

Actions

Wat is een action

Een action is een aanroep naar een methode van OpenWave met de juiste parameters. Die action wordt gekoppeld aan een trigger (zoals een knop, of een tegel of dubbelklikken op een lijstregel). OpenWave roept de methode aan en het gevolg is dat een lijst- of detailscherm wordt geopend, of een nieuw tabblad met een URL, of een wizard die wordt opgestart, of een kaartje dat getoond wordt, een memo.... et cetera.

Waar worden actions gedefinieerd

Op vele plekken in OpenWave kan een action aan een trigger worden gekoppeld. Bijvoorbeeld:

- in de xml van een schermkolomdefinitie van een detailscherm, waarbij een action wordt gedefinieerd bij een knop die binnen het scherm wordt opgenomen
- in de xml van een schermkolomdefinitie van een lijstschermb, waarbij een action wordt gedefinieerd bij een knop die binnen de lijst op elke regel wordt opgenomen
- in de tegeldefinitie (beheertegel *Portal*) waarbij de action wordt aangeroepen indien de gebruiker op de tegel drukt
- in een rapportdefinitie (beheertegel *Rapportagedefinitie*) waarbij de action gekoppeld wordt aan het dubbelklikken op de rapportageregel
- in de definitie van een proces-termijnstap (beheertegel *Processen*) waarbij de action gekoppeld wordt aan een knop bij een processtap
- in de definitie van lijst- en detailschermen via de beheertegel *Tabellen Standaardapi*, waarbij de actions kunnen worden gekoppeld aan:
 - het dubbelklikken op regel van lijstschermb
 - de knoppen van het lijstschermb en die van het detailschermb.

Syntax en substitutie van variabelen

In de meeste gevallen moet de action (methode + parameters) in één string worden gedefinieerd zoals: `geefZaakMemo(12234,W)` of `getFlexList(SysStandardList,nil,nil,G,beheer_tbdocumenttype)`. Alleen de actions gekoppeld aan knoppen van een standaardlijst of standaarddetail (dus via de beheertegel *Tabellen standaard API*) zijn opgesplitst in een kolom action, waarin dit keer alleen de methodenaam moet komen, en kolommen voor de parameters: elke parameter een eigen kolom.

OpenWave kan bij de aanroep van een action binnen die action-definitie de volgende substrings tegenkomen:

- `%query(querynaam)%` of `%query(querynaam,%keypointer)%`. In dat geval zal het SQL-statement van de betreffende query (beheertegel *Queries*) eerst worden geëvalueerd en het resultaat wordt gesubstitueerd in de action-definitie en daarna wordt de action uitgevoerd. Toepassing: schermdefinitie van de omgevingsdetailkaart met de knop *Ga naar OLO-loket* met de volgende action-definitie:

`openTabPage(https://www.omgevingsloket.nl/BevoegdGezag/bevoegdgezag/AanvraagTab/Aanvraag/%query\(omgeving\_olnummer,%keypointer%\)/AanvraagGegevens)`

- de substring `%keypointer%` kan alleen worden gesubstitueerd (altijd met de primary key van het record dat actief is) wanneer de action wordt aangeroepen vanuit een detailscherm
- de substring `%keyparent%` kan alleen worden gesubstitueerd (altijd met de primary key van het moederrecord dat actief is) wanneer de action wordt aangeroepen vanuit een lijst- of detailscherm dat als dochter is gedefinieerd
- `%keyaccount%` zal vervangen worden met de waarde van `tbmedewerkers.dvcode` van de inlogger
- `%inlogger%` met de waarde van `tbmedewerkers.dvcode` van de inlogger
- `{id}` kan alleen worden gebruikt bij een action die wordt uitgevoerd vanuit een lijstschermb (bijv. dubbelklikken op regel). De substring `{id}` wordt vervangen met de regel-id uit de lijst waar de gebruiker op dat moment op staat, die - meestal- de primary key bevat van de tabel waarop de lijst is gebaseerd. Toepassing: Zie het record in standaardapi-tabel met code *beheer_queries*.
- `{kolomnaam_uit_onderliggendeview/tabel}`. Werkt hetzelfde als `{id}` bij een action die wordt uitgevoerd vanuit een lijstschermb, bijv. `{dnkeyopslag}` waarbij deze string zal worden vervangen door de waarde van de kolom `dnkeyopslag` van de betreffende rij.

