

Queries

Applicatiebouw

Portaal *beheerportaal-Nieuw*. Tegel *Queries*.

Screenidentifiers:

- MDLC_getQueriesList.xml
- MDDC_getQueryDetail.xml

De SQL-statements van deze tabel worden gebruikt:

- om opschriften of staaf- of taartdiagrammen te genereren voor tegels
- om bepaalde scherm informatie in detailschermen per blok (on)zichtbaar te maken
- om inhoud van schermattributen in detailschermen zoals labels of actions, context-afhankelijk te maken
- om blokken tekst in te voegen in een document gemaakt vanuit een sjabloon.
- om resultsets te genereren ten behoeve van merge bij documentsjablonen

Voor querybeheer moet de inlogger beheerrechten hebben: `tbmedewerker.dnbeheerniveau = 99` (of hoger). Alle kolommen en knoppen zijn dan toegankelijk.

Een aantal query's in deze tabel is noodzakelijk voor het programma. Dit zijn de query's die het vakje systeem aangevinkt hebben staan. De waarde van het systeem-aanvinkvakje kan niet worden gewijzigd met OpenWave. Aangevinkt betekent dat bij database-updates van OpenWave de bijbehorende query gewijzigd kan worden. Door gebruikers aangemaakte query's (met systeem = niet aangevinkt) zullen bij updates onaantast blijven.

Aan de query worden de volgende eisen gesteld:

- EN de query moet met 'select' beginnen
- EN er mag GEEN puntkomma(';') in voorkomen (vanwege het gevaar voor SQL-injectie).

Voorbeeld

Een voorbeeld van een query is:

```
select
'Aantal: ' | | count(*)
from vwfrmomgorkestrator_adv
where trim(dvadviesbehandelaar) = trim(:keyaccount)
```

Deze query geeft het aantal openstaande adviezen, waarvoor de inlogger de behandelaar is.

Bij query's kunnen twee variabelen worden gebruikt:

- `:keyaccount` zal worden vervangen door `tbmedewerkers.dvcode` van de inlogger
- `{id}` wordt:
 - vervangen met de `dnkey` van `tbomgvergunning` of `tbhandhavingen` of `tbmilinrichtingen` of

tbovvergunningen of tbmilvergunningen wanneer de query gebruikt wordt op een zaak- of inrichtingsportaal als tegelopschrift

- vervangen met de dnkey van de kaart die hoort bij de basistabel van het detailscherm van waaruit de query wordt aangeroepen als onderdeel van schermattributen bij de schermkolomdefinitie
- vervangen met de dnkey van de kaart van de tabel waaruit een document wordt gecreëerd.

OpenWave heeft zelf een aantal functies op de database gedefinieerd - zoals `fn_ddmaandjjjj()` - die gebruikt kunnen worden in allerlei queries. Zie: [OpenWave database functies](#).

De query heeft in de kolom `dvcode` een unieke, maar editbare, identifier, waarmee de query kan worden aangeroepen.

Tot slot KAN het evalueren van een query aan rechten worden gekoppeld. Dat wordt per query geregeld met de velden *rechtentabelnaam* en *rechtenkolomnaam*. De inlogger is via zijn rechtengroep verbonden met *tbrechten* of *tbomgrechten*, *tbhorrechten*, *tbmilrechten*, *tbmilvergcrechten*, *tbhhrechten*, *tbovrechten*, *tbinfoe rechten*. Dat zijn de mogelijke waarden van kolom *rechtentabelnaam*. De kolom *rechtenkolomnaam* verwijst naar een van de veldnamen van die rechtentabellen. De waarde van die veldnaam is F of T. Indien F dan wordt de query niet uitgevoerd. En als deze kolommen worden leeggelaten is er dus geen extra restrictie en wordt de query altijd geëvalueerd.

