

# Actiereeksen in TOPdesk: een handleiding

Auteur: Rogier Visser (Laansloot IT)

Datum: 11 maart 2020

Naam: Laansloot-Actiereeksen-handleiding.pdf

Versie: 1.0

Doelgroep: TOPdesk-applicatiebeheerders

## Introductie

Actiereeksen binnen TOPdesk kun je gebruiken om van alles en nog wat te automatiseren in je eigen TOPdesk. Deze technologie is gewoon beschikbaar in recente versies van TOPdesk, maar er is nog weinig documentatie beschikbaar. Deze handleiding laat zien hoe actiereeksen precies werken, en hoe je daarmee oneindige mogelijkheden hebt om functionaliteit te bouwen in TOPdesk.

Deze handleiding sluit aan bij het actiereeks-archief op het TOPdesk-forum van Laansloot IT. Het actiereeks-archief is een naslagwerk van actiereeksen waarmee je terugkerende taken binnen TOPdesk kunt automatiseren.

Je vindt het actiereeks-archief op de volgende locatie:

<https://topdeskforum.laansloot.nl/c/actiereeks-archief>

## Disclaimer

Deze handleiding is puur van informatieve aard. Laansloot IT kan niet aansprakelijk gesteld worden voor enigerlei schade die zou kunnen voortkomen uit het volgen van deze handleiding. Laansloot IT geeft verder geen garantie dat actiereeksen doen wat ze moeten doen, en Laansloot IT geeft ook geen support hierop. Het volgen van deze handleiding is dus altijd op eigen risico.

Heb je behoefte aan hulp? Stel dan je vraag op het TOPdesk-forum:

<https://topdeskforum.laansloot.nl/>

## Copyright en licentie

Copyright op de teksten in deze handleiding ligt bij Laansloot IT. Copyright op de screenshots van TOPdesk ligt bij TOPdesk. Deze handleiding wordt gepubliceerd onder de licentie Creative Commons BY-NC-ND 4.0:

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.nl>

## Waar gaat dit document precies over?

Actiereeksen in TOPdesk bestaan al sinds 2017. Met deze functionaliteit is het mogelijk om TOPdesk te integreren met andere systemen. Minder bekend is dat je actiereeksen ook heel goed kan inzetten zonder integratie met andere systemen. Je kunt actiereeksen namelijk gebruiken om van alles en nog wat te automatiseren in je eigen TOPdesk. De technologie hiervoor is al helemaal ingebakken in TOPdesk.

Wat tot nog toe ontbreekt is duidelijke documentatie over actiereeksen. De drempel om aan de slag te gaan met actiereeksen ligt daarom dus hoog. Laansloot IT wil deze drempel verlagen, en stelt daarom gratis documentatie beschikbaar. Het doel hiervan is dat TOPdesk-applicatiebeheerders effectief aan de slag kunnen met actiereeksen.

### Werken met de TOPdesk API

Desondanks is werken met actiereeksen, ook met uitgebreide documentatie, niet eenvoudig. Actiereeksen werken namelijk door middel van de TOPdesk API, en werken met APIs is nooit een intuïtieve gebruikerservaring. Heb je al enige programmeerervaring? Dan helpt dat enorm wanneer je met actiereeksen aan de slag gaat.

Op de werking van de TOPdesk API gaat dit document helemaal niet in. Je zult dus geregeld de officiële documentatie daarover moeten raadplegen:

<https://developers.topdesk.com/#documentation>

# Inhoudsopgave

<b>Hoofdstuk 1: Aan de slag met actiereeksen in TOPdesk</b>	<b>5</b>
• Stap 1: Rechtengroep aanmaken	6
• Stap 2: Behandelaar aanmaken	7
• Stap 3: Applicatiewachtwoord aanmaken	8
• Stap 4: Authenticatie-string aanmaken	9
• Stap 5 (optioneel): Een onzichtbaar account	10
• Resultaat	11
<b>Hoofdstuk 2: Tips &amp; tricks</b>	<b>12</b>
• Tip 1: Zorg dat TOPdesk goed is geconfigureerd	13
• Tip 2: Houd je testomgeving gescheiden van je productieomgeving	14
• Tip 3: UNIDs vinden met het pi-symbool	16
<b>Hoofdstuk 3: Je eerste GET-call</b>	<b>18</b>
• Stap 1: Actiereeks aanmaken	19
• Stap 2: Variabelen opvoeren	20
• Stap 3: API-call inregelen	20
• Stap 4: Header invoeren	21
• Stap 5: Activeren	21
• Stap 6: Uitvoeren	22
• Stap 7: Responsedata structureren	23
<b>Hoofdstuk 4: Werken met stappen</b>	<b>24</b>
• De actiereeks in TOPdesk (screenshot)	25
• Koppeling tussen stap 1 en stap 2	26
• JSON-responsedata	27
• JSON-responsedata verbeterd weergegeven	28
• UNID bij direct velden invoegen	29
• Voorbeeld met stappen uit de praktijk	30
• Lijsten: één resultaat gebruiken	31
• Lijsten: meerdere resultaten gebruiken	32
• Lijsten: een waarde opzoeken in een lijst	33
<b>Hoofdstuk 5: Waarden en mappings</b>	<b>34</b>
• Waarde invoegen: Huidige waarden	35
• Waarde invoegen: Oude waarden	36
• Waarde invoegen: Variabelen	38
• Waarde invoegen: Progress trail	39
• Waarde invoegen: Reactiegegevens	43
• Waardetoewijzingen	44

<b>Hoofdstuk 6: If-then-constructies in Freemarker</b>	<b>47</b>
• If-then-constructie in "Aangepaste voorwaarde"	48
• If-then-constructie in de JSON-body	54
• Uitgebreide If-then-constructies	56
• ?contains-functie	58
<b>Hoofdstuk 7: Variabelen</b>	<b>59</b>
• Een variabele toewijzen met <#assign>	60
• Meerdere <#assign>	61
• Gereserveerde variabelen vanuit TOPdesk	62
• Variabele zoeken met een If-then-constructie	63
• Variabele zoeken met een lijst	64
• <#assign> binnen een URL	66
<b>Hoofdstuk 8: Tekstmanipulatie</b>	<b>67</b>
• ?replace	68
• ?keep_before, ?keep_before_last, ?keep_after, ?keep_after_last	71
• ?remove_beginning, ?remove_ending	72
• ?ensure_starts_with, ?ensure_ends_with	73
• Slicing	74
• Functies combineren	76
• Uit de praktijk: data uit een ingediend formulier opzoeken	77
• Regex (geavanceerd)	78
<b>Hoofdstuk 9: Datumvelden</b>	<b>79</b>
• Platte datumvelden en tijdzones	80
• UTC-tijd via de TOPdesk API	82
• Over datumobjecten en strings	83
• String omzetten naar datumobject	84
• Datumobject omzetten naar string	85
• Later dan / eerder dan	87
• Dezelfde dag	88
• Huidige datum/tijd ophalen met .now	88

## Hoofdstuk 1

# Aan de slag met actiereeksen in TOPdesk

Dit hoofdstuk laat zien hoe je TOPdesk in de basis moet configureren om actiereeksen te gebruiken. Volg je deze handleiding nauwgezet, dan is het activeren van een afzonderlijke actiereeks uit het actiereeks-archief van Laansloot IT daarna een eenvoudige invuloefening.

## Stap 1: Rechtengroep aanmaken

Maak een rechtengroep aan. Geef deze een duidelijk herkenbare naam, bijvoorbeeld "TOPdesk serviceaccount: AUTOMATION".

### Algemeen


Naam

TOPdesk serviceaccount: AUTOMATION

Later maken we een behandelaarsaccount met de naam AUTOMATION. Hierdoor is de relatie tussen rechtengroep en account duidelijk.

Geef de nieuwe rechtengroep de juiste twee API-vinkjes:

#### Reporting API

	Read	Write	Create	Delete	Archive	Create Selection
Apply to all	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
REST API	<input checked="" type="checkbox"/>					
Advanced reporting data access	<input type="checkbox"/>					
Use application passwords 		<input checked="" type="checkbox"/>				

Zet natuurlijk ook de juiste vinkjes voor de kaarten die je moet kunnen lezen of bewerken. Bijvoorbeeld als je iets met meldingen wilt gaan doen:

#### Call Management

	Read	Write	Create	Delete	Archive	Create Selection
Apply to all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
First line calls	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Second line calls	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Standard Solutions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Op het aantekeningenveld kun je bij wijze van documentatie iets schrijven:

#### Aantekeningen

Admin

20 maart 2019 13:55

Dit recht is bedoeld voor de serviceaccounts die in TOPdesk via de REST API velden invult en kaarten aanmaakt. Dit zijn alle actiereeksen die te vinden zijn onder Modules -> Actiebeheer -> Actieverkenner.



## Stap 2: Behandelaar aanmaken

Maak een nieuwe behandelaar met zowel achternaam als inlognaam AUTOMATION.