Welke OpenWave methodes kunnen worden gebruikt?

Voor de onderstreepte items geldt dat de substring `{id}` of `{kolomnaam}` in de action - indien aangeroepen van een lijstschermb - vooraf vervangen zal worden door de betreffende regel id (de waarde van de primary key van de regel) c.q. de waarde van de genoemde kolomnaam.

openTabPage

- Wat doet het?: Opent een nieuw tabblad met een kaart, externe website of een intern OpenWave portaal
- aanroep: `openTabPage(param1)`
 - Indien:
 - `param1` begint met de substring (http) dan wordt geacht dat `param1` een URL bevat bijvoorbeeld `https://www.open-wave.nl/`. Let op dat een ampersand-teken (&) in de URL geschreven moet worden als `&` (5 karakters)
 - `param1` bedoeld is een specifiek portaal binnen het domein van OpenWave te openen moet de portaalnaam en de dnkey worden doorgegeven bijv. `openTabPage(#omgevingdetail/x)` waarbij de `x` staat voor een specifieke dnkey van de hoofdtabel van het portaal. Naast `omgevingdetail` kunnen hier de portaalnamen `handhavingdetail`, `apvoverigdetail`, `infodetail`, `horecadetail` en `inrichtingdetail` en `milieugebruikdetail` worden gebruikt.
 - `param1` de waarde *#kaartoverzicht* heeft, dan zal OpenWave de interne kaart openen met de gedefinieerde kaartlagen uit de beheertabel *GEO kaartlagen*. Zie o.a. voor centrering [Kaart](#)
 - `param1` opgebouwd is als URI-aanroep voor openen of bewerken van een fileserver-document via een Microsoft-office pakket, dan zal OpenWave deze aanroep doorzetten in de URL-balk (met de juiste slashes) b.v. `ms-word:ofe|u|file:/oxygen/users/pdeboer/LocalFileLinksTest.docx`.

Waarschuwing

LET OP: indien openTabPage vanuit een tegelaction wordt aangeroepen kunnen de parameters niet via een query aanroep worden gesubstitueerd. Dat moet in dat geval iets ingewikkelder via een flexaction: bijvoorbeeld de action op de tegel is `getflexaction(omgeving_complex_oorsprong,{id})` waarbij de aangeroepen query (in dit voorbeeld `omgeving_complex_oorsprong`) een openTabPage-aanroep dient te construeren zonder te substitueren variabelen zoals: `select 'openTabPage(#omgevingdetail/' || dnkeyparentverg || ' ' from tbomgvergunning where dnkey = {id}`

getFlexList

- Wat doet het?: In een modal venster wordt een standaard lijstscherf over een view of tabel getoond. Zie: [Standaard Lijst- en Detailschermen](#)
- aanroep: `geefFlexList(param1, param2,param3,param4,param5):`
 - param1: De eerste parameter moet de waarde `SysStandardList` bevatten. Er zijn andere mogelijkheden maar die worden hier niet besproken
 - param2: De tweede parameter kan leeg blijven
 - param3: Indien de lijst een dochtertabel is van een andere tabel EN alleen die kaarten die horen bij een specifieke moederkaart getoond moeten worden, dan moet hier de primary key waarde komen van die moederkaart. Anders, de lijst wordt zelfstandig in een modal getoond zonder moeder, dan kan deze parameter leeg blijven
 - param4: Heeft de waarde leeg, G of A. De lijst wordt met G opgestart met alle niet vervallen kaarten. Met A wordt de lijst gestart met alle kaarten. Het programma kijkt daarbij naar de `vervallen-box kolomnaam` in de bijbehorende definitie van standaardapi-tabel
 - param5: De vijfde parameter moet verwijzen naar een dvcode uit de standaardapi-tabel (beheertegel `Tabellen Standaardapi`). Aldaar haalt OpenWave alle informatie op over de lijst (inhoud, lay-out, autorisatie en triggers)
- autorisatie: Wordt in de aangewezen kaart van de standaardapi-tabel geregeld
- voorbeeld aanroep lijst binnen een detailscherm van moeder:
`getFlexList(SysStandardList,,%keypointer%,,beheer_kopcompngem)`
- voorbeeld aanroep vanuit action op tegel zelfstandig lijstscherf
`getFlexList(SysStandardList,,,,beheer_compartiment).`