De kijkrechten op de modules:

- Omgevingszaken: *tbrechten* en *dlaomgvsb*
- Handhavingen: *tbrechten* en *dlahahvsb*
- APV/Overige: *tbrechten* en *dlaovvvsb*
- Objecten/inrichtingen: *tbrechten* en *dlamilinrvsb*
- Milieu/gebruik: *tbmilvergcrechten* en *dlamilvergvsb*

Common table expressie WITH gebruik in queries

WITH mag gebruikt worden in queries waarbij het volgende geldt:

- de query moet beginnen met WITH
- in de gehele query mag NIET de substring *update* of *insert* of *delete* of *merge* voorkomen.
- haakjes (of) mogen alleen voor de syntax gebruikt worden: dus niet *select d1logic val)1* en ook niet: *where lower(dvsectie) = 'ad)viezen'*
- na het sluithaakje van de with expressie begint de query met *select*
- er mag GEEN puntkomma (;) in de query voorkomen.

Voorbeeld

```
with const as (select dllogic val1
                from tbinitialisatie where lower(dvsectie) = 'adviezen' and
                upper(dvitem) = 'RETOURISGELEVERD')
select case
        when (SELECT val1 FROM const) = 'F'
```

```

        then to_char(coalesce(ddadvadvies,ddvervallen),'dd-mm-yyyy') || ' ' ||
dadvinstantie
        else to_char(coalesce(ddadvdatering,ddvervallen),'dd-mm-yyyy') || ' '
|| dadvinstantie
    end
from vwfrmadviezen
    where dnkeyomgvergunningen = {id}
and    case
        when (SELECT val1 FROM const) = 'F'
        then ((ddadvadvies is not null) or (ddvervallen is not null))
        else ((ddadvdatering is not null) or (ddvervallen is not null))
        end
order by 1 LIMIT 4

```

Queries voor tegelopschrift

Het resultaat van een query (dus de evaluatie van het select-statement) mag in dit geval maar uit één kolom bestaan.

De aanroep van het SQL-statement vindt plaats via de tegeldefinitie van de tegel waarop het dynamische opschrift moet verschijnen door het vullen van de kolom *Tegelopschrift dynamisch met API gettilecontent of getFlexChart()* (zie [Portal tegel](#)).

Een voorbeeld van een dynamisch tegelopschrift is de waarde *getTileContent(omgeving_status,{id})*. De codering *omgeving_status* verwijst naar een uniek codering in de tabel *tbqueries*. Het SQL-statement dat aldaar staat zal worden gebruikt om het dynamische deel van het tegelopschrift te genereren.

Die parameter {id} kan gebruikt worden bij tegels op de zaakportals (dus niet bij opening of beheer). Deze wordt vervangen door de primary key van het betreffende zaakportaal (de identifier die opgenomen is in de URL van de portaalpagina). Indien de resultaat set uit meerdere regels bestaat, zal OpenWave deze aan elkaar plakken gescheiden door een puntkomma, zodat elke regel ook een regel op de tegel wordt.

Met HTML-code kan die éne resultaatkolom van een query toch in twee regels op een tegel getoond worden door op de gewenste plek bijvoorbeeld `<br>` in te voegen. Bovendien kunt u ook andere HTML-code gebruiken bijvoorbeeld *kleur*:

```

select
    '<p style="color:red">' || dvstatus || '</p>'
    || '<br>startdatum: ' || to_char(ddaanvraag,'dd-mm-yyyy')
    || '<br>fatale/streefdatum: '
    || to_char(ddfataledatum,'dd-mm-yyyy')
    || case when ddingetrokken is not null
        then '<br>ingetrokken: ' || to_char(ddingetrokken,'dd-mm-yyyy')
        else
            '<br>besluit: ' || to_char(ddbesluitdatum,'dd-mm-yyyy')
        end
from vwfrmomgvergunningen
where dnkeyomgvergunning = {id}

```

Het resultaat als tegelopschrift is bijvoorbeeld:



Wanneer een query niet valide SQL-code gebruikt zal het programma - bij gebruik voor tegelopschrift - het resultaat vervangen door 'fout:xml': dat zal dan op de betreffende tegel verschijnen.

Voorbeeld belang snelheid

Dynamische tegelkopschriften kunnen het uitschrijven van een portaal behoorlijk vertragen.