Kies als wachtwoord een moeilijke random string, bijvoorbeeld van Random.org:  
<https://www.random.org/passwords/?num=100&len=15&format=html&rnd=new>

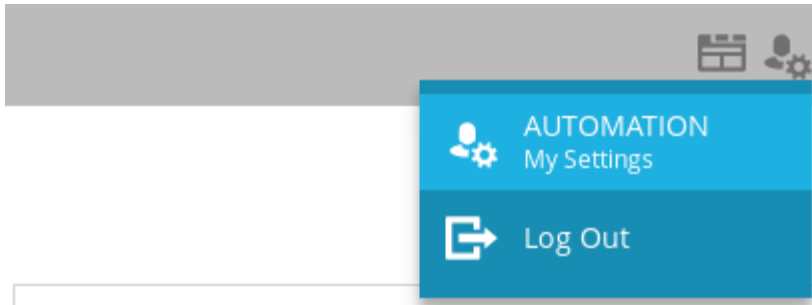
The screenshot shows the 'AUTOMATION Nieuwe behandelaar' form. The 'Algemeen' tab is active. The 'Algemeen' section contains fields for Achternaam (AUTOMATION), Voornaam, Voorletters, Tussenvoegsels, Geboortenaam, Titel, Geslacht, and Taal (Nederlands). The 'Locatie' section contains fields for Vestiging (Laansloot IT), Debiteurennummer, Ruimte, and Plaats. The 'Contactgegevens' section contains fields for Telefoon, Mobiel nummer, Faxnummer, E-mail, and Exchange-account (checked 'Zelfde als e-mail'). An 'Inloggegevens' modal is open, showing fields for Inlognaam (AUTOMATION), Wachtwoord, and Wachtwoord nogmaals. The modal has buttons for 'Verwijder...', 'OK', and 'Annuler...'. Below the modal, there are checkboxes for 'Tweedenivelmeldingbehandelaar', 'Probleembeheerder', 'Probleembehandelaar', 'Externe helpdeskp partij', 'Contractbeheerder', 'Operationeel behandelaar', and 'Operationeel beheerder'.

Sla het wachtwoord versleuteld en veilig op, en koppel het bij stap 1 aangemaakte rechtenprofiel aan de behandelaar AUTOMATION.

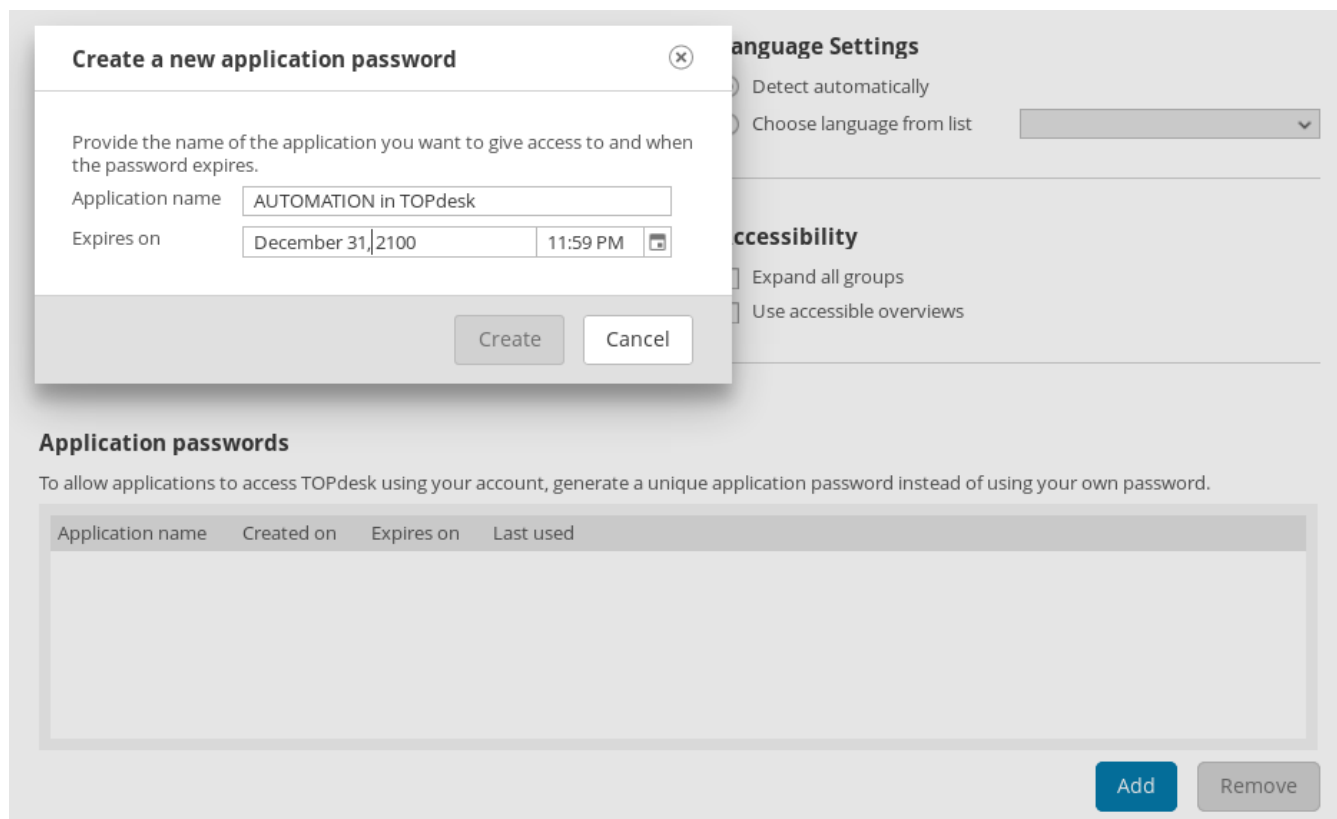
### Stap 3: Applicatiewachtwoord aanmaken

Log handmatig in met het nieuw aangemaakte account via de volgende url:  
<https://locatietopdesk.net.com/tas/secure/login/form>

Kies rechtsboven de optie My Settings onder de Gebruikers-knop:

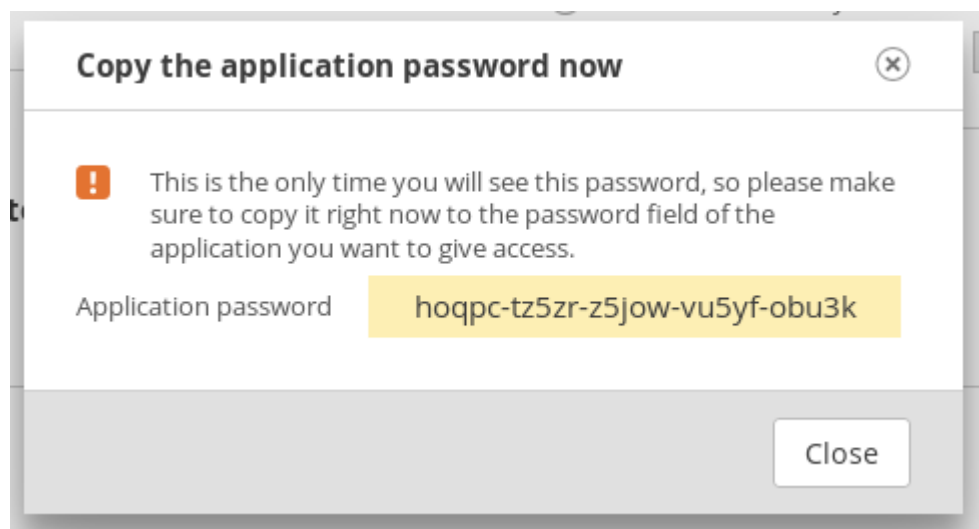


In het nieuwe scherm maak je een applicatiewachtwoord aan met de blauwe knop Add, rechts onderaan. Zet de verloopdatum ergens ver in de toekomst, en geef het applicatiewachtwoord een duidelijke naam:





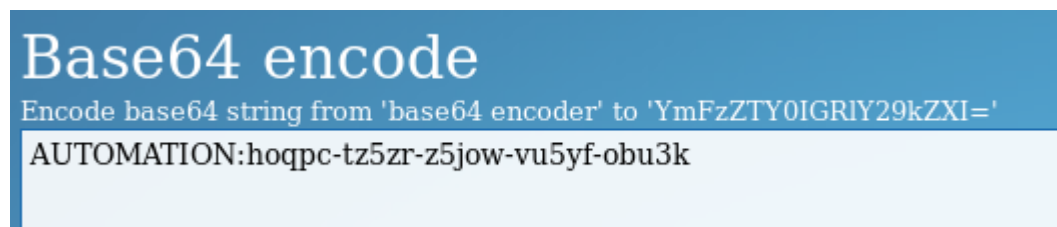
Klik Create, en je applicatiewachtwoord verschijnt. Sla deze string versleuteld en veilig ergens op.



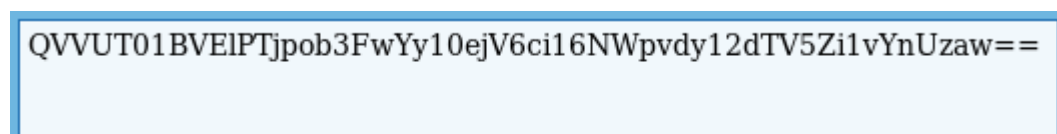
#### Stap 4: Authenticatie-string aanmaken

Ga nu naar de volgende website:  
<https://www.base64encode.net/>

Vul in het eerste veld je inlognaam en applicatiewachtwoord in, gescheiden door dubbele punt (zoals hieronder)



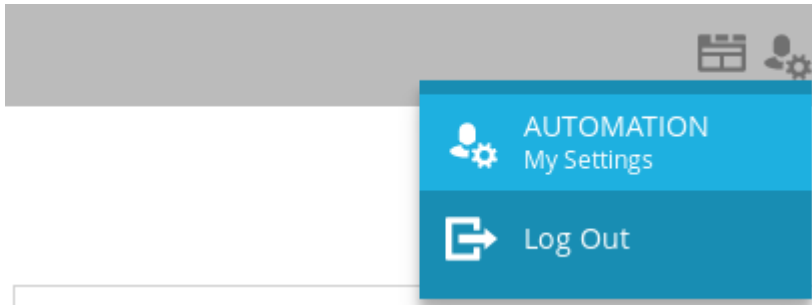
Klik op Encode, en je authenticatie-string komt eruit:



## Stap 5 (optioneel): Een onzichtbaar account

Een optionele extra stap is om een tweede behandelaar aan te maken, die zo is ingesteld dat alle tekstuele updates standaard onzichtbaar voor de aanmelder worden gemaakt. Dit heeft mijn voorkeur, omdat daarmee de werking van je geautomatiseerde systeem wat intuïtiever wordt.