getFlexDetail

- Wat doet het?: In een modal venster wordt een standaard detailscherm over een view of tabel getoond. Zie: [Standaard Lijst- en Detailschermen](#)
- aanroep: `geefFlexDetail(param1, param2,param3):`
 - param1: De eerste parameter moet de waarde `SysStandardDetail` bevatten. Er zijn andere mogelijkheden maar die worden hier niet besproken
 - param2: De tweede parameter bevat de dnkey van de kaart waarvan het detailscherm wordt opgeroepen
 - param3: De derde parameter moet verwijzen naar een dvcode uit de standaardapi-tabel (beheertegel `Tabellen Standaardapi`). Aldaar haalt OpenWave alle informatie op over het

detailscherm (inhoud, lay-out, autorisatie en triggers)

- autorisatie: Wordt in de aangewezen kaart van de standaardapi-tabel geregeld
- voorbeeld aanroep bij trigger dubbelklikken op rij in een lijst:
getFlexdetail(SysStandardDetail,{id},beheer_compartiment).

getFlexAction

- Wat doet het?: Handig bij knoppen op lijsten, waarbij de action pas gedefinieerd kan worden nadat de gebruiker een regel actief heeft gemaakt. De uiteindelijk uit te voeren action wordt met deze methode eerst opgehaald uit een gedefinieerde query, waarbij vooraf de substring {id} van die query wordt gesubstitueerd door de waarde van param2. Indien param2 de waarde {id} heeft dan zal OpenWave deze vervangen door de identifier van de betreffende regel: meestal de dnkey van de tabel (de methode moet dan vanaf bijv. een schermknop op een lijst worden aangeroepen)
- aanroep: getFlexAction(param1, param2):
 - param1: De eerste parameter moet een bestaande dvcode uit de tabel tbqueries bevatten
 - param2: De tweede parameter is {id} of bevat een waarde waarmee de substring {id} van die query wordt gesubstitueerd
- autorisatie: Wordt geregeld in de autorisatiekolommen van tbqueries en anderzijds - indien mogelijk - door de API die door de uiteindelijke action wordt aangeroepen
- voorbeeld aanroep bij tag <action> van een schermknop in lijst:
getFlexAction(testAction,{id}). De query met dvcode = testAction kan bijvoorbeeld zijn:

```
select 'openTabPage(' | | dvhyperlink | | ' )' from tbtermijnbewstappen
where dnkey = {id}
```

geefGeoVanLokatie

- Wat doet het?: In een modal venster wordt een kaart getoond op basis van de coördinaatgegevens van een detailkaart. Zie [Kaart](#)
- aanroep: geefGeoVanLokatie(param1, param2):
 - param1: De eerste parameter moet de een primary key zijn van een tabel die hoort bij de tweede parameter. Met uitzondering indien param2 = *AlgemeneKaart*. In dat laatste geval kan param1 een lege waarde hebben
 - param2: De tweede parameter is OF een tabelnaam OF de waarde *AlgemeneKaart*. De tabelnaam moet één van volgende zijn: tbperceeladressen, tbmilinrichtingen, tbmildiversen, tbmilemlucht, tbmilemwater, tbmilopslag, tbhorontheffingen, tbhandhavingen, tbovvergunningen, tbomgvergunning, tbmilasbest, tbmilvergunningen, tbbouwvergunningen, tbinfoaanveragen, tbhorecavergunningen, tbmilstal of tbzaakdperc of tbmilafvalstoffen
 - autorisatie: Niet van toepassing
- Voorbeeld: *geefGeoVanLokatie(%keypointer%,tbperceeladressen)* aangeroepen vanuit een knop op detailscherm.

