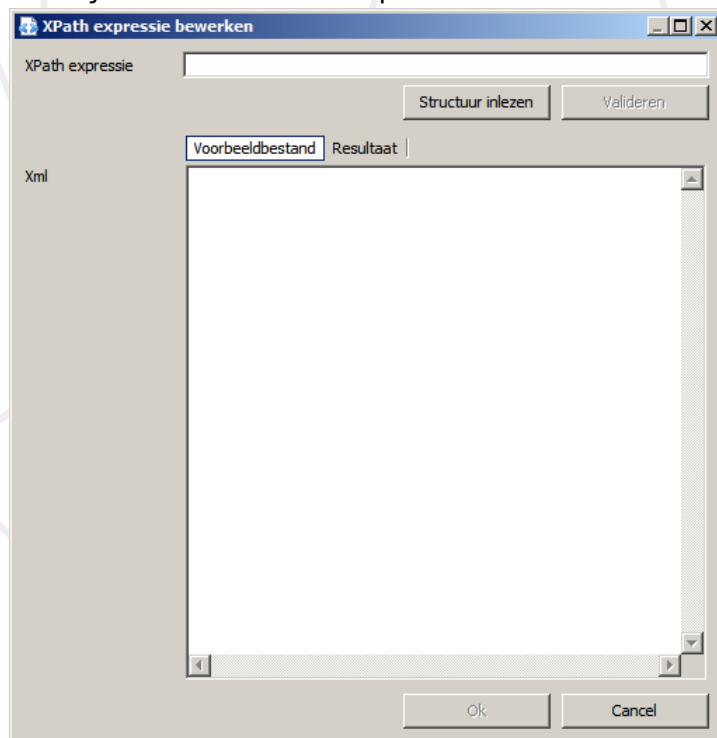
## 4 XPath-expressies

Naast het grafisch selecteren van een herhalende tag is het dus ook mogelijk om een XPath-expressie in te voeren. Selecteer hiervoor bij 'Selectiemethode' de optie 'M.b.v. een XPath-expressie':



*Afbeelding 16: m.b.v. een XPath-expressie*

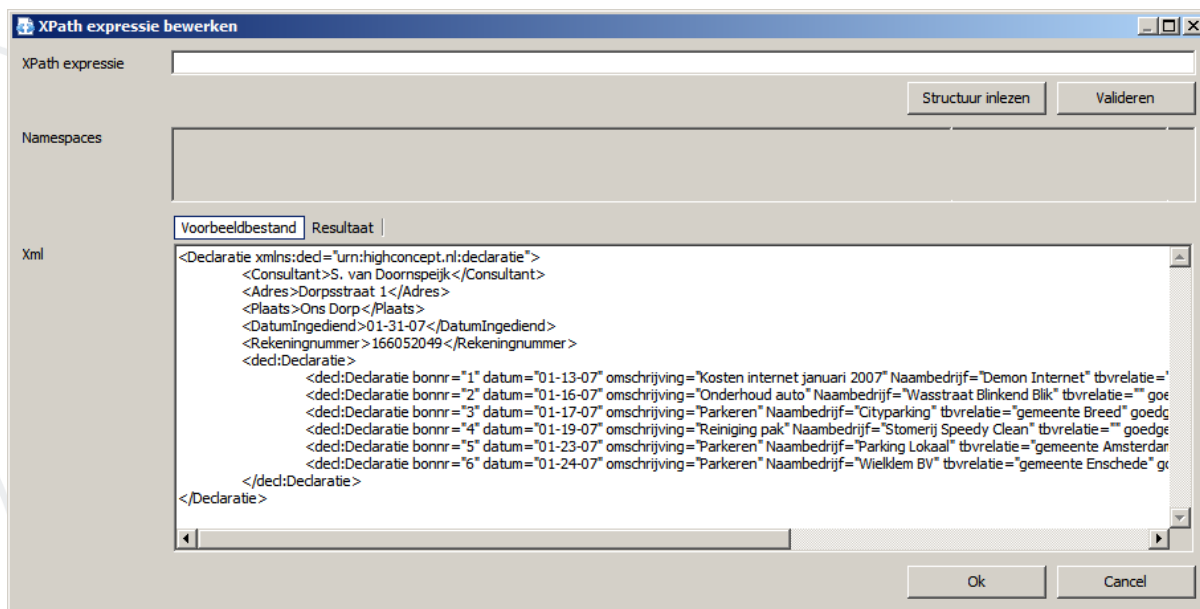
Als er vervolgens op de knop bij het tekstvak 'XPath-expressie' geklikt wordt, zal er een scherm verschijnen waarin de XPath-expressie bewerkt kan worden:



*Afbeelding 17: het scherm 'XPath-expressie bewerken'*



Als er op de knop 'Structuur inlezen' geklikt wordt, verschijnt er een scherm waar een XML-bestand geselecteerd kan worden. Als het XML-bestand is ingelezen zal de structuur getoond worden:

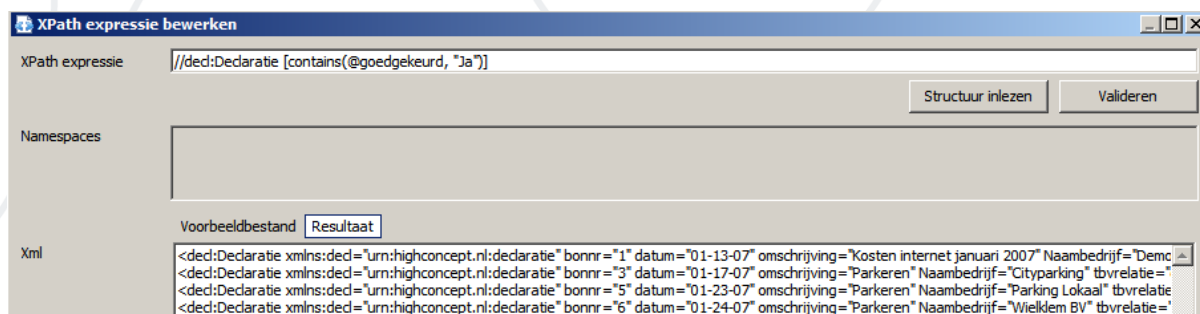


Afbeelding 18: de structuur van het ingelezen XML-bestand

In het invoervak achter 'XPath expressie' kan nu een expressie in worden gevoerd. Het is de bedoeling dat het document een opsomming bevat van alle goedgekeurde declaraties. De waarde van het attribuut 'Goedgekeurd' in de tag 'decl:Declaratie' moet dus 'Ja' zijn. De XPath-expressie hiervoor is als volgt:

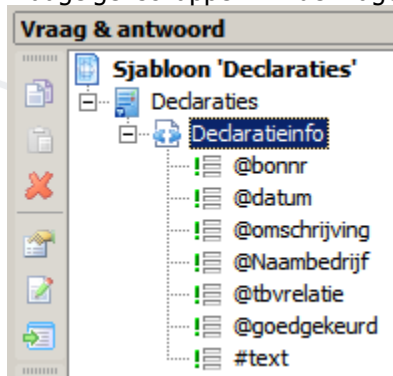
**`//decl:Declaratie [contains(@goedgekeurd, "Ja")]`**

Deze expressie zorgt ervoor dat er in de tag 'decl:Declaratie' gezocht wordt naar een attribuut 'goedgekeurd' (onderscheiden door de '@') die de waarde 'Ja' bevat. Als deze expressie ingevoerd is, kan er op 'Valideren' geklikt worden. Er zal een opsomming van de tags getoond worden die voldoen aan de ingevoerde expressie:



Afbeelding 19: het resultaat van de expressie

Bevestig de invoer van de expressie door op 'OK' te klikken op het expressiescherm en bij de vraageigenschappen. In de vragenboom worden de herhalende velden onder de XML-vraag geplaatst:



Afbeelding 20: de velden afkomstig uit de XPath-expressie

Met behulp van de vanaf pagina 6 beschreven functionaliteit kan er nu een tabel worden gemaakt waarin de herhalende gegevens onder elkaar worden opgesomd. Als SmartDocuments doorgestart wordt met het XML-bestand en het declaratieformulier wordt opgebouwd, zal er dus een opsomming van alle goedgekeurde declaratie getoond worden:

Goedgekeurde declaraties:					
Bonnr.	Datum	Omschrijving	Naam bedrijf	t.b.v relatie	Bedrag
1	01-13-07	Kosten internet januari 2007	Demon Internet		25,00
3	01-17-07	Parkeren	Cityparking	gemeente Breed	12,50
5	01-23-07	Parkeren	Parking Lokaal	gemeente Amsterdam	75,00
6	01-24-07	Parkeren	Wielklem BV	gemeente Enschede	7,50

Afbeelding 21: een opsomming van de goedgekeurde declaraties

Als er ook een opsomming van de afgekeurde declaraties moet komen, zal er een XML-vraag toe moeten worden gevoegd. De XPath-expressie zal ook moeten worden aangepast, er moet nu immers gezocht worden naar attributen met de naam 'Goedgekeurd' die als waarde 'Nee' hebben:

**//decl:Declaratie [contains(@goedgekeurd, "Nee")]**

Met behulp van de op de voorgaande pagina's beschreven functionaliteit kan er ook voor de afgekeurde declaraties een tabel gemaakt worden waar de afgekeurde declaraties in opgesomd worden. Op afbeelding 22 is deze opsomming te zien.

Goedgekeurde declaraties:

Bonnr.	Datum	Omschrijving	Naam bedrijf	t.b.v relatie	Bedrag
1	01-13-07	Kosten internet januari 2007	Demon Internet		25,00
3	01-17-07	Parkeren	Cityparking	gemeente Breed	12,50
5	01-23-07	Parkeren	Parking Lokaal	gemeente Amsterdam	75,00
6	01-24-07	Parkeren	Wielklem BV	gemeente Enschede	7,50

Afgekeurde declaraties:

Bonnummer	Datum	Omschrijving	Naam bedrijf	t.b.v relatie	Bedrag	Reden
2	01-16-07	Onderhoud auto	Wasstraat Blinkend Blik		7,50	Schoonmaak auto is voor eigen rekening
4	01-19-07	Reiniging pak	Stomerij Speedy Clean		15,00	Reiniging werkkleding is voor eigen rekening

Afbeelding 22: de opsommingen van de goed- en afgekeurde declaraties

## **Contactgegevens**

Voor vragen of opmerkingen over deze documentatie kunt u contact opnemen met de servicedesk via e-mail op [servicedesk@smartdocuments.nl](mailto:servicedesk@smartdocuments.nl). Op werkdagen is de servicedesk ook telefonisch bereikbaar tussen 8.30 en 17.00 op 0570 – 62 85 49 of

## **Disclaimer**

Ondanks de voortdurende inspanningen van SmartDocuments Nederland BV om ervoor te zorgen dat de informatie in dit document zo compleet en up-to-date mogelijk is, kan SmartDocuments niet instaan voor de juistheid en/of volledigheid en/of de specifieke toepasbaarheid van de gepubliceerde en/of de gevraagde informatie in dit document. SmartDocuments is niet aansprakelijk voor enige directe, indirecte, incidentele, bijzondere of gevolgschade, gederfde winst of voor onderbreking van de bedrijfsvoering die voortvloeien uit het gebruik van dit document. De winning en het gebruik van informatie uit dit document blijft te allen tijde voor eigen risico van de gebruiker.