Maak hiervoor op dezelfde manier een behandelaar aan, maar nu met de naam AUTOMATION\_ONZICHTBAAR. Log handmatig in, net als bij stap 3, en open weer de optie My Settings onder de Gebruikers-knop:

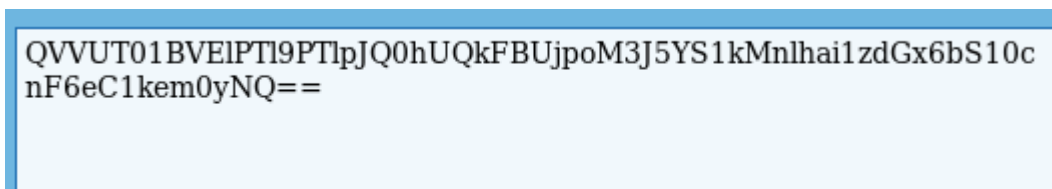


Halverwege zie je daar de optie om de standaard zichtbaarheid in te stellen. Zet deze op Invisible:

### Default for Make invisible to caller

- ☐ Visible  
☒ Invisible

Je kunt verder op dezelfde manier de authenticatie-string aanmaken.



Aan je rechtengroep hangen daarna 2 behandelaars:

## Rechtengroep TOPdesk serviceaccount: A...

ALGEMEEN

BEHANDELAARS (2)

AANTEKENINGEN

FILTER: geen

- |                          |                        |
|--------------------------|------------------------|
| <input type="checkbox"/> | Inlognaam              |
| <input type="checkbox"/> | AUTOMATION             |
| <input type="checkbox"/> | AUTOMATION_ONZICHTBAAR |

Een onzichtbaar account is overigens niet strikt noodzakelijk. De TOPdesk API staat het namelijk ook toe om bij losse actie-updates aan te geven of ze onzichtbaar moeten worden of niet.

## Resultaat

Uiteindelijk heb je twee strings, voor een account dat updates standaard zichtbaar maakt, en voor een account dat updates standaard onzichtbaar maakt:

QVVUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==

QVVUT01BVEIPTI9PTlpjQ0hUQkFBUjpoM3J5YS1kMnlhai1zdGx6bS10cnF6eC1kem0yNQ==

Dit zijn de strings die je later aanroept als variabelen in je actiereeksen:

Variabelen	
Naam	Waarde
TOPdeskURL	https://locatietopdesk.net.com
TOPdeskLoginZichtbaar	QVVUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==
TOPdeskLoginOnzichtbaar	QVVUT01BVEIPTI9PTlpjQ0hUQkFBUjpoM3J5YS1kMnlhai1zdGx6bS10cnF6eC1kem0yNQ==

De actiereeksen in het actiereeks-archief zijn op deze manier opgebouwd, dus je kunt aan de slag!

<https://topdeskforum.laansloot.nl/c/actiereeks-archief>

## Hoofdstuk 2

# Tips & tricks

Actiereeksen zijn ontzettend krachtig om terugkerende taken in TOPdesk te automatiseren. De drempel om ermee aan de slag te gaan is echter hoog, en dat komt vooral door een gebrek aan documentatie.

In dit hoofdstuk geef ik drie belangrijke tips die het werken met actiereeksen een stuk veiliger en duidelijker maken.

## **Tip 1: Zorg dat TOPdesk goed is geconfigureerd**

### **Vorbereiding**

Beetje open deur, maar zorg ervoor dat je basis goed is. Dat doe je simpelweg door de alle stappen uit het eerste hoofdstuk nauwgezet te volgen,

Uiteindelijk heb je twee authenticatie-strings, voor een account dat updates standaard zichtbaar maakt, en voor een account dat updates standaard onzichtbaar maakt. Die accounts roep je aan in de actiereeksen.

### **De juiste rechten**

Je gebruikt een serviceaccount in TOPdesk om actiereeksen uit te voeren. Dit account moet de juiste rechten hebben om te kunnen doen wat de actiereeksen moeten doen. Het account moet ook niet meer rechten hebben dan noodzakelijk.

Actiereeksen kunnen omgaan met bijna alle modules van TOPdesk. Wil je een melding geautomatiseerd laten aanmaken wanneer er import mislukt? Zorg er dan bijvoorbeeld voor dat je serviceaccount rechten heeft om meldingen te kunnen aanmaken.

Denk dus na over de rechten die je serviceaccount nodig heeft.

## Tip 2: Houd je testomgeving gescheiden van je productieomgeving

### Probleemstelling

De actiereeksen in het actiereeks-archief op het TOPdesk-forum van Laansloot IT hebben allemaal hetzelfde doel, namelijk om taken in TOPdesk te automatiseren. Die actiereeksen doen dat door API-calls op de eigen TOPdesk af te vuren. Heb je bijvoorbeeld je TOPdesk draaien op <https://locatietopdesk.net.com>, dan stel je in de actiereeksen de URL hardcoded op <https://locatietopdesk.net.com>:



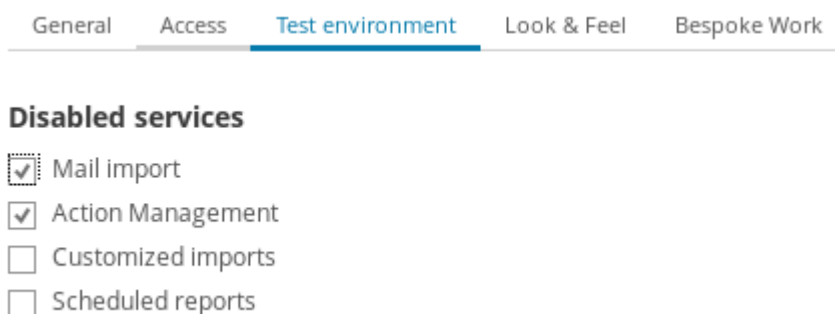
Naam	Waarde
TOPdeskURL	https://locatietopdesk.net.com
TOPdeskLoginZichttt	QWUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==

Ga je nu je productieomgeving kopiëren naar je testomgeving, dan gaan je actiereeksen (met hardcoded de URL van je productieomgeving!) ook mee. Het kan dan gebeuren dat de actiereeksen vanuit je testomgeving data gaan aanpassen in je productieomgeving. Dat wil je absoluut vermijden.

### Oplossing 1: events en actions helemaal uitschakelen

Het probleem is opgelost wanneer je de events en actions in je testomgeving uitschakelt. Je weet dan 100% zeker dat je testomgeving nooit je productieomgeving gaat aanpassen.

Draai je TOPdesk on premise? Log dan in de Administrator console in, en zet het vinkje "Action Management" aan (aanvinken is hier gelijk aan functionaliteit uitschakelen).



General Access **Test environment** Look & Feel Bespoke Work

**Disabled services**

- ☒ Mail import
- ☒ Action Management
- ☐ Customized imports
- ☐ Scheduled reports

Draai je TOPdesk SAAS, dan kan TOPdesk support de events en actions voor je uitschakelen.

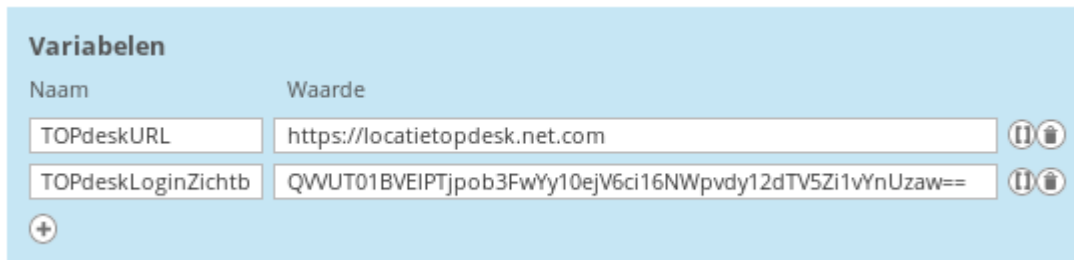
NB: "Action Management" uitschakelen zorgt er ook voor dat er geen uitgaande emails in TOPdesk meer beschikbaar zijn.

## Oplossing 2: URLs aanpassen na kopieerslag

Het nadeel van de eerste oplossing is dat je je testomgeving niet kan gebruiken om je actiereeksen in te testen. Dan heb je alleen je productieomgeving om in te ontwikkelen, en dat is niet best practice.

Wil je wel je testomgeving houden om actiereeksen in te ontwikkelen? Dan ontkom je er niet aan om met de hand de URLs in de actiereeksen in de testomgeving aan te passen. Het gaat dan om de variabele TOPdeskURL in de actiereeksen. **Zorg ervoor dat je in je testomgeving geen URLs hebt die verwijzen naar de productieomgeving.**

Vind je dus een actiereeks in je testomgeving zoals onderstaande:



Naam	Waarde
TOPdeskURL	https://locatietopdesk.net.com
TOPdeskLoginZichtb	QWUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==

verander dan gelijk de URL zodat deze daadwerkelijk naar de testomgeving verwijst:



Naam	Waarde
TOPdeskURL	https://locatietesttopdesk.net.com
TOPdeskLoginZichtb	QWUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==

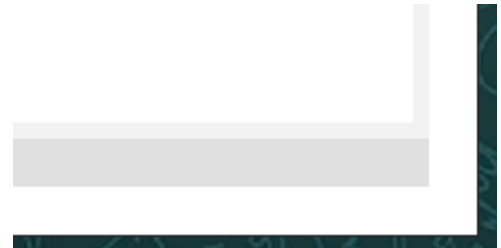
Het resultaat is dat je productieomgeving je productieomgeving aanspreekt, en je testomgeving je testomgeving. Precies zoals het hoort dus.

