

# ZGW API Notificaties

De ZGW API standaard definieert ook de notificatie component (NRC). Dit is een component waarmee afnemers (consumers) zich op wijzigingen kunnen abonneren en waarop bronnen (ZRC, ZTC..) de wijzigingen kunnen aanmelden.

En consumer meldt zich als volgt aan:

```
{
  "callbackUrl": "https://zaken-api.vng.cloud/api/v1/callbacks",
  "auth": "Bearer aef34gh...",
  "kanalen": [{
    "naam": "zaken",
    "filters": {
      "bronorganisatie": "082096752011",
      "zaaktype": "https://example.com/api/v1/zaaktypen/5aa5c",
      "vertrouwelijkheidaanduiding": "*"
    }
  }, {
    "naam": "documenten",
    "filters": {
      "bronorganisatie": "082096752011",
      "informatieobjecttype": "https://example.com/api/v1/informatieobjecttype/b8c11",
      "vertrouwelijkheidaanduiding": "openbaar"
    }
  }
]}
```

De callbackUrl is de url van de afnemer. Hierop zal de consumer de notificaties krijgen. Auth geeft aan welke authorization header gebruikt moet worden. De kanalen geven aan op welke kanalen de afnemer zich abonneert eventueel aangevuld met extra filters zodat enkel die notificaties gestuurd worden die door de filters komen.

De notificaties zien er als volgt uit:

```
[{
  "kanaal": "zaken",
  "hoofdObject": "https://zaken-api.vng.cloud/api/v1/zaken/d7a22",
  "resource": "status",
  "resourceUrl": "https://zaken-api.vng.cloud/api/v1/statussen/d7a22/721c9",
  "actie": "create",
  "aanmaakdatum": "2018-01-01T17:00:00Z",
  "kenmerken": {
    "bron": "082096752011",
    "zaaktype": "https://example.com/api/v1/zaaktypen/5aa5c",
    "vertrouwelijkheidaanduiding": "openbaar"
  }
}]
```

De onderliggende reference implementatie vanuit de ZGW API is op AMQP gebaseerd. Dat betekent dat men werkt met topics in AMQP waarop de notificaties gepubliceerd worden. Vervolgens worden deze notificaties (na toepassing van filtering) verspreid naar de correcte partijen.

## ESuite implementatie

ESuite biedt inmiddels een NRC implementatie aan.

Net als bij de AC component is het mogelijk om een centrale NRC component los aan te bieden of de NRC te gebruiken die eSuite aanbiedt. Het wel of niet doorsturen van de JWT is reeds behandeld in [ZGW API Authenticatie en Autorisaties](#). We gaan dat hier niet nogmaals behandelen aangezien de keuze bepaalde gevolgen heeft voor de positionering van de AC.

## De NRC van eSuite gebruiken

In deze opzet wordt de NRC van eSuite gebruikt. Dat ziet er als volgt uit als partijen zich willen abonneren op notificaties:

1. Taakapplicatie stuurt een abonneer bericht naar de NRC op de ESB.
2. De ESB stuurt bericht door naar de NRC van eSuite.
3. eSuite registreert abonnement.

Omdat de eSuite NRC belast is met het "pushen" van notificaties betekent dit wel dat de url die men gebruikt bij het abonneren bereikbaar moet zijn vanaf de eSuite cloud omgeving. Dit is waarschijnlijk zeer moeilijk te beheren. Een mogelijk alternatief is om de NRC via de ESB te ontsluiten.

## NRC Proxy

In deze opzet wordt er een NRC op de ESB aangeboden. Het doel hierbij is om een NRC neer te zetten zonder state en vrijwel geen logica. Er wordt dan zwaar geleund op de eSuite NRC. Het voordeel van deze opzet is dat het component op de ESB draait en zodanig bij alle taakapplicaties kan komen. ESuite hoeft in deze opzet enkel notificaties te sturen naar de ESB.

In deze opzet ziet de workflow er als volgt uit:

1. Taakapplicatie abonneert zich bij de ESB NRC. Hierbij geeft taakapplicatie een url op waarop het notificaties wil ontvangen.
2. De ESB maakt een abonnement aan bij de eSuite NRC. Hierbij wordt als url de ESB url gebruikt aangevuld met extra gegevens zodat later de originele url die de taakapplicatie opgaf kan worden hersteld alsmede de bearer token die gebruikt moet worden.
3. eSuite verstuurt een notificatie (bijv. zaak aangemaakt) naar de ESB.
4. De ESB kan aan de hand van de parameters in de url zien waar de notificatie bezorgd moet worden en wat de bearer token moet zijn.
5. De ESB levert de notificatie bij de taakapplicatie af.

