

# Portalkolommen

## Applicatiebouw

Screenidentifiers: MDLC\_getPortalColumnsList.xml en MDDC\_getPortalColumnsDetail.xml

## Betekenis van de kolommen

- De **kolomnaam** mag aangepast worden, maar bij updatescripts van OpenWave versie x naar OpenWave versie y worden sommige kolomnamen verondersteld aanwezig te zijn om bijvoorbeeld nieuwe tegels onder te plaatsen. Indien die kolomnamen er niet meer zijn dan worden deze kolommen opnieuw aangemaakt met hun oorspronkelijke naam. Zo kunnen er doublures in kolommen ontstaan. Op zich niet erg, want de nieuwe tegels kunnen zonder gevaar verplaatst worden naar een andere kolom binnen hetzelfde portaal.
- De **kolombreedte** kan hier aangepast worden zodat iedere kolom in een portal een eigen breedte heeft. Houd rekening met de lengte van de kolom- en tegelnamen bij het opgeven van deze waarde. Word dit veld niet ingevuld, dan kijkt de programmatuur naar de opgegeven waarde van het veld "Kolombreedte" in het detailscherm van het portaal (MDDC\_getPortalNamesDetail.xml (miscellaneous)). De default waarde is 210.

### Waarschuwing

**Idealiter geen waardes die korter zijn dan de opgegeven kolom/tegel namen. Dit kan namelijk de leesbaarheid van het portaal scherm aantasten**

- Met de **kolomvolgorde** kan de volgorde van de kolommen horizontaal binnen een [portal](#) worden bepaald.
- Indien de kolom **Is kolom voor van buitenaf toevoegen nieuwe tegels** is aangevinkt dan is deze kolom bedoeld om van buiten af - bijvoorbeeld met een updatescript - tegels toe te voegen. De nieuwe tegels worden dan default onder deze aangevinkte kolom geplaatst. De beheerder kan ze dan desgewenst verplaatsen naar een andere kolom. Per [portalname](#) kan er maar één kolom deze eigenschap hebben. Dit wordt op de database afgevangen.

## Verplaatsen van kolom

Verplaatsen van kolommen van de ene portal naar de andere portal.

### Waarschuwing

Niet aan te raden

Eigenwijs: de eenvoudigste manier is om op het detailscherm van de kolom de waarde van *primary key van portalname* aan te passen (met een bestaande waarde).

Dit gaat echter mis wanneer in de actions van de onderliggende tegels de variabele {id} wordt

gebruikt. Die is namelijk echt gelieerd aan de primary key van de zaakportalen. Voorbeeld: een kolom bij portaal omgevingdetail waaronder een tegel geplaatst is met een action `getFlexList(TBADVIEZEN,TBOMGVERGUNNING,{id},O,W)` kan natuurlijk verplaatst worden naar het portaal APV/Overigdetail. De tegel verhuist dan mee, terwijl die niet gedefinieerd is voor dit portaal. Het indrukken van de tegel heeft tot gevolg heeft dat adviezen worden gezocht bij een omgevingszaak met de id van een APV/Overige zaak.

## Triggers onderaan detailscherm

- **controleer en toon responsetijd van onderliggende dynamische tegelopschriften** . Met de knop wordt een wizard gestart die de duur per aangeroepen query meet van de tegelopschriften van alle onderliggende tegels. OpenWave vraagt alvorens de metingen uit te voeren voor welke inlogger de eventuele substitutie van de variabele :keyaccount in de queries moet plaatsvinden en ook , indien het om een zaakportaal of inrichtingsportaal gaat, voor welke portalid (is de primary key - dnkey - van de zaak die de variabele {id} vervangt ) de queries uitgevoerd moeten worden. Indien deze laatste portalid niet wordt opgegeven gebruikt Openwave de niet bestaande dnkey met waarde 0

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

Permanent link:  
[https://doc.open-wave.nl/doku.php/openwave/1.33/applicatiebeheer/instellen\\_inrichten/portaldefinitie/portal\\_kolommen](https://doc.open-wave.nl/doku.php/openwave/1.33/applicatiebeheer/instellen_inrichten/portaldefinitie/portal_kolommen)

Last update: 2025/12/18 12:44