**Ga je je productieomgeving kopiëren naar je testomgeving? Pas dan daarna meteen je actiereeksen één-voor-één aan in de bijgewerkte testomgeving.**

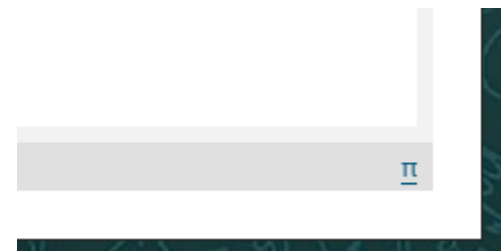
### Tip 3: UNIDs vinden met het pi-symbool

Vroeger was internet nieuw. In die tijd had je de film "The Net", waarin Sandra Bullock de muis naar de rechteronderhoek van het scherm beweegt. Daar zit een pi-tekentje, en dat zit ook in TOPdesk.

Open een overzichtsscherm, bijvoorbeeld een lijst met meldingen. Selecteer in de lijst een melding, maar open de kaart niet. Rechtonder zie je niets.

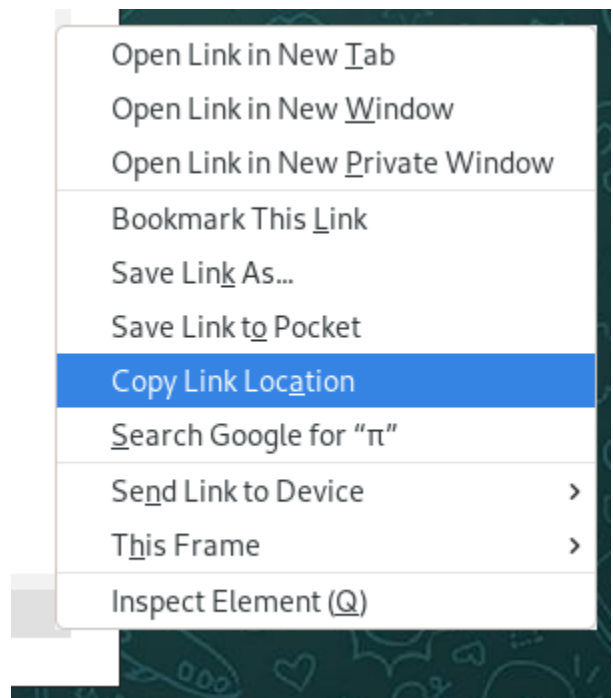


Maar beweeg je de muis nu in de rechterhoek van het grijze gedeelte, dan verschijnt er een pi-tekentje:



Klik nu rechtermuisknop, en kies "Copy Link Location" (of "Snelkoppeling kopiëren" oid).

Wat je hiermee kopieert is de directe link naar de kaart die je hebt gekozen. Dat werkt voor bijna alle soorten kaarten in TOPdesk, zoals meldingen, personen, rapporten, selecties, etc.





Open nu een tekstverwerker en plak de link. Deze link is opgebouwd uit twee delen

Deel 1	Deel 2
<code>https://locatietopdesk.net.com/tas/secure/incident?</code>	<code>unid=8e6b2adcc50941b29eff391a6b3ea136</code>

Het tweede deel bevat de UNID:

`8e6b2adcc50941b29eff391a6b3ea136`

De API accepteert echter alleen UNIDs in het format 8-4-4-4-12. Er moeten dus nog een paar streepjes tussen:

`8e6b2adc-c509-41b2-9eff-391a6b3ea136`

Dit is de waarde die je kunt gebruiken in je actiereeksen, bijvoorbeeld als variabele.

## Hoofdstuk 3

# Je eerste GET-call

In feite zijn actiereeksen niets anders dan API-calls die je op TOPdesk zelf afvuurt. De simpelste API-call is een GET-call, waarmee je een kaart kunt ophalen. Met TOPdesk kun je helemaal zelf een GET-call uitvoeren. Je hebt hier geen externe programmatuur of browser-plugins voor nodig.

In dit hoofdstuk laat ik zien hoe je met een actiereeks een GET-call uitvoert op een tweedelijns melding, met als doel om waardes op te halen uit de meldingkaart. Dit is functionaliteit die vaak terugkeert als je later zelf actiereeksen gaat ontwikkelen.

Op de werking van de API van TOPdesk gaat dit document niet in. Dit is allemaal gedocumenteerd op de volgende pagina:

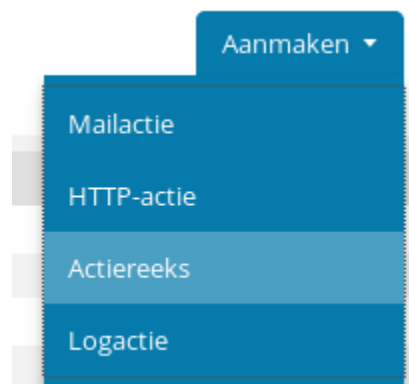
<https://developers.topdesk.com/#documentation>

## Stap 1: Actiereeks aanmaken

Ga naar Modules → Actiebeheer → Actieverkenner, en kies daar Aanmaken → Actiereeks.

Kies bij Kaartsoort bijvoorbeeld Meldingenbeheer – Tweedelijns melding, en daarna Nieuwe actiereeks aanmaken.

Daarna krijg je het lege actiereeks-scherm te zien. Dat gaan we invullen.



**ALGEMEEN**   WAARDETOEWIJZINGEN   UITVOERINGSLOGS

---

**Tweedelijns melding**

**Naam \***

**Beschikbaar**

☐ Actief

☐ Toepassen in de Selfserviceportal

☐ Toepassen in het Behandelaarsgedeelte

**Gekoppelde gebeurtenissen**

**Beschikbaar in contextmenu**

**Variabelen**

**Stap 1**

Naam

HTTP-methode

**URL \***

**Headers**

**Deze stap uitvoeren**

## Stap 2: Variabelen opvoeren

Bovenaan in het scherm is het onderdeel "Variabelen" van de actiereeks. Met de plusknop kun je daar de volgende twee variabelen aanmaken:

- TOPdeskURL
- TOPdeskLoginZichtbaar

Resultaat:

Naam	Waarde
TOPdeskURL	https://locatietopdesk.net.com
TOPdeskLoginZichtt	QWUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==

## Stap 3: API-call inregelen

In het vak "Stap 1" geef je de informatie op voor de API-call zelf:

**Stap 1**

Naam: getmelding

HTTP-methode: GET

URL: \$\_variables.TOPdeskURL?no\_esc}/tas/api/incidents/id/\${unid}

**Headers**

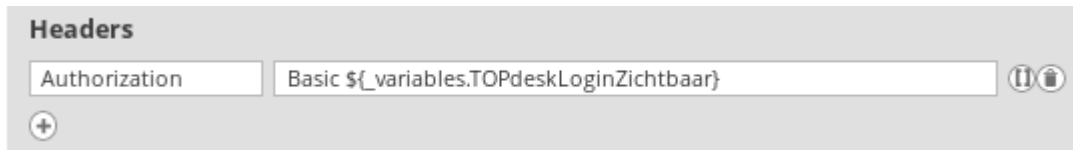
Deze stap uitvoeren: Altijd

- Het veld "Naam" kun je zelf kiezen
- Het veld "HTTP-methode" is GET
- Het veld "URL" moet precies kloppen:

```
$_variables.TOPdeskURL?no_esc}/tas/api/incidents/id/${unid}
```

## Stap 4: Header invoeren

In de vorige screenshot zie je het knopje "Headers", met de plus-knop daaronder. Daar gaan we een header invoeren waarmee we authenticeren. Voor een GET-call is alleen de header "Authorization" nodig.



Headers	
Authorization	Basic \${_variables.TOPdeskLoginZichtbaar}

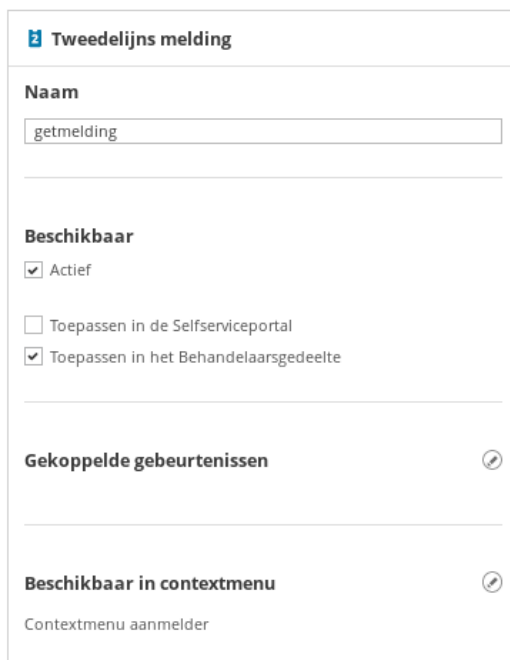
De waarde van het veld "Authorization" moet ook hier precies kloppen:

Basic \${\_variables.TOPdeskLoginZichtbaar}

## Stap 5: Activeren

Geef de actiereeks tenslotte een naam, maak actief op het Behandelaarsgedeelte, en voeg het toe in een contextmenu, bijvoorbeeld het contextmenu "Aanmelder". Sla op.