getFlexMemo

- Wat doet het?: In een modal venster wordt een memo (de kolom dvmemo) getoond
- aanroep: `getFlexMemo(param1, param2)`:
 - param1: De eerste parameter moet de primary key zijn van de tabel die hoort bij de tweede parameter
 - param2: De tweede parameter is de letter B (tbbouwvergunningen), C (tbhorecavergunningen), E (tbmilvergunningen), H (tbhandhavingen), I (tbinfoaanvragen) O (tbovvergunningen) of W (tbomgvergunning) of V (tbmilinrichtingen)
- autorisatie: De gebruiker moet memokijkrechten hebben voor de betreffende module en de compartimentsrechten moeten kloppen
- Voorbeeld: `getFlexMemo(%keypointer%,W)` aangeroepen vanuit een knop op detailscherm.

getFlexBalloon

- Wat doet het?: In een hint-venster (ballontekst) behorende bij de knop waarmee deze methode wordt aangeroepen wordt een tekst getoond. `<adm warning>LET OP: de tag refresh bij de knop moet de waarde false hebben (of leeg zijn)</adm>`
- aanroep: `getFlexBalloon(param1, param2)`:
 - param1: Een gecrypte tekst of een niet gecrypte tekst of een evalueerbare query. Afhankelijk van param2
 - param2:
 - D dan wordt de tekst in param1 in twee regels getoond in het ballonnetje. Eerste regel is param1 voorafgegaan door 'encrypt:'. De tweede regel is de gedecrypte versie van param1 voorafgegaan door 'decrypt'. (Zie [2-way encryptie van externe wachtwoorden](#))
 - P dan wordt param1 ongewijzigd in de tekstballoon getoond. Een semicolon (;) wordt daarbij geïnterpreteerd als harde return
 - QD dan bevat param1 een evalueerbare query die één regel en één kolom als resultaat teruggeeft, bijvoorbeeld: `select dvpas from tabelA where dnkey = %keypointer%`. De query moet beginnen met select en mag geen semicolons bevatten. De variabele %keypointer% wordt hierbij vervangen door de ID (dnkey) van de tabelkaart waar de gebruiker op staat. Nadat de query is geëvalueerd wordt het resultaat getoond als bij param1 = 'D'
 - QP dan bevat param1 een evalueerbare query die één regel en één kolom als resultaat teruggeeft, bijvoorbeeld: `select 'Let OP:' || dvnaam from tabelA where dnkey = %keypointer%`. De query moet beginnen met select en mag geen semicolons bevatten. De variabele %keypointer% wordt hierbij vervangen door de ID (dnkey) van de tabelkaart waar de gebruiker op staat. Nadat de query is geëvalueerd wordt het resultaat getoond als bij param1 = 'P'
- autorisatie: Niet van toepassing
- Voorbeeld: `getFlexBalloon(Hier alleen voorletters; zonder punt en spaties,P)`.

noAction

- Wat doet het?: Niets
- aanroep: `noAction()`

refreshActiveDialog

- Wat doet het?: Het scherm waarvandaan deze action wordt aangeroepen wordt opnieuw uitgeschreven
- aanroep: refreshActiveDialog()
- autorisatie: OpenWave kijkt opnieuw naar de rechten van het te overschrijven scherm.

refreshActiveDialog(parameterlist)

- Wat doet het?: Overschrijft het flexDetail- of flexListscherm waar je op staat, maar dan op basis van parameterlist. Bijvoorbeeld: vanuit het flexdetailscherm van de tabel tblegesregels is een knop gedefinieerd met de action: refreshActiveDialog(tblegesregels,13,W). De browser vraagt dan de detailgegevens op met getFlexDetail(tblegesregels,13,W) en overschrijft het bestaande detailscherm daarmee.

