

# REST Flexfuncties

Portal beheerportaal-Nieuw. Tegel *Rest Flexfuncties*.

Screenidentifiers (beheertegel *Schermkolomdefinitie standaardapi*)

- MDLC\_getRestflexfunctiesList.xml
- MDDC\_getRestflexfunctiesDetail.xml
- MDWC\_insertTbRestflexfuncties.xml

Tabel: tbrestflexfuncties.

## Omschrijving

In deze tabel kunnen bestaande API's van lijst- en detailschermen en wizards worden benoemd die via de OpenWave REST-service door een derde partij - veilig - kunnen worden aangeroepen.

### Aanroep van REST-service door derde partij

Bijvoorbeeld: REST-rommeldam.open-wave.nl/TrackNTraceOmgBasis/12345

Hierbij is *TrackNTraceOmgBasis* de gevraagde REST-functienaam en *12345* is het eerste en enige argument. De OpenWave REST-service moet apart geïnstalleerd worden naast de OpenWave-applicatie.

### REST-service -> API `translateRestCall`

De tabel `tbrestflexfuncties` wordt geraadpleegd door de OpenWave API: `translateRestCall`.

Deze API wordt door de OpenWave REST-service aangeroepen met de REST-functienaam (paramcode) en een of meer argumenten (paramparameters). Deze functienaam en argumenten komen uit de aanroep waarmee de OpenWave REST-service door een derde partij wordt benaderd, zoals in bovenstaand aanroepvoorbeeld.

De API zoekt op basis van die argumenten een kaart in de tabel `tbrestflexfuncties` alwaar verwezen wordt naar een bepaalde flexlijst of flexdetail of flexwizard van OpenWave, waarbij de substrings `{arg1}`, `(arg2)`, `{argn}` vervangen worden door de argumenten van de API-aanroep.

De API retourneert deze aanroep naar de gevonden flexdetail, flexwizard of flexlijst - met de juiste parameters - als string naar de OpenWave REST-service. De REST-service roept vervolgens de betreffende flexlist-API, flexdetail-API of flexwizard-API aan en krijgt als resultaat een xml met daarin de gevraagde data en scherm informatie op basis van de argumenten. Deze xml wordt door de OpenWave REST-service doorgezet naar de derde partij die de OpenWave REST-service aanriep.

Deze derde partij moet de xml met scherm- en data-informatie zelf renderen naar een webpagina.

## Rendering door derde partij naar webpagina

De OpenWave REST-functies retourneren een xml met daarin de gevraagde data, maar waarin ook een representatie van een lijst, of van een detailscherm of een wizardscherm. Voor elke wizard, lijst of detail is de opbouw van de xml gelijk. Er is in elk geval een blok metadata met de gewenste schermopbouw informatie, en een blok data. De derde partij is vrij om alleen de data te gebruiken. De opbouw en xsd's zijn bij OpenWave opvraagbaar.

## Authenticatie

Bij de installatie van de OpenWave REST-service wordt naam en pass afgesproken waarmee de REST-service de OpenWave API's kan aanroepen. De naam en het wachtwoord zullen ook in de medewerkerstabel moeten zijn opgenomen en de vereiste rechten zullen aan die robotmedewerker moeten worden toegekend.

Voor de buitenwereld is het wachtwoord dat de REST-service gebruikt NIET zichtbaar.

## Parameters

De API `translateRestCall` wordt door OpenWave REST-service aangeroepen met de volgende parameters:

- `paramsessie` (deze wordt verkregen door de aanroep van `getauthenticatie` (door de REST-service) op basis van afgesproken naam en wachtwoord)
- `paramretourformaat` (XML)
- `paramcode`. Dit is de REST-functienaam. De waarde van deze parameter wordt opgezocht in de tabel `tbrestflexfuncties` in de kolom `dvcode`
- `paramparameters`. Dit is een string van een of meer argumenten. De substrings `{arg1}`, `{arg2}` `{argn}` die voorkomen in de kolom `dvaanroep` van de gevonden kaart in `tbrestflexfuncties` worden vervangen met de waarde van de argumenten (`{arg1}` met het eerste, `{arg2}` met de tweede enz.).