ALGEMEEN   WAARDETOEWIJZINGEN   UITVOERINGSLOGS



**Tweedelijns melding**

**Naam**

getmelding

**Beschikbaar**

☒ Actief

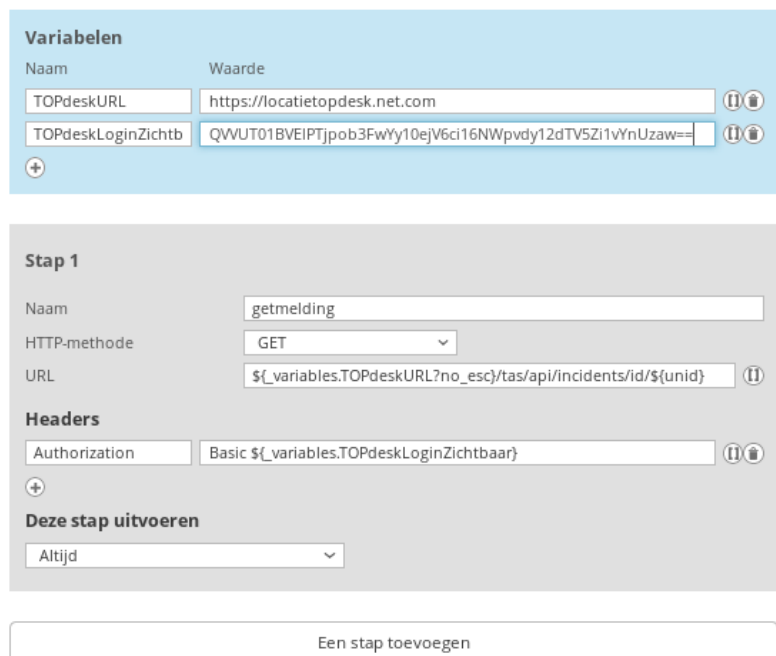
☐ Toepassen in de Selfserviceportal

☒ Toepassen in het Behandelaarsgedeelte

**Gekoppelde gebeurtenissen**

**Beschikbaar in contextmenu**

Contextmenu aanmelder



**Variabelen**

Naam	Waarde
TOPdeskURL	https://locatietopdesk.net.com
TOPdeskLoginZichtb	QVWUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==

**Stap 1**

**Naam**

getmelding

**HTTP-methode**

GET

**URL**

\${\_variables.TOPdeskURL?no\_esc}/tas/api/incidents/id/\${\_variables.TOPdeskLoginZichtb}

**Headers**

Headers	
Authorization	Basic \${_variables.TOPdeskLoginZichtbaar}

**Deze stap uitvoeren**

Altijd

Een stap toevoegen

## Stap 6: Uitvoeren

De actiereeks kunnen we nu eindelijk uitvoeren. In het contextmenu "Aanmelder" hebben we hiervoor nu het knopje "getmelding", waarmee we de actiereeks kunnen aftrappen.



Na het uitvoeren van de actiereeks kun je op het tabblad "Uitvoeringslogs" van de actiereeks het resultaat zien. In onderstaande screenshot zie je dat de eerste drie pogingen zijn mislukt, maar dat de meest recente wel is gelukt:

ALGEMEEN	WAARDETOEWIJZINGEN	UITVOERINGSLOGS
Kaartnaam	Uitvoeringsdatum	Gelukt
M19020004	2019-03-27T11:09:46.350+0000	✓
M19020004	2019-03-27T11:09:33.797+0000	✗
M19020004	2019-03-27T11:08:36.860+0000	✗
M19020004	2019-03-27T11:08:08.683+0000	✗

Klik op een logentry om daaronder de daadwerkelijke logs te zien. Dat zijn vaak heel veel gegevens.

- Bij mislukte runs zijn de foutmeldingen het meest relevant.
- Bij gelukte runs vind je de meest relevante info in de responsedata, vanaf de tekst Response body. Daarin zitten ook de waarden die we zoeken en die je kunt uitlezen met de API. Scroll daarvoor verder in de logs:

```
Length=13613}
Response body: {"id":"2b582268-d652-4e9a-8517-77b0e2db4e22","status":"secondLine","number":"M19020005","request":"28-02-2019 15:37 Admin: \nDit is een melding van Klant2\nAfdeling is IV","requests":"/tas/api/incidents/id/2b582268-d652-4e9a-8517-77b0e2db4e22/requests","action":"/tas/api/incidents/id/2b582268-d652-4e9a-8517-77b0e2db4e22/actions","attachments":"/tas/api/incidents/id/2b582268-d652-4e9a-8517-77b0e2db4e22/attachments","caller":{"id":"33c11f65-897d-4c36-ba0b-6a2a5186ad21","dynamicName":"klant2","branch":{"id":"52027c9e-6542-4f96-993c-2428d9e1cc4c","name":"Klant 2","clientReferenceNumber":"","timeZone":"GMT+01:00","extraA":null,"extraB":null},"callerBranch":{"id":"52027c9e-6542-4f96-993c-2428d9e1cc4c","name":"Klant 2","clientReferenceNumber":"","timeZone":"GMT+01:00","extraA":null,"extraB":null},"callerLocation":null,"branchExtraFieldA":null,"branchExtraFieldB":null,"briefDescription":"test","externalNumber":"","category":{"id":"50e57348-7818-5ba3-933b-8c4730ca5dd1","name":"Basisdiensten"},"subcategory":{"id":"4a56f1c2-3108-5fe0-bad0-3f4464f069be","name":"Printen"},"callType":null,"entryType":null,"object":null,"branch":null,"location":null,"imn
```

## Stap 7: Responsedata structureren

De responsedata van je GET-call is een grote lap tekst die niet overzichtelijk is. Gelukkig kun je de data structureren zodat het beter leesbaar is. Dit kan met een gratis SAAS-dienstje van <https://unminify.com/>.

Kopieer de hele response body (vanaf de tekst Response body) en plak het integraal in het invoerveld op <https://unminify.com/>. Alles komt waarschijnlijk op 1 lijn terecht:

```
PASTE YOUR CODE HERE

1: null, "memo5": null, "searchlist1": null, "searchlist2": null, "searchlist3": null, "searchlist4": null, "searchlist5": null, "externalLinks": []}
```

Klik nu UNMINIFY, en het resultaat is leesbare en gestructureerde data!

```
PASTE YOUR CODE HERE

1 Response body: {
2   "id": "2b582268-d652-4e9a-8517-77b0e2db4e22",
3   "status": "secondLine",
4   "number": "M19020005",
5   "request": "28-02-2019 15:37 Admin: \nDit is een melding van Klant2\nAfdeling is IV",
6   "requests": "/tas/api/incidents/id/2b582268-d652-4e9a-8517-77b0e2db4e22/requests",
7   "action": "/tas/api/incidents/id/2b582268-d652-4e9a-8517-77b0e2db4e22/actions",
8   "attachments": "/tas/api/incidents/id/2b582268-d652-4e9a-8517-77b0e2db4e22/attachments",
9   "caller": {
10    "id": "33c11f65-897d-4c36-ba0b-6a2a5186ad21",
11    "dynamicName": "klant2",
12    "branch": {
13      "id": "52027c9e-6542-4f96-993c-2428d9e1cc4c", "name": "Klant 2", "clientReferenceNumber": "", "timeZone": "GMT+01:00",
14    }
15  }
16  "callerBranch": {
17    "id": "52027c9e-6542-4f96-993c-2428d9e1cc4c", "name": "Klant 2", "clientReferenceNumber": "", "timeZone": "GMT+01:00", "ext
18  }
19  "callerLocation": null,
20  "branchExtraFieldA": null,
21  "branchExtraFieldB": null,
22  "briefDescription": "test",
23  "externalNumber": "",
24  "category": {
25    "id": "50e57348-7818-5ba3-933b-8c4730ca5dd1", "name": "Basisdiensten"
26  }
27 }
28
29
30
```

Op deze manier kun je snel en eenvoudig de structuur herkennen van de data in de responsedata uit TOPdesk. Dit is belangrijk wanneer je verder actiereeksen gaat ontwikkelen.

### Laatste opmerkingen

Er zijn natuurlijk ook allerlei andere manieren om een API-call uit te voeren en zo de gevraagde data op te halen. Deze manier is juist relevant omdat het geïntegreerd is in TOPdesk. Ga je op een gegeven moment uitgebreide processen in TOPdesk automatiseren, dan zul je zien dat je losse GET-calls in je actiereeks nodig gaat hebben.

Verder kun je op dezelfde manier heel veel andere kaartsoorten in TOPdesk ophalen, zoals wijzigingen, personen, behandelaars, bijlages, etc. Ook in die gevallen ga je op dezelfde manier te werk.

## Hoofdstuk 4

# Werken met stappen

Onderstaand screenshot laat het resultaat van een actiereeks zien die geautomatiseerd een actie-update plaatst die de categorie van de melding toont:



De actiereeks die hiervoor zorgt is opgebouwd uit twee stappen:




1. een GET-call om de data van de melding op te halen
2. een PUT-call om de actie-update te plaatsen

Stap 2 kan alleen uitgevoerd worden wanneer stap 1 is uitgevoerd. Dat komt omdat de naam van de categorie (hier: "Applicaties") wordt teruggegeven in de responsedata van stap 1.

Dit hoofdstuk laat zien hoe je de responsedata van stappen in een actiereeks kunt gebruiken om stappen aan elkaar te koppelen.



## Variabelen

Naam	Waarde	
TOPdeskURL	https://locatietopdesk.net.com	 
TOPdeskLoginZichtbaar	QVVUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==	 
		


### Stap 1





Verwijderen

Naam

HTTP-methode

URL  

#### Headers



#### Deze stap uitvoeren


### Stap 2



Verwijderen

Naam

HTTP-methode

URL  

#### Headers





#### Body

☒ JSON-escaping toepassen op ingevoegde waarden  

```
{
  "action" : "De categorie is ${responses.getincident.body.category.name}"
}
```

#### Deze stap uitvoeren

## Koppeling tussen stap 1 en stap 2

De actiereeks op de vorige pagina bestaat uit twee stappen. In de eerste stap, een GET-call, wordt alle informatie van de melding opgehaald. In de tweede stap, een PUT-call, wordt uit de informatie van stap 1 één bepaalde waarde eruit gehaald – namelijk de categorie- en die wordt in de actie-update geplaatst.