startWizard

- **startwizard(deleteContactadres,param2)**
 - Voorbeeld: startwizard(deleteContactadres,333)
 - param1: deleteContactadres
 - param2: De dnkeywaarde van de contactadreskaart die verwijderd moet worden. Indien param2 de waarde {id} bevat: de API wordt aangeroepen vanuit een lijst, dan wordt deze string {id} on the fly door OpenWave vervangen met deze primary key-waarde van de actieve kaart uit een lijst).
- **startwizard(deleteSysStandardRow,param2,param3,param4)**
 - Voorbeeld:
startwizard(deleteSysStandardRow,tbadressoort.{id},dvomschrijving,beheer_tbadressoort)
 - Aanroep van een standaard verwijderactie van een kaart van een tabel die gedefinieerd is in tbsysstandardtable (beheertegel *Tabellen Standaardapi*). Deze action kan bijv. aan een verwijderknop onder aan een lijst gekoppeld worden. De functie houdt rekening met de in de tbsysstandardbutton gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de tbsysstandardtable gedefinieerde blokkeringsvelden. De wizard geeft een waarschuwing indien de te verwijderen kaart komt uit tbomgvergunning of tbmilinrichtingen of tbhandhavingen of tbhorecavergunningen of tbovergunningen of tbinfoaanvragen of tbmilvergunningen m.b.t. fysieke documenten die niet mee verwijderd worden. De wavezaakcode van een verwijderde kaart uit een van deze tabellen kan opnieuw worden gebruikt
 - param1: deleteSysStandardRow
 - param2: De tabelnaam waaruit een kaart verwijderd moet worden gevolgd door een punt gevolgd door {id}. Die {id} wordt on the fly door OpenWave vervangen met primary key-waarde van de kaart die verwijderd moet worden (bijv. de actieve kaart uit een lijst)
 - param3: een kolomnaam uit de view of tabel die aan de lijst ten grondslag ligt, waarvan de achterliggende waarde gebruikt wordt voor de *weet u het zeker* tekst
 - param4: de code uit tbsysstandardtable die verwijst naar de kaart waar de betreffende standaardlijst in is gedefinieerd.
- **startwizard(insertContactadres)**
 - Aanroep vanuit een situatie dat een adres moet worden aangemaakt zonder deze via een

rol te koppelen aan een inrichting of zaak. De wizard vraagt om elementaire gegevens waarmee een nieuwe kaart wordt aangemaakt waarna vervolgens automatisch het detailscherm van de contactadreskaart wordt aangeroepen
`getFlexDetail(tbcontactadressen,denieuweaangemaaktednkey)`.

- **startwizard(insertSysStandardRow,param2,param3,param4)**

- Voorbeeld:
`startwizard(insertSysStandardRow,MDWC_insertTbMwTeams.xml,%keyparent%,beheer_tbmwteams)`
- Aanroep van een standaard insertactie van een kaart van een tabel die gedefinieerd is in `tbsysstandardtable` (beheertegel *Tabellen Standaardapi*). Deze action kan bijv. aan een insertknop onder aan een lijst gekoppeld worden. De functie houdt rekening met de in de `tbsysstandardbutton` gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de `tbsysstandardtable` gedefinieerde blokkeringsvelden
- param1: `insertSysStandardRow`
- param2: De naam van de screen.xml waarin de opmaak van het insertscherm is geregeld. De naam moet beginnen 'MDWC_'. De xml moet aan een aantal voorwaarden voldoen. Zie: [Scherm informatie voor standaard insert- en kopieer](#)
- param3: Alleen gevuld indien de tabel waarop een insert wordt uitgevoerd een parenttabel heeft. Hier wordt de betreffende keywaarde van die parenttabel ingevuld. De waarde `%keyparent%` betekent dat OpenWave dit zelf onder water regelt
- param4: de code uit `tbsysstandardtable` die verwijst naar de kaart waar de betreffende tabel in is gedefinieerd.