Het kan zijn dat een of meer argumenten gecrypt zijn doorgegeven. In dat geval moet in de kolom `dvaanroep` van `tbrestflexfuncties` het betreffende argument worden genoteerd als `{strdecrypt(argn)}`. Een voorbeeld is het gecrypt uitgeven van een identifier van de omgevingszaak in een ontvangstbevestiging wanneer in het sjabloon de tekst `<%strEncrypt(:dnkey)%>` wordt opgenomen. Deze gecrypte key dient als basis voor de aanroep van de track and trace in onderstaand voorbeeld.

## Uitgewerkt voorbeeld Track and Trace

Een aanvrager kan via een ontvangstbevestiging een track and trace link krijgen. Deze link zal uit drie delen bestaan:

- De URL van de derde partij. Bijvoorbeeld: `REST-rommeldam.open-wave.nl/`
- Als het om een omgevingsvergunning gaat, de code: `TrackNTraceOmgBasis/` (zie onder)

- De geëncrypte key van de gevraagde omgevingszaak (zie [Documentjablonen en Sjabloongroepen](#) onder kopje Tonen gecrypte versie van een kolomwaarde).

Dit maakt dat de link die bijgeleverd wordt in de brief naar de aanvrager er als volgt uit zal zien: [REST-rommeldam.open-wave.nl/TrackNTraceOmgBasis/02\\_RT125678J](https://rest-rommeldam.open-wave.nl/TrackNTraceOmgBasis/02_RT125678J)

Een derde partij maakt een webpagina waarmee op basis van deze link de stand van zaken met betrekking tot de behandeling van deze zaak getoond wordt.

Deze derde partij moet informatie halen uit de OpenWave database en wel: informatie van de basiskaart (detailscherm informatie), van de adviezen (een lijst) en informatie van het proces (ook een lijst).

Zij roept daartoe DRIE maal de OpenWave REST-service aan:

- [REST-rommeldam.open-wave.nl/TrackNTraceOmgBasis/02\\_RT125678J](https://rest-rommeldam.open-wave.nl/TrackNTraceOmgBasis/02_RT125678J) waarbij *02\_RT125678J* het eerste argument is en staat voor de geëncrypte dnkey van de gevraagde omgevingszaak
- [REST-rommeldam.open-wave.nl/TrackNTrace\\_AdviezenOmg/02\\_RT125678J](https://rest-rommeldam.open-wave.nl/TrackNTrace_AdviezenOmg/02_RT125678J)
- [REST-rommeldam.open-wave.nl/TrackNTrace\\_ProcesstappenOmg/02\\_RT125678J](https://rest-rommeldam.open-wave.nl/TrackNTrace_ProcesstappenOmg/02_RT125678J)

In de JSON-resultaat set van de eerste detail-aanroep is de plek opgenomen waar de twee lijsten op de webpagina geplaatst moeten worden (zie verderop).

Bij de installatie van de REST-service is afgesproken onder welke naam en wachtwoord de OpenWave API's door de REST-Service worden aangeroepen.

De applicatiebeheerder van OpenWave moet nu het volgende doen:

- een robotmedewerker aanmaken met een afgesproken naam en wachtwoord en deze net voldoende functionele rechten geven om de gevraagde informatie te lezen. De robot in dit voorbeeld zal daartoe deel uit moeten maken van een rechtengroep (bijv. Robots) die de module kijkrechten heeft op Omgevingszaken en daarbinnen de vinkjes *zichtbaar* bij *Adviezen* en *Proces checklijst*.

Verder moet deze robotmedewerker in de medewerkerstabel:

- een interne medewerker zijn;
- EN toegang moet hebben tot de browserversie
- EN *Voor deze medewerker is geen 2-factor authentication nodig* is aangevinkt;
- EN *Het password van deze medewerker verloopt nooit* is aangevinkt;
- EN *Deze medewerker hoeft geen loginverklaring af te vinken* is aangevinkt.