In stap 2 van de actiereeks zie je dat de afhankelijkheid met stap 1 zit een variabele:

Variabele	Waarde
<code>\${_responses.getincident.body.category.name}</code>	Applicaties

Deze variabele is uit twee delen opgebouwd. Het eerste deel geeft aan welke brondata gebruikt moet worden. Dat is de responsedata van de stap `getincident`:

Body
<pre>{   "action" : "De categorie is \${_responses.getincident.body.category.name}" }</pre>

Het tweede deel geeft aan in welk element van de brondata precies de benodigde waarde zit. In dit geval is dat het element `.body.category.name`:

Body
<pre>{   "action" : "De categorie is \${_responses.getincident.body.category.name}" }</pre>

TOPdesk zoekt het element in de brondata op, pakt de waarde van dat element (in dit geval dus "Applicaties"), en voert daarna de PUT-call uit:

```
Step 2 'putincident'
[2020-02-03T21:24:51.103400Z] Starting FreeMarker template processing
[2020-02-03T21:24:51.104228Z] Finished FreeMarker template processing
[2020-02-03T21:24:51.104284Z] Request: PUT https://locatietopdesk.net.com/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da
Request headers: {Authorization=[BASIC QVVUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5ZI1vYnUzaw==], Content-Type=[application/json]}
Request body: {
  "action" : "De categorie is Applicaties"
}
```

## JSON-responsedata in TOPdesk

De JSON-responsedata van een GET-call is gestructureerde data, met een vaste opbouw in elementen. Wil je een bepaalde waarde eruit plukken, zoals "categorie"? Dan moet je het pad naar het juiste element in die opbouw achterhalen.

De JSON-responsedata van stap 1 kun je inzien in de uitvoeringslogs van de actiereeks:

```
Step 1 getincident
[2020-02-03T20:55:32.784287Z] Starting FreeMarker template processing
[2020-02-03T20:55:32.785460Z] Finished FreeMarker template processing
[2020-02-03T20:55:32.785531Z] Request: GET https://locatietopdesk.net.com/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da  ib8a-9333-9df8a3e610da
Request headers: {Authorization=[BASIC QVVUT01BVEIPT]pob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==]}
[2020-02-03T19:40:59.753438Z] Response status: 200 OK
Response headers: {Date=[Mon, 03 Feb 2020 19:40:59 GMT], X-Content-Type-Options=[nosniff], X-XSS-Protection=[1; mode=block], X-Robots-Tag=[noindex,nofollow], Access-Control-Allow-Origin=[*], Cache-Control=[no-store], Content-Type=[application/json; charset=utf-8], Content-Length=[4415]}
Response body: {"id":"798e0389-28e1-5b8a-9333-9df8a3e610da","status":"secondLine","number":"M 1707 002","request":"MARCELO: Wil graag werken met de nieuwste versie van TOPdesk","requests":"/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/requests","action":"/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/actions","attachments":"/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/attachments","caller":{"id":"be11c590-3f69-5744-81ea-572c350d5d69","dynamicName":"Kootstra, Hajo","branch":{"id":"deb5b857-9ae9-54dd-ad2ad0e7b3b9","name":"TOPdesk Nederland","clientReferenceNumber":"","timeZone":"Europe/Amsterdam","extraA":null,"extraB":null},"callerBranch":{"id":"deb5b857-9ae9-54dd-ad2ad0e7b3b9","name":"TOPdesk Nederland","clientReferenceNumber":"","timeZone":"Europe/Amsterdam","extraA":null,"extraB":null},"callerLocation":{"id":"ed4ff130-2952-598b-8f4c-001b72697650","name":"Helpdesk"},"branchExtraFieldA":null,"branchExtraFieldB":null,"briefDescription":"","externalNumber":"","category":{"id":"02c2b593-b1fa-53eb-8576-9b1538438003","name":"Applicaties"},"subcategory":{"id":"cd16d39b-c72d-4c76-8ebd-2aa4232d6da8","name":"TOPdesk"},"callType":{"id":"d230d6c4-9fcf-5312-9429-5953e725d903","name":"Service Request"},"entryType":{"id":"1c818506-af91-5438-9d7b-e0e25caa4d96","name":"E-mail"},"object":
```

- Het pad naar het precieze element binnen de responsedata is geel gehighlight.
- De waarde van het element is blauw gehighlight.

Variabele	Waarde
<code>\${_responses.<b>getincident.body.category.name</b>}</code>	<b>Applicaties</b>

Zo kun je de variabele die je in stap 2 van de actiereeks opvoert vinden.

### Ja maar de JSON-responsedata is niet overzichtelijk

In de uitvoeringslogs van de actiereeks is de JSON-responsedata onoverzichtelijk. Hier is gelukkig makkelijk iets aan te doen.

Kopieer de hele response body (vanaf de tekst Response body) en plak het integraal in het invoerveld op <https://unminify.com/>. Het resultaat is een weergave waarin je de structuur van de JSON-data veel duidelijker is (zie volgende pagina).

## JSON-responsedata verbeterd weergegeven

Onderstaande data is dezelfde responsedata als in de screenshot op de vorige pagina, maar dan gestructureerd via <https://unminify.com/>. Ook hier is het pad naar het juiste element gehighlight in geel, en de waarde van het element in blauw.

### getincident

```
Response body: {
  "id": "798e0389-28e1-5b8a-9333-9df8a3e610da",
  "status": "secondLine",
  "number": "M 1707 002",
  "request": "MARCELO: Wil graag werken met de nieuwste versie van TOPdesk",
  "requests": "/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/requests",
  "action": "/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/actions",
  "attachments": "/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/attachments",
  "caller": {
    "id": "be11c590-3f69-5744-81ea-572c350d5d69",
    "dynamicName": "Kootstra, Hajo",
    "branch": {
      "id": "deb5b857-9ae9-54dd-a769-ad2ad0e7b3b9", "name": "TOPdesk Nederland",
      "clientReferenceNumber": "", "timeZone": "Europe/Amsterdam",
      "extraA": null, "extraB": null }
    },
    "callerBranch": {
      "id": "deb5b857-9ae9-54dd-a769-ad2ad0e7b3b9", "name": "TOPdesk Nederland",
      "clientReferenceNumber": "", "timeZone": "Europe/Amsterdam",
      "extraA": null, "extraB": null
    },
    "callerLocation": { "id": "ed4ff130-2952-598b-8f4c-001b72697650", "name": "Helpdesk" },
    "branchExtraFieldA": null,
    "branchExtraFieldB": null,
    "briefDescription": "",
    "externalNumber": "",
    "category": { "id": "02c2b593-b1fa-53eb-8576-9b1538438003", "name": "Applicaties" },
    "subcategory": { "id": "cd16d39b-c72d-4c76-8ebd-2aa4232d6da8", "name": "TOPdesk" },
    "callType": { "id": "d230d6c4-9fcf-5312-9429-5953e725d903", "name": "Service Request" },
    "entryType": { "id": "1c818506-af91-5438-9d7b-e0e25caa4d96", "name": "E-mail" },
    "object": {
      "id": "655ba623-0ddb-5202-8d93-8ccd866a68db",
      "name": "TOPdesk Professional",
      "type": { "id": "bd8628a4-7ace-58f4-965e-b9ffd41ae0e1", "name": "Database" },
      "make": { "id": "863b50f4-56ef-5896-9b43-412a4fe42ce2", "name": "TOPdesk Nederland" },
      "model": null,
      "branch": { "id": "deb5b857-9ae9-54dd-a769-ad2ad0e7b3b9", "name": "TOPdesk Nederland" },
      "location": { "id": "75fa3598-1462-5f57-8618-5157da6faf4c", "name": "Computerruimte" },
      "specification": "Service Management applicatie",
      "serialNumber": ""
    },
    "asset": null,
    "branch": null,
    "location": null,
  },
  --afgebroken--
}
```

### Variabele

### Waarde

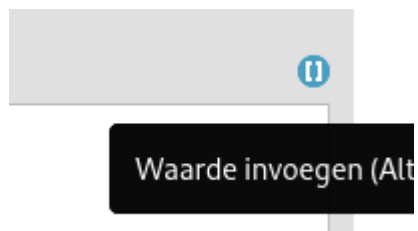
`$_responses.getincident.body.category.name`

Applicaties

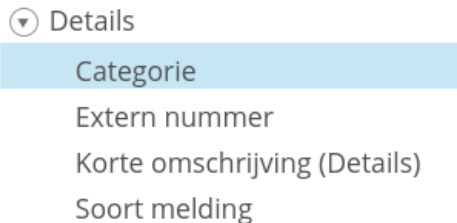
## UNID bij direct velden invoegen

In de actiereeks gebruiken we stap 1 om de naam van de categorie te achterhalen, via een GET-call.

Maar waarom moet dat in een aparte stap? Is dat niet overbodig? Je kunt toch ook de categorie als variable direct in stap 2 invoegen. Daarvoor is het knopje "[ ] -Waarde invoegen" rechtsboven het Body-veld van de actiereeks:



In het volgende scherm kun je dan ook "Categorie" kiezen (onder "Details").



### Voorbeeld

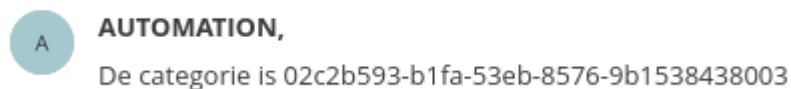
`${incident_domeinid}`

Met de code `${incident_domeinid}` kun je dus ook direct in stap 2 de categorie oproepen:

### JSON-body

```
{
  "action" : "De categorie is ${incident_domeinid}"
}
```

Maar wat je terugkrijgt met de code `${incident_domeinid}` is niet de naam van de categorie, maar de UNID:



Wil je dus iets anders dan de UNID terug, zoals de naam? Dan moet je een GET-call als extra stap inbouwen om het juiste veld op te vragen.