De volgende query zal traag zijn

```
SELECT
  'Aantal: ' || count(*) from vwfrmomgorkestrator_lopend
  where (trim(dvibbcodemw) = trim(:keyaccount)
  or trim(dvcodeaccountman) = trim(:keyaccount))
```

maar onderstaande query geeft eenzelfde soort resultaat, maar dan in een fractie van de tijd

```
SELECT
  CASE
    WHEN count(*) < 5 THEN 'Aantal: ' || count(*)
    ELSE 'Aantal: 5 of meer'
  END opschrift
  FROM (select a.dnkey from tbomgvergunning a
    inner join tbinbehandelingbij b
    on (a.dnkey = b.dnkeyomgvergunningen and b.dlactief = 'T')
    where (trim(b.dvcodemedewerker) = trim(:keyaccount) or
  trim(a.dvcodeaccountman) = trim(:keyaccount))
    and a.ddbesluitdatum is null
    and a.ddblokkering is null
    and a.ddingetrokken is null
  limit 5) as c
```

In het beheerportaal bij de definitie van de portalen (portalnames) staat (sinds versie 1.30.1) onderaan het detailscherm van een portaaldefinitie een knop waarmee de duur van het uitvoeren van de dynamische tegelopschriften voor dat portaal kan worden gemeten.

Queries voor tonen van een diagram als tegelopschrift

Het gaat hierom dat de query een JSON-string genereert die OpenWave via de charts.js library rendert naar een staaf-, lijn- of taartdiagram.

Zie hiervoor [Grafieken](#).

Query's om blokken onzichtbaar te maken in detailscherm

```
<blok>
  <label>Hyperlink</label>
  <width>100</width>
  <height>40</height>
  <type>doorlopend</type>
  <notvisibleif>%query(omgeving_blokhypertext)%</notvisibleif>
  <column value="dvhyperlink">
```

Zie [Scherm informatie voor detailschermen](#). In de tags van een blok van een detailscherm kan (hoofdt dus niet) de tag `<notvisibleif>` opgenomen worden (onder de tag `<type>`). De inhoud van de tag verwijst naar een bepaalde query. Zo zal `<notvisibleif>%query(omgeving_hypertext)%</notvisibleif>` verwijzen naar een rij in tbqueries met dvcode = *omgeving_hypertext*.

De bijbehorende query wordt geëvalueerd. De uitkomst van de query moet 1 of true (het blok is NIET zichtbaar) dan wel 0 of false (WEL zichtbaar) zijn. Voorbeeld:

```
select case when (dlmappenhyperlink = 'F') then 1 else 0 end from tbrechten
      where dnkey = (select dnkeyrechten from tbmedewerkers where
trim(dvcode) = trim(:keyaccount))
```

Indien wordt verwezen naar een niet bestaande query of het element is leeg of indien error bij evaluatie of uitkomst is `<> 1` dan beschouwt het programma de uitkomst van de query als 0 (het blok is wel zichtbaar).

Indien in de aangeroepen query gebruik wordt gemaakt van de variabele {id} dan moet een tweede parameter worden toegevoegd aan de functie query(), waarbij de string %keypointer% vervangen zal worden door de dnkey van de actieve rij uit de tabel.

Voorbeeld onzichtbaarheidsquery bij twee bijna identieke blokken

Het detailscherm van het inspectietraject bevat twee blokken *indeling*. Eén blok heeft drie mogelijkheden om een onderwerp in te vullen en het andere blok heeft slechts één onderwerp-editbox. Er is een instelling gemaakt onder de naam *Sectie: Inspecties en Item: ExtraOnderwerpVelden* die bepaalt welk van de twee blokken zichtbaar is. Een query onder de naam *inspecties_extra_ow* kijkt naar deze instelling. Die query is als volgt gedefinieerd:

```
select case when not exists (select dnkey from tbinitialisatie where
upper(dvsectie) = 'INSPECTIES' and upper(dvitem)= 'EXTRAONDERWERPVELDEN')
```

```
then case when {id} = '1' then 'true' else 'false' end when (select d1logic from tbinitialisatie where
upper(dvsectie) = 'INSPECTIES' and upper(dvitem)= 'EXTRAONDERWERPVELDEN') = 'F' and {id} = '1'
```