Vervolgens moet de applicatiebeheerder:

- in de tabel `tbrestflexfuncties` drie kaarten aanmaken met de gevraagde REST-functies (de namen van die functies zijn vrij voor de applicatiebeheerder in zoverre dat zij overeen moeten komen met de REST aanroepen van de derde partij)
- een kaart met `dvcode = TrackNTraceOmgBasis` waarbij de waarde van kolom `dvaanroep` bijvoorbeeld kan zijn:  
`getFlexDetail(SysStandardDetail,{strdecrypt(arg1)},TrackNTrace_BasisOmg)`
- een kaart met `dvcode = TrackNTrace_AdviezenOmg` waarbij de waarde van kolom `dvaanroep` bijvoorbeeld kan zijn: `getFlexList(SysStandardList,,{arg1},,TrackNTrace_AdviezenOmg)`
- een kaart met `dvcode = TrackNTrace_ProcesstappenOmg` waarbij de waarde van kolom

dvaanroep bijvoorbeeld kan zijn:

```
getFlexList(SysStandardList,,{arg1},,TrackNTrace_ProcesstappenOmg).
```

In bovenstaand voorbeeld wordt met `getflexdetail(sysstandarddetail,n,TrackNTrace_BasisOmg)` verwezen naar een kaart in de tabel `tbsysstandardtable` (beheertegel: *Tabellen Standaardapi*) met `dvcode = TrackNTrace_BasisOmg`. Aldaar is geregeld welke data van welke tabel/view en scherm moeten worden opgehaald. En zo wordt ook verwezen naar een kaart in `tbsysstandardtable` met `dvcode = TrackNTrace_AdviezenOmg`. En zo wordt ook verwezen naar een kaart in `tbsysstandardtable` met `dvcode = TrackNTrace_ProcesstappenOmg`.

De applicatiebeheerder gaat in `sysstandardtable` zelf bepalen welke track en trace data worden opgehaald. En de applicatiebeheerder gaat ook de schermlayout definiëren waarin deze track and trace kunnen worden getoond.

## Detailgegevens van de omgevingszaak

De applicatiebeheerder zal als volgt een `sysstandardtable` kaart aanmaken voor tonen van de detailgegevens van de omgevingszaak:

- **Code:** `TrackNTrace_BasisOmg`
- **Hoofdtabel of viewnaam:** `vwfrmomgvergunningen`
- **Kolomnaam van de primary key:** `dnkeyomgvergunning`
- **Tabelnaam waarop hoofdtabel/view is gebaseerd:** `tbomgvergunning`
- **Schermidentificer voor detail:** `MDDC_getTrackNTrace_OmgDetail.xml`
- **Tbqueries.dvcode voor kijkrechten:** `omgeving_trackntrace`

De overige kolommen van deze kaart kunnen leeg blijven.

De API `getflexdetail` zal de data van de kaart ophalen uit `vwfrmomgvergunningen where dnkeyomgvergunning = strdecrypt(arg1)}` (dus in ons voorbeeld `strdecrypt(02_RT125678J)`).

Daartoe wordt eerst een rechtencheck gedaan door het statement uit `tbqueries` (beheerportaal-Nieuw) met `dvcode = omgeving_trackntrace` te evalueren. Die moet dus bestaan. Het SQL-statement:

```
select case when a.dlaomgvsb = 'T' and b.dlcomgadvvsb = 'T' and
b.dlcomprovsvb = 'T'
      then 'true' else 'false' end
      from tbomgrechten b inner join tbrechten a
      on (a.dnkey = b.dnkeyrechten)
      where b.dnkeyrechten =
      (select dnkeyrechten from tbmedewerkers where trim(dvcode) =
      trim(:keyaccount))
```

De (robot)loginnaam die de REST-service gebruikt moet dus zowel het kijkrecht op omgevingszaken hebben, als het kijkrecht op adviezen bij een omgevingszaak als het kijkrecht op processtappen bij een omgevingszaak.

Met de schermidentificer voor detail wordt verwezen naar een kaart in `tbscreencolumns` (beheertegel *Schermkolomdefinitie tabellen standaard-api*) met `xml-filename =`

*MDDC\_getTrackNTrace\_OmgDetail.xml*. Als die kaart niet bestaat moet deze aangemaakt worden met:

- klasse: sysStandard
- API: getSysStandardDetail
- view/tabel: vwfrmomgvergunningen
- identifier scherm: xml-filename : MDDC\_getTrackNTrace\_OmgDetail.xml
- kopegel1: select dvobjadres | | dvobjplaats from vwfrmomgvergunningen where dnkeyomgvergunning = {id}

In de kolom *Kolominformatie Toggle F11* moet vervolgens de layout van het scherm gedefinieerd worden in xml-formaat met een of meer kolommen uit vwfrmomgvergunningen.