- **startwizard(kopieerSysStandardRow,param2,param3,param4)**

- Voorbeeld:
`startwizard(kopieerSysStandardRow,MDWC_insertTbMwTeams.xml,{id},beheer_tbmwteams)`
- Aanroep van een standaard insertactie van een kaart van een tabel die gedefinieerd is in `tbsysstandardtable` (beheertegel *Tabellen Standaardapi*). Deze action kan bijv. aan een insertknop onder aan een lijst gekoppeld worden. De functie houdt rekening met de in de `tbsysstandardbutton` gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de `tbsysstandardtable` gedefinieerde blokkeringsvelden
- param1: `kopieerSysStandardRow`
- param2: De naam van de screen.xml waarin de opmaak van het insertscherm is geregeld. De naam moet beginnen 'MDWC_'. De xml moet aan een aantal voorwaarden voldoen. Zie: [Scherm informatie voor standaard insert- en kopieer](#). Kan in veel gevallen dus gelijk zijn aan het scherm dat hoort bij de `insertStandardRow`
- param3: Wordt gevuld met de dnkey van de rij waar je op staat. Indien de kopieerknop onderaan een lijst staat kan `{id}` worden gebruikt, en anders, op een detailscherm, `%keypointer%`
- param4: de code uit `tbsysstandardtable` die verwijst naar de kaart waar de betreffende tabel in is gedefinieerd.

- **startwizard(koppelSysStandardRow,param2,param3,param4)**

- Voorbeeld:
`startwizard(kopieerSysStandardRow,MDWC_insertTbMwTeams.xml,{id},beheer_tbmwteams)`
- Aanroep van een standaard insertactie van een kaart van een tabel die gedefinieerd is in `tbsysstandardtable` (beheertegel *Tabellen Standaardapi*). Deze action kan bijv. aan een insertknop onder aan een lijst gekoppeld worden. De functie houdt rekening met de in de `tbsysstandardbutton` gedefinieerde rechten bij die knop en met het al of niet gevuld zijn van de in de `tbsysstandardtable` gedefinieerde blokkeringsvelden
- param1: `kopieerSysStandardRow`

- param2: De naam van de screen.xml waarin de opmaak van het insertscherf is geregeld. De naam moet beginnen 'MDWC_'. De xml moet aan een aantal voorwaarden voldoen. Zie: [Scherm informatie voor standaard insert- en kopieer](#). Kan in veel gevallen dus gelijk zijn aan het scherm dat hoort bij de insertStandardRow
- param3: Wordt gevuld met de dnkey van de rij waar je op staat. Indien de kopieerknop onderaan een lijst staat kan {id} worden gebruikt, en anders, op een detailscherf, %keypointer%
- param4: de code uit tbsysstandardtable die verwijst naar de kaart waar de betreffende tabel in is gedefinieerd.
- **startWizard(selecteerTaak,param2)** waarbij een wizard wordt gestart om een selectie te maken van openstaande taken op medewerker(s), modules en taaksoorten. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(maakDocument,param2,param3,param4)** waarbij een wizard wordt gestart teneinde een documentsjabloon aan te wijzen. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#).
- **startWizard(maakEmail,param2,param3,param4)** Idem als documentsjabloon, maar dan voor e-mails.
- **startWizard(maaknieuwelnrichting,param2)** waarbij een wizard wordt gestart teneinde een nieuwe inrichting te definiëren. Param2 kan leeg zijn. In dat geval wordt de wizard geopend, waarbij de gebruiker eerst gemeente, woonplaats en straat moet kiezen. Als param2 gevuld is verwacht OpenWave dat dit een dnkey uit de tabel TbOpenBareRuimte is. De inlogger zal dan alleen het adres binnen die straat moeten kiezen.
- **startWizard(maaknieuwproces,param2,param3)** waarbij een wizard wordt gestart teneinde een vervolgproces te kiezen vanuit de procesbewaking. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#).
- **startWizard(maaknieuwzaak,param2,param3,param4)** waarbij een wizard wordt gestart teneinde een nieuwe hoofdzaak te definiëren. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#) EN bij lemma [Aanmaken van nieuwe zaak](#).
- **startwizard(showTekst,param2,param3,param4)** waarbij een wizard wordt gestart van één scherm met alleen een sluitknop die de tekst uit param2 toont.
 - Voorbeeld: startWizard(showTekst, dit is een tekst,dit is de koptekst,400)
 - param1: showTekst
 - param2: de tekst die getoond wordt in het wizardscherf. Mag een lange tekst zijn
 - param3: de koptekst. Mag leeg zijn
 - param4: hoogte van tekstvak in pixles. Indien leeg dan is de default 120.
- **startwizard(sluitZaak,param2,param3,param4)** waarbij een wizard wordt gestart teneinde een hoofdzaak af te sluiten. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#) en het lemma [Sluiten van zaak](#).
- **startwizard(startreport,param2,param3)** waarbij een bepaald rapport wordt gestart (tbrapporten.dnkey = param2). Param3 mag een lege waarde hebben, maar indien gevuld dan moet het rapport aangeroepen worden vanuit een zaakportaal, waarbij param3 de id is van die hoofdzaak (dus bijv. een dnkey uit tbomgvergunning). Zie voorbeeld voor het gebruik van deze param3 identifier van zaakportal (nportalid) in [Rapportages](#).
- **startWizard(zoeKinrichtingopNaam,param2)** waarbij een wizard wordt gestart teneinde een inrichting te zoeken. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan. Bij 0 wordt de wizard gesloten.
- **startWizard(ZoekZaakViaZaaknummer,param2)** waarbij een wizard wordt gestart teneinde een zaak te zoeken op zaakcodering. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(ZoekZaakViaAdres,param2)** waarbij een wizard wordt gestart teneinde een