De taakapplicatie abonneert zich met een url + bearer token bij de ESB. Dat ziet er als volgt uit:

```
POST https://esb.rotterdam.nl/zaken/nrc/api/v1/abonnement
{
  "callbackUrl": "https://webhook.site/1eab0260-1a7b-4d10-b7c9-9b557e74d1f1",
  "auth": "dummy",
  "kanalen": [
    {
      "naam": "zaken",
      "filters": {
        "zaaktype": "https://esb.rotterdam.nl/zaken/ztc/api/v1/zaaktypen/6ab29afa-c3b2-4199-af11-6187b2133158"
      }
    }
  ]
}
```

De ESB meldt zich aan bij eSuite met het volgende bericht:

```
POST https://dontwikkelmidoffice.dommeldam.nl/zgw-apis-provider/nrc/api/v1/abonnement
{
  "callbackUrl": "https://esb.rotterdam.nl/zaken/nrc/api/v1/notificaties?url=aHR0cHM6Ly93ZWJob29rLnNpdGUvMWVhYjAyNjAtMWE3Yi00ZDEwLWI3YzktOWI1NTdlNzRkMWYx&token=ZHVtbXk=",
  "auth": "dummy",
  "kanalen": [
    {
      "naam": "zaken",
      "filters": {
        "zaaktype": "https://dontwikkelmidoffice.dommeldam.nl/zgw-apis-provider/ztc/api/v1/zaaktypen/6ab29afa-c3b2-4199-af11-6187b2133158"
      }
    }
  ]
}
```

De callbackurl bevat de url van de taakapplicatie + bearer token in een base64 encoded string.

ESuite zal bij wijzigingen dit bericht sturen:

```
{
  "aanmaakdatum": "2020-08-21T10:59:47.213472",
  "actie": "zaak_nieuw",
  "hoofdObject": "https://dontwikkelmidoffice.dommeldam.nl/zgw-apis-provider/zrc/api/v1/zaken/ba5aef4b-975e-4397-b264-23bcd1c18851",
  "kanaal": "zaken",
  "kenmerken": {
    "zaaktype": "https://dontwikkelmidoffice.dommeldam.nl/zgw-apis-provider/ztc/api/v1/zaaktypen/6ab29afa-c3b2-4199-af11-6187b2133158"
  },
  "resource": "zaken",
  "resourceUrl": "https://dontwikkelmidoffice.dommeldam.nl/zgw-apis-provider/zrc/api/v1/zaken/ba5aef4b-975e-4397-b264-23bcd1c18851"
}
```

Dit bericht gaat naar de url: <https://esb.rotterdam.nl/zaken/nrc/api/v1/notificaties?url=aHR0cHM6Ly93ZWJob29rLnNpdGUvMWVhYjAyNjAtMWE3Yi00ZDEwLWI3YzktOWI1NTdlNzRkMWYx&token=ZHVtbXk=>

Als de ESB de notificatie krijgt op bovenstaand url doet de ESB het volgende:

1. Decode de url parameter: <https://webhook.site/1eab0260-1a7b-4d10-b7c9-9b557e74d1f1>
2. Decode de bearer token: dummy

De ESB stuurt vervolgens deze notificatie naar de taakapplicatie:

```
{
  "aanmaakdatum": "2020-08-21T10:59:47.213472",
  "actie": "zaak_nieuw",
  "hoofdObject": "https://esb.rotterdam.nl/zaken/zrc/api/v1/zaken/ba5aef4b-975e-4397-b264-23bcd1c18851",
  "kanaal": "zaken",
  "kenmerken": {
    "zaaktype": "https://esb.rotterdam.nl/zaken/ztc/api/v1/zaaktypen/6ab29afa-c3b2-4199-af11-6187b2133158"
  },
  "resource": "zaken",
  "resourceUrl": "https://esb.rotterdam.nl/zaken/zrc/api/v1/zaken/ba5aef4b-975e-4397-b264-23bcd1c18851"
}
```

Dit bericht gaat naar de url <https://webhook.site/1eab0260-1a7b-4d10-b7c9-9b557e74d1f1> met de bearer token dummy.

## Een centrale NRC component

Een centrale NRC component stelt de organisatie in staat om notificaties makkelijker door de organisatie te verspreiden. De workflow ziet er dan als volgt uit:

1. De ESB abonneert zich bij de NRC van eSuite voor alle kanalen.
2. Taakapplicatie abonneert zich bij de ESB NRC service op zaken met het zaaktype x.
3. eSuite verstuurt een notificatie (bijv. zaak aangemaakt) naar de ESB.
4. De ESB stuurt de notificatie door naar de centrale NRC.
5. De centrale NRC kijkt welke taakapplicatie zich geabonneerd hebben.
6. De centrale NRC stuurt de notificatie naar de taakapplicatie. De door de taakapplicatie aangemelde url wordt hierbij gebruikt.

Deze opzet heeft als grote voordeel dat de centrale NRC intern draait en de notificatie makkelijk kan verspreiden.

Vanwege het modulaire karakter van de componenten in de ZGW API standaard is het mogelijk om de centrale NRC apart af te nemen. Hiervoor bestaan er meerdere opties:

1. Inpassen van de Referentie Implementatie in Rotterdam
2. Zelfbouw
3. Aanschaffen kant en klare NRC component

De Referentie Implementatie is een snelle manier om up en running te zijn maar zal waarschijnlijk een generiek karakter hebben en mogelijk zijn de beheerschermen moeilijk in te passen in de infrastructuur. Bovendien hebben we nog geen mogelijkheid gevonden om de broncode van de Referentie Implementatie te krijgen.

Zelfbouw is tegenwoordig zeer krachtig en kan op een snelle manier plaats vinden. Hierbij wordt voor de frontend Mendix toegepast en voor de backend Java services op basis van JSON/Rest. Dit biedt zeer veel flexibiliteit voor het bedienen van beheerders van de notificaties. Krachtige tools zoals inzicht in aantallen notificaties, notificaties die klaar staan enz. zijn dan mogelijk.

Mogelijk bestaat er op de markt een kant en klare NRC component waardoor zelfbouw niet nodig is. Mogelijk is de toekomstige eSuite NRC los te trekken en zelfstandig te gebruiken.