Een zeer karig voorbeeld:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <!--schermdata voor sysstandardapi met code TrackNTrace_BasisOmg-->
  <!--element tagnaam verwijst naar de kolomnamen van vwfrmomgvergunningen
-->
  <screenwidth>900</screenwidth>
  <columns>
    <blok>
      <label>Zaak</label>
      <width>100</width>
      <height>130</height>
      <type>doorlopend</type>
      <column>
        <regel>1</regel>
        <tagnaam>dvaanvraagnaam</tagnaam>
        <label>Korte omschrijving</label>
        <divwidth>300</divwidth>
        <divheight>30</divheight>
        <edit>>false</edit>
        <showhint>>false</showhint>
        <wavetype>string</wavetype>
        <source/>
        <filter/>
        <refresh>>false</refresh>
        <backcolor/>
        <fontcolor/>
        <nullable>>false</nullable>
        <icoon/>
      </column>
      <column>
        <regel>1</regel>
        <tagnaam>dvsoortaanvraag</tagnaam>
        <label>Soort aanvraag</label>
        <divwidth>250</divwidth>
        <divheight>30</divheight>
        <edit>>false</edit>
```

```
<showhint>true</showhint>
<wavetype>keuzelijst</wavetype>
<source>tbsoortomgverg</source>
<filter/>
<refresh>true</refresh>
<backcolor/>
<fontcolor/>
<nullable>>false</nullable>
<icoon/>
</column>
<column>
  <regel>2</regel>
  <tagnaam>ddaanvraag</tagnaam>
  <label>Aanvraagdatum</label>
  <divwidth>100</divwidth>
  <divheight>30</divheight>
  <edit>>false</edit>
  <showhint>>false</showhint>
  <wavetype>datum</wavetype>
  <source/>
  <filter/>
  <refresh>true</refresh>
  <backcolor/>
  <fontcolor/>
  <nullable>>false</nullable>
  <icoon/>
</column>
<column>
  <regel>2</regel>
  <tagnaam>dvstatus</tagnaam>
  <label>Status</label>
  <divwidth>140</divwidth>
  <divheight>30</divheight>
  <edit>>false</edit>
  <showhint>>false</showhint>
  <wavetype>string</wavetype>
  <source/>
  <filter/>
  <refresh>>false</refresh>
  <backcolor/>
  <fontcolor>red</fontcolor>
  <nullable>>true</nullable>
  <icoon/>
</column>
<column>
  <regel>3</regel>
  <tagnaam>dvzaakcode</tagnaam>
  <label>0W zaakcode</label>
  <divwidth>140</divwidth>
  <divheight>30</divheight>
```