zaak te zoeken op adres. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.

- **startWizard(ZoekZaakViaBetreftDatum,param2)** waarbij een wizard wordt gestart teneinde een zaak te zoeken op omschrijving of datums. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(ZoekZaakViaContact,param2)** waarbij een wizard wordt gestart om een zaak te zoeken op contactpersoon. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(ZoekInspectieViaZaaknummer,param2)** waarbij een zoekwizard wordt gestart om een inspectiezaak te zoeken op een zaakcodering. De waarde 1 bij param2 geeft aan dat de zoekwizard blijft staan.
- **startWizard(StuurDSOOntvangstbevestiging,param2,param3,param4)** waarbij een wizard wordt gestart om een DSO ontvangstbevestigingsmail te versturen.
 - param1: StuurDSOOntvangstbevestiging
 - param2: Primary key van tabel genoemd bij param3. Dnkey van tbomgvergunning of tbomgdsoaanvullintrek
 - param3: Naam van de tabel waarvoor na genereren de verstuurdatum gevuld moet worden. Moet gevuld zijn en of waarde *tbomgvergunning* (voor DSO initieel) of *tbomgdsoaanvullintrek* (voor DSO aanvulling) hebben
 - param4: optioneel, indien gevuld dan dnkey van processtap (tbtermijnbewstappen) die moet worden afgesloten. Let op als param4 gevuld dan mag param3 alleen waarde *tbomgvergunning* hebben: DSO ontvangstbevestiging Aanvulling versturen vanaf processtap kan niet. Er kunnen immers meer dan 1 aanvullingen zijn, dnkey is niet bekend bij termijnbewakingsstappen. Zie uitgewerkte voorbeelden onder kopje action bij [Termijnstappen](#).

Action column:kolomnaam

Heeft dezelfde functie als getFlexAction. Alleen wordt de uiteindelijke uit te voeren actie NIET opgehaald uit een query, maar uit een kolom van de view die aan de lijst ten grondslag ligt.

De tag <action> bij de knop wordt zonder verdere parameters gevuld wordt door de vaste tekst 'column:' die direct gevolgd wordt door een kolomnaam uit de betreffende lijst. De waarde van die kolomnaam bevat de action die uitgevoerd wordt bij het indrukken van de bewuste knop. Het gaat dan om de waarde uit de actieve regel van die lijst. In zo'n kolom staat dus bijv. als waarde: *geefGeovanLokatie(1234,tbperceeladressen)* en in diezelfde lijst bij een volgende regel *geefGeovanLokatie(5678,tbperceeladressen)*.

Toepassing: zie beheertegel *Tabellen Standaardapi* en zoek de kaart met dvcode = *opening_vwfrmtevolgenzaken*.

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

Permanent link:
https://acc-doc.open-wave.nl/doku.php/openwave/1.30/applicatiebeheer/instellen_inrichten/actions?rev=1707312875

Last update: 2025/05/26 12:29