## Voorbeeld met stappen uit de praktijk

De actiereeks uit het bovenstaande voorbeeld bestaat uit twee stappen, een GET-call en een PUT-call. Er zijn echter allerlei scenarios te bedenken waarbij veel meer stappen benodigd zijn.

Eén van die scenario's is het geautomatiseerd aanmaken van behandelaarsaccounts vanuit een formulier. Het formulier dat daarvoor heb aangemaakt heeft maar 2 vragen:

### Benodigde informatie

Emailadres *	<input type="text"/>
Behandelaarsgroep *	<input type="text" value="v"/>

Na het indienen wordt een wijziging aangemaakt, en dat triggert een actiereeks. Die actiereeks doet het volgende:

- zoekt bij het ingevulde emailadres een persoon op,
- gebruikt de data vanuit de persoonskaart op de nieuwe behandelaarskaart,
- koppelt rechten, behandelaarsgroepen en eventuele filters op basis van de gekozen behandelaarsgroep

Deze actiereeks bestaat uit tien stappen:

	Soort	Kaart	Omschrijving	Responsesdata benodigd?
1	GET	wijziging	Wijziging ophalen	
2	GET	persoon	Persoonskaart ophalen	Stap 1
3	GET	behandelaarsgroep	Behandelaarsgroep ophalen	Stap 1
4	POST	behandelaar	Behandelaarsaccount aanmaken	Stap 2
5	PUT	behandelaar	Vinkjes bij taken zetten	Stap 1 & 4
6	POST	behandelaar	Behandelaarsgroep koppelen	Stap 3 & 4
7	POST	behandelaar	Rechtengroep koppelen	Stap 1 & 4
8	POST	behandelaar	Categoriefilter koppelen	Stap 1 & 4
9	POST	behandelaar	Behandelaarsfilter koppelen	Stap 1 & 4
10	PATCH	wijziging	Wijziging bijwerken en sluiten	Stap 1

Het aanmaken van behandelaarsaccounts is op deze manier dus helemaal weggeautomatiseerd met een actiereeks.

## Lijsten: één resultaat gebruiken

In bovenstaande voorbeeld krijg je in stap 1 maar *één* resultaat terug, namelijk één melding. Dat is de melding waarop je de GET-call doet.

In andere gevallen kan het echter ook zijn dat je *meerdere* resultaten terugkrijgen, in een lijst. Onderstaande JSON is de responsedata van een GET-call op de statussen van meldingen. Wat je terugkrijgt is in dit geval een lijst met *drie* resultaten.

### getstatuses

```
Response body: [
  {
    "id": "141a1a7a-0658-5768-ba02-642a8f754802",
    "name": "Nieuw",
    "onHold": "NOT_ON_HOLD",
    "processingState": "NOT_COMPLETED",
    "response": "CURRENT_VALUE",
    "default": true
  },
  {
    "id": "ef369639-3656-5449-acf0-e8a501b12922",
    "name": "In behandeling",
    "onHold": "NOT_ON_HOLD",
    "processingState": "NOT_COMPLETED",
    "response": "CURRENT_VALUE",
    "default": false
  },
  {
    "id": "a284e905-cc05-52ee-9f20-b1424a6c0f88",
    "name": "Niet reproduceerbaar",
    "onHold": "NOT_ON_HOLD",
    "processingState": "COMPLETED",
    "response": "CURRENT_VALUE",
    "default": false
  }
]
```

Lijsten herken je aan de vierkante begin- en eindhaken, zoals hierboven direct na Response body:.

De blauw gehighlighte waardes roep je als volgt op in een variabele:

Variabele	Waarde
<code>\${_responses.getstatuses.body[0].name}</code>	Nieuw
<code>\${_responses.getstatuses.body[1].name}</code>	In behandeling
<code>\${_responses.getstatuses.body[2].name}</code>	Niet reproduceerbaar

Je verwijst dus naar een specifiek resultaat in de body met de verwijzing `body[0]`, `body[1]`, of `body[2]`. NB: je begint altijd met tellen met `[0]`.

Een alternatieve syntax voor `body[0]` is `body?first`. Het laatste resultaat in de lijst kun je oproepen met `body?last` (en hier is geen alternatief met vierkante haken voor).

## Lijsten: meerdere resultaten gebruiken

Je kunt ook meerdere resultaten oproepen in een actiereeks, in plaats van één. Hieronder zie je de drie resultaten van de GET-call op de statuses in een actie-update:



**AUTOMATION,**

Nieuw In behandeling Niet reproduceerbaar

Daarvoor gebruik je de volgende code:

### Body

```
{
  "action" : "<#list _responses.getstatuses.body as result> ${result.name} </#list>"
}
```

NB: Zie je de spatie achter `${result.name}` en voor `</#list>`? Die zie je terug tussen de statuses in de screenshot hierboven. De syntax luistert dus nauw!

Wil je alleen de eerste twee resultaten in een lijst? Dan geef je dat aan op de volgende manier:

### Body

```
{
  "action" : "<#list _responses.getstatuses.body[0..1] as result> ${result.name} </#list>"
}
```

Resultaat:



**AUTOMATION,**

Nieuw In behandeling

Deze techniek heet slicing. Een cheatsheet:

Variabele	Waarde
body[0..1]	1e, 2e
body[1..3]	2e, 3e, 4e
body[2..<5]	3e, 4e, 5e
body[2..5]	3e, 4e, 5e, 6e
body[10..*3]	11e, 12e, 13e
body[2..]	3e t/m de laatste



## Lijsten: een waarde opzoeken in een lijst

De actiereeks om meldingen af te melden via een status is één van de meest gedownloade actiereeks in het actiereeks-archief van Laansloot IT. In twee stappen doet deze actiereeks in feite niets anders dan het veld Status veranderen.

Dezelfde actiereeks kun je net zo goed gebruiken om een andere status in te vullen, zoals "In behandeling". Dat is het scenario in onderstaand voorbeeld.

- In stap 1 haal je de statuses op met een GET-call
- In stap 2 zoek je in de lijst met statuses uit stap 1 de juiste status

### Body

```
{
  "processingStatus" :
  { "id" : "<#list _responses.getstatuses.body as status><#if status.name == "In behandeling">${status.id}</#if></#list>" }
}
```

Resultaat:

Status

In behandeling



Bovenstaande code met `<#list>` en `<#if>` laat TOPdesk dus zoeken in de lijst die terugkomt in stap 1. Is er een resultaat gevonden waarbij het element name de waarde "In behandeling" heeft? Dan pakt TOPdesk van dat resultaat het element id en zet de waarde ervan in stap 2.

NB1: De syntax luistert wederom erg nauw. Zet je hier een spatie tussen `${status.id}` en `</#if>`, dan werkt de actiereeks waarschijnlijk niet!

NB2: de syntax met `<#if>` wordt in dit hoofdstuk niet verder behandeld. Zie daarvoor hoofdstuk 6, "If-then-constructies in Freemarker".

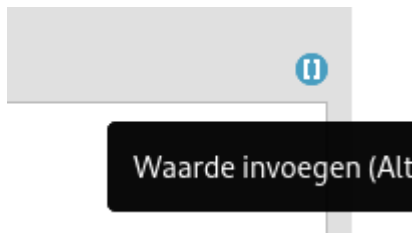
## Hoofdstuk 5

# Waarden en mappings

Als je met actiereeksen aan de gang gaat, dan krijg je met verschillende invoervelden te maken:

- Variabelen
- Naam
- URL
- Headers
- Body
- Aangepaste voorwaarde

In al die velden kun je variabelen invoegen met de optie "Waarde invoegen":



Hiermee kun je velden uit de TOPdesk-database, JSON-responsedata van eerdere stappen, en vooraf gedefinieerde variabelen oproepen. De mogelijkheden en bijzonderheden van deze optie "Waarde invoegen" is het onderwerp van dit hoofdstuk.

Ook aan de orde komen mappings. TOPdesk gebruikt hiervoor de term "Waardetoewijzingen". Deze functionaliteit maakt het mogelijk om geautomatiseerd bronwaarden te vervangen met doelwaarden, op basis van een sleutel die je opstelt.

## Waarde invoegen: Huidige waarden

### Tweedelijns melding

#### HUIDIGE WAARDEN

#### OUDE WAARDEN

#### VARIABELEN

#### PROGRESS TRAIL

#### REACTIEGEGEVENS

Via het tabblad "Huidige waarden" voeg je heel eenvoudig een bepaald veld toe aan je actiereeks. Dit werkt grofweg op dezelfde manier als bij emailsjablonen of HTTP-requests. Je zoekt gewoon het juiste veld op in de verschillende opzoeklijsten.

Er is echter één groot verschil: je krijgt meestal de UNID van een kaart terug in plaats van de naam. Dit is omdat je dan makkelijker een API-call kunt uitvoeren op de betreffende kaart, in een losse stap in je actiereeks.

De oproepbare velden die beginnen met "Naam" geven echter wel de naam van de kaart terug. In het dropdownmenu "Aanmelder" zie je bijvoorbeeld "Naam aanmelder" en "Naam vestiging":

- ▼ Aanmelder
  - 'Kostenplaats' (Aanmelder)
  - Aanmelder (geregistreerd)
  - Afdeling
  - E-mail (Aanmelder)
  - Naam aanmelder
  - Naam vestiging

Om het verschil te laten zien zet ik beide opties, "Naam vestiging" en "Vestiging (Aanmelder)", onder elkaar in een actie-update:

#### JSON-body

```
{
  "action": "Naam vestiging is ${ref_vestiging}<BR>Vestiging (Aanmelder) is  
${aanmeldervestigingid}."
```

### Resultaat:



#### AUTOMATION,

Naam vestiging is Ministerie van Serieuze Zaken

Vestiging (Aanmelder) is bd280bbf-79e0-4f70-8617-74fce2db4932.