then 'true' when (select d1logic from tbinitialisatie where upper(dvsectie) = 'INSPECTIES' and upper(dvitem) = 'EXTRAONDERWERPVELDEN') = 'F' and {id} = '0' then 'false' when (select d1logic from tbinitialisatie where upper(dvsectie) = 'INSPECTIES' and upper(dvitem) = 'EXTRAONDERWERPVELDEN') = 'T' and {id} = '0' then 'true' when (select d1logic from tbinitialisatie where upper(dvsectie) = 'INSPECTIES' and upper(dvitem) = 'EXTRAONDERWERPVELDEN') = 'T' and {id} = '1' then 'false' else 'false' end

In de schermdefinitie van het inspectietrajectdetails *MDDC_geefInspTrajectDetail.xml* is het blok indeling dat maar één onderwerp toont als volgt gedefinieerd (in de tag <notvisibleif> wordt de query *inspecties_extra_ow* aangeroepen met '0' als parameter: de {id} in de query wordt daarmee gesubstitueerd).

```
<blok>
<label>Indeling</label>
<width>105</width>
<height>85</height>
<type>doorlopend</type>
<notvisibleif>%query(inspecties_extra_ow,'0')%</notvisibleif>
<column value="dnkeyinspaanleiding">
  <regel>1</regel>
  . . . . .
```

In de schermdefinitie van het inspectietrajectdetails *MDDC_geefInspTrajectDetail.xml* is het blok indeling dat drie onderwerpregels toont als volgt gedefinieerd (in de tag <notvisibleif> wordt de query *inspecties_extra_ow* aangeroepen met '1' als parameter: de {id} in de query wordt daarmee gesubstitueerd).

```
<blok>
<label>Indeling</label>
<width>105</width>
<height>170</height>
<type>doorlopend</type>
<notvisibleif>%query(inspecties_extra_ow,'1')%</notvisibleif>
<column value="dnkeyinspaanleiding">
  <regel>1</regel>
  . . . . .
```

Zie verder voorbeelden hieronder bij Query's voor contextafhankelijke attributen.

Query's voor contextafhankelijke schermkolommen

Vooralsnog wordt een queryaanroep ondersteunt bij de tags label, edit, bgcolor, fontcolor, visible en action en filter.

Label

Een query kan worden gebruikt om de inhoud van de tag label contextgevoelig te maken.

Voorbeeld: wanneer in de schermkolomdefinitie van een detailscherm staat `<label>Datum %query(omg_labelbesluit)%</label>` zal het programma dynamisch het label samenstellen met de tekst 'Datum ' + daarachter het resultaat van de geëvalueerde query uit de tabel tbqueries met dvcode = `omg_labelbesluit`.

Indien de contextgevoeligheid is gebaseerd op andere informatie uit de kaart die getoond wordt (de query zal dan bestaan uit `select x from y where dnkey = {id}`) dan moet een tweede parameter `%keypointer%` meegegeven worden: `<label>Datum %query('omg_labelbesluit,%keypointer%')%</label>`. De variabele `{id}` uit de query zal vervangen worden met de primary key (dnkey) van de kaart waar je op staat.

Zie de voorbeelden hieronder bij [action](#).

Visible

Een query kan worden gebruikt om een kolom contextgevoelig in het scherm op te nemen. In de schermkolomdefinitie (zowel voor kolommen van een lijst- als die voor het detailscherm) kan tag `<visible>` opgenomen zijn. Indien deze optionele tag de waarde false of 0 heeft dan moet de betreffende kolom NIET worden opgenomen. Indien deze optionele tag de waarde true of 1 heeft, of indien de tag niet bestaat, dan wordt de kolom wel opgenomen.