```

        <edit>false</edit>
        <showhint>false</showhint>
        <wavetype>string</wavetype>
        <source/>
        <filter/>
        <refresh>false</refresh>
        <backcolor/>
        <fontcolor/>
        <nullable>false</nullable>
        <icoon/>
    </column>
    <column>
        <regel>3</regel>
        <tagnaam>dvlvoaanvraagnr</tagnaam>
        <label>OLO-nummer</label>
        <divwidth>140</divwidth>
        <divheight>30</divheight>
        <edit>false</edit>
        <showhint>false</showhint>
        <wavetype>string</wavetype>
        <source/>
        <filter/>
        <refresh>true</refresh>
        <backcolor/>
        <fontcolor/>
        <nullable>true</nullable>
        <icoon/>
    </column>
</blok>
    <blok>
        <label>Adviezen</label>
        <width>100</width>
        <height>160</height>
        <type>{REST}/TrackNTrace_Adviezenomg/%keypointer%</type>
    </blok>
    <blok>
        <label>Proces</label>
        <width>100</width>
        <height>160</height>
        <type>{REST}/TrackNTrace_Processtappen0mg/%keypointer%</type>
    </blok>
</columns>
</document>

```

Met de blokken label *Adviezen* en *Proces* wordt aangegeven op welke plek in het detailscherm de lijsten moeten worden ingevoegd. Dit is dus van belang voor de derde party die de gegevens van drie REST-aanroepen moet renderen naar een webpagina.

De API getflexdetail maakt een xml met data en metadata op basis van de bovenstaande schermlayout en de gegevens van de gewenste kaart van vwfrmomgvergunningen. De REST-service

stuurt deze informatie naar de derde partij die vervolgens de twee vervolgaanroepen doet naar de REST-service.

De REST-functies aanroepen voor de twee lijsten hebben een parameter die hier in de blokken gevuld worden met de waarde van de dnkey van de omgevingszaak waar de adviezen en termijnstappen aan zijn verbonden (dat doet de variabele %keypointer%). De {arg1} bij de definitie van de RestFlex-functies wordt hiermee gevuld bij uitvoering.

## Adviezenlijst bij een omgevingszaak

De applicatiebeheerder zal als volgt een sysstandardtable kaart aanmaken voor tonen van de lijstgegevens van de adviezen bij een omgevingszaak:

- **Code:** TrackNTrace\_AdviezenOmg
- **Hoofdtabel of viewnaam:** vwfrmadviezen
- **Kolomnaam van de primary key:** dnadvkey
- **Tabelnaam waarop hoofdtabel/view is gebaseerd:** tbadviezen
- **Kolomnaam foreign key uit deze achterliggende tabel:** dnkeyomgvergunningen
- **Kolomnaam foreign key (uit hoofdtabel/view):** dnkeyomgvergunningen
- **Parenttabelnaam:** tbomgvergunning
- **Schermidentificer voor detail:** MDLC\_getTrackNTrace\_AdviezenOmgList.xml
- **Tbqueries.dvcode voor lijst:** omgeving\_trackntrace
- **where clause bij lijst:** ddvervallen is null

De overige kolommen van deze kaart kunnen leeg blijven.

De API getflexlist zal de data van de advieskaarten ophalen uit vwfrmadviezen where dnkeyomgvergunningen = {arg1} (dus in ons voorbeeld bijv. 2345 (de %keypointer%)). Hierbij worden alleen de kaarten met ddvervallen is null meegenomen.

Allereerst wordt een rechtencheck gedaan door het statement uit tbqueries (beheerportaal-Nieuw) met dvcode = *omgeving\_trackntrace* te evalueren. Die moet dus bestaan (zie hierboven).

Met de schermidentificer voor lijst wordt verwezen naar een kaart in tbscreencolumns (beheertegel *Schermkolomdefinitie tabellen standaard-api*) met xml-filename = *MDLC\_getTrackNTrace\_AdviezenOmgList.xml*. Als die kaart niet bestaat moet deze aangemaakt worden met:

- klasse: sysStandard
- API: getSysStandardList
- view/tabel: vwfrmadviezen
- identificer scherm: xml-filename: MDLC\_getTrackNTrace\_AdviezenOmgList.xml
- default sortering: ddadvaanvraag, ddadvadvies

In de kolom *Kolominformatie Toggle F11* moet vervolgens de layout van het scherm gedefinieerd worden in xml-formaat met een of meer kolommen uit vwfrmadviezen.

Een voorbeeld:

```
<document>
  <!--tagnaam slaat op een kolom uit vwfrmadviezen-->
  <column tagnaam="dadvinstantie">
    <label>Adviesinstantie</label>
    <index>1</index>
    <length>200</length>
    <wavetype>string</wavetype>
    <icoon/>
    <showhint>>false</showhint>
  </column>
  <column tagnaam="ddadvaanvraag">
    <label>Aangevraagd</label>
    <index>2</index>
    <length>100</length>
    <wavetype>datum</wavetype>
    <icoon/>
    <showhint>>false</showhint>
  </column>
  <column tagnaam="ddadvrappel">
    <label>Rappeldatum</label>
    <index>5</index>
    <length>100</length>
    <wavetype>datum</wavetype>
    <icoon/>
    <showhint>>false</showhint>
  </column>
  <column tagnaam="ddadvadvies">
    <label>Advies retour</label>
    <index>6</index>
    <length>120</length>
    <wavetype>datum</wavetype>
    <icoon/>
    <showhint>>false</showhint>
  </column>
  <column tagnaam="dadvpositief">
    <label>Resultaat</label>
    <index>7</index>
    <length>93</length>
    <wavetype>boolean+</wavetype>
    <icoon/>
    <showhint>>false</showhint>
  </column>
  <column tagnaam="dadvomschrijving">
    <label>Betreft</label>
    <index>9</index>
    <length>200</length>
    <wavetype>string</wavetype>
    <icoon/>
    <showhint>>false</showhint>
  </column>
```

</document>

## Processtappenlijst bij een omgevingszaak

De applicatiebeheerder zal als volgt een sysstandardtable kaart aanmaken voor de lijstgegevens van de processtappen bij een omgevingszaak:

- **Code:** TrackNTrace\_ProcesstappenOmg
- **Hoofdtabel of viewnaam:** vwfrmtermijnbewstappen
- **Kolomnaam van de primary key:** dnkey
- **Tabelnaam waarop hoofdtabel/view is gebaseerd:** tbtermijnbewstappen
- **Kolomnaam foreign key uit deze achterliggende tabel:** dnkeyomgvergunningen
- **Kolomnaam foreign key (uit hoofdtabel/view):** dnkeyomgvergunningen
- **Parenttabelnaam:** tbomgvergunning
- **Schermidentificer voor detail:** MDLC\_getTrackNTrace\_ProcesstappenOmgList.xml
- **Tbqueries.dvcode voor lijst:** omgeving\_trackntrace
- **where clause bij lijst:** dlingebruik = 'T' and dvtrmvoorwijn = 'N'

De overige kolommen van deze kaart kunnen leeg blijven.

De API getflexlist zal de data van de processtappen ophalen uit vwfrmtermijnbewstappen where dnkeyomgvergunningen = {arg1} (dus in ons voorbeeld bijv. 2345 (de %keypointer%)). Hierbij worden alleen de kaarten met dlingebruik = 'T' and dvtrmvoorwijn = 'N' meegenomen (alleen de termijnstappen die zichtbaar en muteerbaar zijn en niet de ja/nee vragen).

Allereerst wordt een rechtencheck gedaan door het statement uit tbqueries (beheerportaal-Nieuw) met dvcode = *omgeving\_trackntrace* te evalueren. Die moet dus bestaan (zie hierboven).

Met de schermidentificer voor lijst wordt verwezen naar een kaart in tbscreencolumns (beheertegel *Schermkolomdefinitie tabellen standaard-api*) met xml-filename = *MDLC\_getTrackNTrace\_ProcesstappenOmgList.xml*. Als die kaart niet bestaat moet deze aangemaakt worden met:

- klasse: sysStandard
- API: getSysStandardList
- view/tabel: vwfrmtermijnbewstappen
- identificer scherm: xml-filename : MDLC\_getTrackNTrace\_ProcesstappenOmgList.xml
- default sortering: dntrmvolgnr

In de kolom *Kolominformatie Toggle F11* moet vervolgens de layout van het scherm gedefinieerd worden in xml-formaat met een of meer kolommen uit vwfrmtermijnbewstappen.

Een voorbeeld:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <!--tagnaam slaat op een kolom uit vwfrmtermijnbewstappen gefilterd op
termijnstappen-->
  <column tagnaam="dntrmvolgnr">
```

```
<label>Volgnr</label>
<index>10</index>
<length>70</length>
<wavetype>integer</wavetype>
<icoon/>
<showhint>>false</showhint>
</column>
<column tagnaam="ddtrmdeadline">
  <label>Streefdatum</label>
  <index>20</index>
  <length>110</length>
  <wavetype>datum</wavetype>
  <icoon/>
  <showhint>>false</showhint>
</column>
<column tagnaam="ddtrmafgehandeld">
  <label>Afgehandeld</label>
  <index>30</index>
  <length>110</length>
  <wavetype>datum</wavetype>
  <icoon/>
  <showhint>>false</showhint>
</column>
<column tagnaam="dvtrmprocedurenaam">
  <label>Proces</label>
  <index>40</index>
  <length>150</length>
  <wavetype>string</wavetype>
  <icoon/>
  <showhint>>false</showhint>
</column>
<column tagnaam="dvtrmomschrijving">
  <label>Stap</label>
  <index>50</index>
  <length>300</length>
  <wavetype>string</wavetype>
  <icoon/>
  <showhint>>false</showhint>
</column>
</document>
```

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

Permanent link:  
[https://doc.open-wave.nl/doku.php/openwave/1.31/applicatiebeheer/instellen\\_inrichten/rest\\_flexfuncties?rev=1732597661](https://doc.open-wave.nl/doku.php/openwave/1.31/applicatiebeheer/instellen_inrichten/rest_flexfuncties?rev=1732597661)

Last update: 2024/11/26 06:07