Een query kan worden gebruikt om de inhoud van de tag contextgevoelig de waarde true of false te geven. Voorbeeld: wanneer in de schermkolomdefinitie van een detailscherm staat `<visible>%query(kolomXzichtbaar,%keypointer%)%</visible>` zal het programma de tag visible vullen met de waarde die komt uit de aangeroepen geëvalueerde query. Indien die waarde ongelijk aan 'true' is dan moet de kolom NIET worden opgenomen.

De query zal dan bestaan uit zoiets als `select case when x then 'true' else 'false' from y where dnkey = {id}` waarbij `{id}` onder water wordt vervangen door de dnkey van kaart waar je op staat (de parameter `%keypointer%` in de aanroep). Indien de query niet valide is, dan wordt de tag gevuld met 'false'.

Er bestaat een systeemquery met dvcode = `geefWaardeRechtenkolom` die makkelijk toegang geeft tot het rechtensysteem van OpenWave en false of true retourneert. Voorbeeld is het zichtbaar zijn van de knop wijzig bevoegd gezag in het detailscherm van de omgevingszaak. In de schermkolomdefinitie is de tag visible als volgt:

```
<visible>%query(geefwaarderechtenkolom,'tbomgrechten.dlbomgbevgezede')%</visible>
```

De query kijkt voor de betreffende inlogger naar de waarde van de kolom `tbomgrechten.dlbomgbevgezede`.

Edit

Een query kan worden gebruikt om de tag `edit` contextgevoelig de waarde true of false te geven.

Voorbeeld: wanneer in de schermkolomdefinitie van een detailscherm staat `<edit>%query(omgeving_mddc_schermknopolo,%keypointer%)%</edit>` zal het programma het attribuut *edit* vullen met de waarde die komt uit de aangeroepen geëvalueerde query.

De query zal dan bestaan uit zoiets als *select case when x then 'true' else 'false' from y where dnkey = {id}* waarbij {id} onder water wordt vervangen door de dnkey van kaart waar je op staat (de parameter %keypointer% in de aanroep). Indien de query niet valide is, dan wordt het attribuut *edit* gevuld met 'false'.

Er bestaat een systeemquery met dvcode = *geefWaardeRechtenkolom* die makkelijk toegang geeft tot het rechtensysteem van OpenWave en false of true retourneert. Zie hierboven bij visible: *geefwaarderechtenkolom*.

Backcolor / Fontcolor

Een query kan worden gebruikt om de tags backcolor en fontcolor contextgevoelig de waarde true of false te geven. Het resultaat van de query moet gevuld zijn met standaard html colorname (https://www.w3schools.com/colors/colors_names.asp).

Action

Een action kan in een schermkolomdefinitie worden gekoppeld aan een schermknop. Bijvoorbeeld een knop achter de kolom OLO-nummer in de omgevingsdetailkaart waarmee het OLO-loket met de juiste aanvraag wordt geopend afhankelijk van het OLO-nummer op de kaart: Of een knop die een PDOK-kaart opent op basis van de coördinaten van het bijbehorende locatie adres. Die action is dus contextgevoelig.

Voorbeeld 1: ga naar OLO-loket op juiste aanvraagnummer

De URL daartoe is:

<https://www.omgevingsloket.nl/BevoegdGezag/bevoegdgezag/AanvraagTab/Aanvraag/????/AanvraagGegevens>

Waarbij op de plaats van de vraagtekens een valide OLO-aanvraagnummer moet worden gevuld. Dat lossen we op door:

- een knop te definiëren op het detailscherm van de omgevingszaak met een action waarin bovengenoemde URL wordt gedefinieerd
- in de URL een queryverwijzing te maken teneinde het juiste OLO-aanvraagnummer op te halen
- een query te maken die het gevraagde OLO-nummer retourneert.

De knop kan als volgt worden gedefinieerd:

```
<column>  
  <regel>3</regel>  
  <tagnaam/>
```



```

<label/>
<divwidth>20</divwidth>
<divheight>30</divheight>
<edit>>true</edit>
<showhint>>false</showhint>
<wavetype>schermmknop</wavetype>
<source></source>
<filter>enable</filter>
<refresh>>false</refresh>
<backcolor/>
<fontcolor/>
<nullable>>true</nullable>
<icoon>14</icoon>
<align/>
<pretext/>
<posttext/>
<hint>ga naar omgevingsloket</hint>
<action>openTabPage(https://www.omgevingsloket.nl/BevoegdGezag/bevoegdgezag/
AanvraagTab/Aanvraag/%query(omgeving_olnummer,%keypointer%)/AanvraagGegeve
ns)</action>
</column>

```

Een action die begint met de functie *openTabPage* laat het programma als het ware de inhoud van *openTabPage()* intikken in de URL-editbox van de browser met als gevolg dat in een nieuw browsertabblad de gevraagde site wordt geopend. In de URL is het OLO-nummer opgenomen door een verwijzing naar een query met dvcode = *omgeving_olnummer*. Die query ziet er als volgt uit:

```
select dvlvoaanvraagnr from tbomgvergunning where dnkey = {id}
```

Aangezien in de query de variabele {id} is opgenomen die vervangen moet worden door de dnkey van de betreffende omgevingszaak, is de aanroep naar de query voorzien van een tweede parameter %keypointer% dus: %query(omgeving_olnummer,%keypointer%)%.

Stel het OLO-nummer is '123456' dan zal het uiteindelijke resultaat van het indrukken van de schermknop zijn het openen van de URL:

```
https://www.omgevingsloket.nl/BevoegdGezag/bevoegdgezag/AanvraagTab/Aanvraag/123456/AanvraagGegevens
```

Voorbeeld 2: ga naar ruimtelijke plannen op grond van adresgegevens

De URL daartoe is:

<http://www.ruimtelijkeplannen.nl/web-roo/roo/bestemmingsplannen?xxxx&yyyy>

Waarbij op de plaats van de xxxx de tekst 'postcode=' + een valide postcode moet worden ingevoerd en op de plaats van de yyyy moet de tekst 'huisnummer=' + een huisnummer en eventueel een huisletter worden ingevoerd.

Dat lossen we op door:

- een knop te definiëren op het detailscherm van het locatie adres met een action waarin bovengenoemde URL wordt gedefinieerd
- in de URL een queryverwijzing te maken teneinde de tekst 'postcode=' + de gewenste postcode op te halen
- in de URL een queryverwijzing te maken teneinde de tekst 'huisnummer=' + een huisnummer en huisletter op te halen
- een query te maken die de gevraagde postcode informatie retourneert
- een query te maken die het gevraagde huisnummerinformatie retourneert.

De knop kan als volgt worden gedefinieerd:

```
<column>
  <regel>1</regel>
  <tagnaam/>
  <label/>
  <divwidth>20</divwidth>
  <divheight>30</divheight>
  <edit>true</edit>
  <showhint>false</showhint>
  <wavetype>schermknop</wavetype>
  <source></source>
  <filter>enable</filter>
  <refresh>false</refresh>
  <backcolor/>
  <fontcolor/>
  <nullable>true</nullable>
  <icoon>14</icoon>
  <align/>
  <pretext/>
  <posttext/>
  <hint>ga naar ruimtelijke plannen</hint>
  <action>openTabPage(http://www.ruimtelijkeplannen.nl/web-roo/roo/bestemmingsplannen?%query(locatie_postcode,%keypointer%)&%query(locatie_huisnr,%keypointer%))</action>
</column>
```

In de URL is het postcodegedeelte opgenomen door een verwijzing naar een query met dvcode = locatie_postcode. Die query ziet er als volgt uit:

```
select 'postcode=' || coalesce(dvpostcode,'') from tbperceeladressen where
dnkey = {id}
```

Aangezien in de query de variabele {id} is opgenomen die vervangen moet worden door de dnkey van de betreffende locatie (tbperceeladressen), is de aanroep naar de query voorzien van een tweede parameter %keypointer%, dus: %query(locatie_postcode,%keypointer%)%.

In de URL is het huisnummargedeelte opgenomen door een verwijzing naar een query met dvcode = locatie_huisnr. Die query ziet er als volgt uit:

```
select 'huisnummer=' || coalesce(trim(dvhuisnummer),'') ||
coalesce(dvhuisletter,'') from tbperceeladressen where dnkey = {id}
```

Aangezien in de query de variabele {id} is opgenomen die vervangen moet worden door de dnkey van de betreffende locatie (tbperceeladressen), is de aanroep naar de query voorzien van een tweede parameter %keypointer%, dus: %query(locatie_huisnummer,%keypointer%)%.

Stel de postcode van de locatiekaart waarvandaan de knop wordt aangesproken = 1074XK en het huisnummer = 41, dan zal een URL geopend worden met:

```
http://www.ruimtelijkeplannen.nl/web-roo/roo/bestemmingsplannen?postcode=1074XK&huisnummer=41
```

Voorbeeld 3 Open een PDOK-kaart op basis van een coördinatenpaar

De URL van bijvoorbeeld het x, y punt (168875,382535) op een PDOK-kaart is:

```
https://kaart.pdok.nl/api/api.html?mloc=168875,382535&mt=mt3&loc=168875,382535&zoom=10&tekst=Locatie:%3C/BR%3E168875,382535
```

De action:

```
opentabpage(https://kaart.pdok.nl/api/api.html?mloc=%query(coordinaten,%keypointer%)%&mt=mt3&loc=%query(coordinaten,%keypointer%)%&zoom=10&tekst=Locatie:%3C/BR%3E%query(coordinaten,%keypointer%)%)
```

De query *coördinaten* moet dus als resultaat een x- en een y-coördinaat teruggeven (in rijksdriehoek) gescheiden door een komma.

Filter

Een query kan worden gebruikt om de inhoud van de tag filter contextgevoelig te maken.

Voorbeeld: wanneer bij een processtap een extra invoer kolom is gedefinieerd van type dropdown, dan wordt de bijbehorende query voor het dropdownlijstje in het beheer bij de definitie van de processtap gedefinieerd. Deze wordt bij het binnenhalen van een proces vervolgens gekopieerd naar de tabel tbtermijnbewstappen om de kolom dvinvoerdrops aldaar met een dropdownlijstje te kunnen vullen. Om nu - dynamisch - het juiste dropdownlijstje te krijgen is de kolom dvinvoerdrops in de schermkolomdefinitie van het detailscherm van een processtap als volgt gedefinieerd (inclusief contextgevoelige zichtbaarheid en label):

```
<column value="dvinvoerdrops">
    <regel>3</regel>
```

```
<tagnaam>dvinvoerdropd</tagnaam>
<label>%query(termijnstappen_mddc_getinvoerlabeldropd,%keypointer%)%</label>
<divwidth>400</divwidth>
<divheight>30</divheight>
<edit>true</edit>
<showhint>>false</showhint>
<wavetype>keuzelijst</wavetype>
<source>generalwithoutemptyline</source>
<filter>%query(termijnstappen_mddc_getinvoerquerydropd,%keypointer%)%</filter>
<refresh>>false</refresh>
<backcolor/>
<fontcolor/>
<nullable>>true</nullable>
<icoon/>
<visible>%query(termijnstappen_mddc_isinvoerdropdvsvb,%keypointer%)%</visible>
</column>
```

Nullable

Een query kan worden gebruikt om de inhoud van de tag nullable contextgevoelig te maken.

Query's voor invoegen tekstblokken bij sjablonen

Zie ook kopje **Invoegen tekstblokken op basis van een query-aanroep naar tbqueries** bij [Documentjsjablonen](#). De query wordt in het documentsjabloon wordt aangeroepen door met de merge-codering `<%query(param1,param2)%>`.

Bijvoorbeeld: `<%query(apvoverig_tkstblk1,:keyvergunning)%>`.

Waarschuwing

De hier gebruikte merge-codering zal niet door de spellingschecker heenkomen, voor een goede werking zal het sjabloon daarom opgeslagen moeten zijn zonder spellingschecker!

Param1 wordt opgezocht in de kolom dvcode van tbqueries. Die query wordt geëvalueerd waarbij de string {id} in de query - indien aanwezig - eerst wordt vervangen door de waarde van param2. Param2 is dus optioneel.

Param2 zal meestal gevuld worden met de dnkey van de kaart van waaruit het document wordt gecreëerd. In het sjabloon wordt dit opgegeven door param2 één van de volgende waarden te geven (er zijn geen andere mogelijkheden voor param2):

- :keyvergunning (staat voor de dnkey van een zaak: tbovergunningen, tbomgvergunning, tbmilvergunningen, tbhandhavingen, tbbouwvergunningen, tbhorecavergunningen en

tbinfoaanvragen)

- :keyadvies
- :keyinspectie
- :keyinspectiebezoek
- :keyinrichting (staat voor de dnkey van de inrichting)
- :keyadres (staat voor de dnkey van een aangewezen contactpersoon)
- :keybezwaarberoep
- :keylocatie (staat voor de dnkey van de locatie van de zaak/inrichting).

Voorbeeld 1: invoegen tekst uit query zelf

Invoegen tekst uit query

De verwijzing in het sjabloon kan zijn: <%query(apvoverig_tkstblk1,:keyvergunning)%>.

De query uit tbqueries waarnaar wordt verwezen via apvoverig_tkstblk1 kan zijn:

```
select case when (select dvcodesoortovverg from tbovvergunningen where dnkey
= {id}) = 'O'
      then 'Volgens artikel X van wet y zal de ontheffing ....'
      else ''::text
end
```

Hetgeen betekent dat indien het zaaktype van de APV/Overige vergunning waar vandaan het document wordt gecreëerd van het soort 'O' is dat alleen dan de tekst *Volgens artikel X van wet y zal de ontheffing* zal worden ingevoegd.

Voorbeeld 2: invoegen tekst uit de tabel tekstblokken via de query

Invoegen tekst uit tekstblokken

```
select dvtekstblok from tbtekstblokken where lower(dvcode) =
      case when (select dvcodesoortovverg from tbovvergunningen
where dnkey = {id}) = 'O'
      then 'tkstblk_1'
      when (select dvcodesoortovverg from tbovvergunningen
where dnkey = {id}) = 'E'
      then 'tkstblk_2'
      else 'xxx'
end
```

Hetgeen betekent dat indien het zaaktype van de APV/Overige vergunning waar vandaan het document wordt gecreëerd van het soort 'O' is dat dan de inhoud van de kolom dvtekstblok uit de tabel tbtekstblokken (beheer) met dvcode = *tkstblk_1* wordt afgedrukt.

Indien soort = 'E' wordt de inhoud van de kolom dvtekstblok met dvcode = *tkstblk_2* afgedrukt. En

anders niets.

Zie voor invoegen plaatjes op basis van query onder Kopje: *Invoegen plaatjes op basis van een query-aanroep naar tbqueries die verwijst naar tbimages* bij [Documentsjablonen](#).

Query's als vervanging voor formqueries en childqueries bij definitie document- en emailsjablonen

De inhoud van de kolommen van de formqueries en childqueries uit de definitie document- en emailsjablonen (beheer) kan ook bestaan uit een verwijzing naar een query in deze tabel tbqueries. Hierdoor hoeft een query die in meerdere sjablonen gebruikt wordt maar eenmalig te worden gedefinieerd. De opmaak van de sjablonen wijzigt hierdoor niet. In tbqueries kan bovendien een select statement ingevoerd worden van onbeperkte grootte.

Zie het kopje *formquery en childquery-verwijzingen naar tbqueries* bij [Documentsjablonen en Sjabloongroepen](#)

From:
<https://doc.open-wave.nl/> - Documentatie

Permanent link:
https://doc.open-wave.nl/doku.php/openwave/1.33/applicatiebeheer/instellen_inrichten/queries?rev=1762184611

Last update: 2025/11/03 16:43