## Waarde invoegen: Oude waarden

### Tweedelijns melding

HUDIGE WAARDEN

**OUDE WAARDEN**

VARIABELEN

PROGRESS TRAIL

REACTIEGEGEVENS

Het tabblad "Oude waarden" gebruik je wanneer je wijzigingen op velden wil loggen.

### Voorbeeld

Stel dat je in het actieveld wil registreren wanneer het veld "Categorie" wijzigt. De eerste stap is een gebeurtenis maken op "Wijzigen van de kaart", en specifiek op het veld "Categorie":

Gebeurtenis **Categorie gewijzigd**

Opslaan



Verwijderen

**Tweedelijns meldingen**

**Details**

Naam

Categorie gewijzigd

Actief

☒

Soort kiezen

Wijzigen kaart

**Omschrijving**

**Gekoppelde acties**

Categorie gewijzigd

#### Voorwaarden

Toepassen ☐

#### Specificaties wijzigen

Toepassen ☒

Gebeurtenis treedt op met enkele wijziging

Veld	Oude waarde	Nieuwe waarde
Categorie	Geen specifieke waarde	Geen specifieke waarde

Toevoegen

De tweede stap is om de actiereeks te maken en te koppelen aan de gebeurtenis. In je actie-update verwerk je dan zowel de "Oude waarde" van het veld "Categorie" als de "Huidige waarde" van het veld:

#### JSON-body

```
{
  "action": "De categorie is veranderd van ${_previous.incident_domeinid} naar ${incident_domeinid}."
}
```

Ga je nu de categorie wijzigen en daarna de kaart herladen? Dan zie je (als het goed is) een nieuwe actie-update:



#### AUTOMATION,

10 maart 2020 19:55

De categorie is veranderd van 6c40acaf-4f64-5373-839e-4b66ded9fe93 naar 02c2b593-b1fa-53eb-8576-9b1538438003.



Dit toont aan dat TOPdesk de vorige waarde van een veld kan "vangen" via het tabblad "Oude waarden", via een gebeurtenis en een actiereeks.

### Voorbeeld met namen

Zoals gezegd geeft de TOPdesk API standaard UNIDs terug. Wil je de *naam* van de categorie ophalen? Dan krijg je een complexere actiereeks met 2 stappen:

- een GET-call op het endpoint /tas/api/incidents/categories
- een PUT-call waarin je de naam uit de lijst met categorieën opzoekt met `<#list>` en `<#if>`

#### JSON-body

```
<#list _responses.getcategories.body as category>
<#if category.id == _previous.incident_domeinid>
<#assign oudecategorie = category.name></#if></#list>
<#list _responses.getcategories.body as category>
<#if category.id == incident_domeinid>
<#assign nieuwecategorie = category.name></#if></#list>

{
  "action": "De categorie is veranderd van ${oudecategorie} naar ${nieuwecategorie}."
}
```

#### Resultaat:



#### AUTOMATION,

De categorie is veranderd van Netwerk hardware naar FM Diensten.

De syntax met `<#list>`, `<#if>` en `<#assign>` wordt verder uitgelegd in hoofdstuk 4 "Werken met stappen", en in hoofdstuk 7 "Variabelen".

## Waarde invoegen: Variabelen

### Tweedelijns melding

HUDIGE WAARDEN






OUDE WAARDEN

**VARIABELEN**




PROGRESS TRAIL





REACTIEGEGEVENS

Het tabblad "Variabelen" gebruik je wanneer je variabelen wilt aanroepen die je bovenaan je actiereeks hebt gedefinieerd:

Naam	Waarde	
TOPdeskURL	https://locatietopdesk.net.com	 
TOPdeskLoginZichtbaar	QVWUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==	 
		

De URL en de authenticatie-string roep je bijvoorbeeld aan in de velden URL en Headers:

**Stap 1**   

Naam	getcategories
HTTP-methode	GET
URL	<code>\$_variables.TOPdeskURL?no_esc/tas/api/incidents/categories</code> 
<b>Headers</b>	
Authorization	BASIC <code>\$_variables.TOPdeskLoginZichtbaar</code>  
	
<b>Deze stap uitvoeren</b>	
Altijd	

Heb je een URL als variabele? Gebruik dan altijd de optie `?no_esc`. Doe je dat niet? Dan werkt je actiereeks waarschijnlijk niet.

## Waarde invoegen: Progress trail

### Tweedelijns melding

HUIDIGE WAARDEN

OUDE WAARDEN

VARIABELEN

**PROGRESS TRAIL**

REACTIEGEGEVENS

Het tabblad "Progress trail" gebruik je om data uit het zoekveld of het actieveld op te halen. Deze velden zijn bijzonder vanwege de volgende redenen:

- Ze kunnen bestaan uit *meerdere* losse entries. Je kunt tenslotte altijd een nieuwe actie-update erbij plaatsen.
- Alle entries hebben extra eigenschappen:
  - een datum/tijd van aanmaak
  - een aanmaker van de kaart
  - "zichtbaar" of "onzichtbaar" voor de aanmelder
- Ze kunnen extra opmaakopties hebben (vetgedrukt, schreef, onderstreept)
- Ze kunnen worden aangemaakt vanuit zowel de Selfserviceportal als het behandelaarsgedeelte.

Vanwege deze bijzonderheden geeft TOPdesk een nieuw scherm met allerlei opties.

### Voorbeeld: enkele waarde invoegen

Wil je de tekst van de laatste actie-update die zichtbaar is voor de aanmelder, en dan zonder opmaak? Dan kies je de volgende opties:

Aanvraag		Actie	
Alleen laatste vermelding		Lijst van alle vermeldingen	
Zichtbaar voor aanmelder	Onzichtbaar voor aanmelder	Alle	

Vermelding (platte tekst)
Vermelding (tekst met opmaak)
Tijdstempel
Aanmaakdatum
Naam aanmaker
ID van behandelaar
Naam van behandelaar
ID van persoon
Naam van persoon

Dat geeft de volgende code:

```
${(_progresstrail.actions_visible?first.plaintext)!}
```

Die code kun je weer in je JSON-body plaatsen:

#### JSON-body

```
{
  "action": "De laatste zichtbare actie-update is ${(_progresstrail.actions_visible?first.plaintext!)}"
}
```

Resultaat:



#### AUTOMATION,

De laatste zichtbare actie-update is Gevonden, bij de deur!



#### Visser, Rogier

Gevonden, bij de deur!

Je ziet dat de laatste actie-update "Gevonden bij de deur!" was. Die heeft de actiereeks opgepakt en verwerkt in een nieuwe actie-update.

### Voorbeeld: lijst invoegen

Een stuk complexer wordt het wanneer je data uit meerdere entries wil invoegen. De volgende opties zorgen ervoor dat je van alle actie-updates de naam van de behandelaar en de ingevoegde tekst oproept:

Aanvraag	Actie	
Alleen laatste vermelding	Lijst van alle vermeldingen	
Zichtbaar voor aanmelder	Onzichtbaar voor aanmelder	Alle

<input checked="" type="checkbox"/>	Vermelding (platte tekst)
<input type="checkbox"/>	Vermelding (tekst met opmaak)
<input type="checkbox"/>	Tijdstempel
<input type="checkbox"/>	Aanmaakdatum
<input type="checkbox"/>	Naam aanmaker
<input type="checkbox"/>	ID van behandelaar
<input checked="" type="checkbox"/>	Naam van behandelaar
<input type="checkbox"/>	ID van persoon
<input type="checkbox"/>	Naam van persoon

Dat geeft de volgende code:

```
<#list _progresstrail.actions as action> ${action.plaintext!} ${action.operator.name!}</#list>
```



Hieronder zie je twee actie-updates uit een testticket. Deze gaan we ophalen en vervolgens wegschrijven in de actiereeks:

---

**RV Visser, Rogier**  
Gevonden, bij de deur!

---

**RV Visser, Rogier**  
De kabelbreuk zit ergens onder de grond.

De ontvangen code (zie vorige pagina) plaatsen we in de JSON-body:

#### JSON-body

```
{  
  "action": "Dit zijn de actie-updates: <#list _progresstrail.actions as action>${action.plaintext!}  
  ${action.operator.name!}</#list>"  
}
```

Resultaat:

---

**A AUTOMATION,** 11 maart 2020 20:31  
Dit zijn de actie-updates: Gevonden, bij de deur! Visser, RogierDe kabelbreuk zit ergens onder de grond. Visser, Rogier



Het resultaat is waarschijnlijk niet optimaal. Om dit prettig leesbaar te maken moeten we sleutelen aan de ontvangen code:

- Nieuwe regels toevoegen op de juiste plekken, met de tag <br>
- Volgorde aanpassen: eerst de naam van de behandelaar, en daarna de tekst van de actie-update
- De naam van de behandelaar vetgedrukt, met de tag <b>

De aangepaste code is als volgt:

```
<br><#list _progresstrail.actions as action><br><b>${action.operator.name!}</b><br>  
${action.plaintext!}</#list>
```

Die code zetten we weer in de JSON-body:

#### JSON-body

```
{
  "action": "Dit zijn de actie-updates: <br> <#list _progresstrail.actions as action> <br> <b>${action.operator.name!}</b> <br> ${action.plaintext!}</#list>"
}
```

Resultaat:



#### AUTOMATION,

Dit zijn de actie-updates:

**Visser, Rogier**

Gevonden, bij de deur!

**Visser, Rogier**

De kabelbreuk zit ergens onder de grond.

### Opmaak-opties

In het bovenstaande voorbeeld voeren we opmaak-opties door via HTML-tags <b> en <br>. Andere mogelijke tags zijn <i> voor scheve tekst, en <u> voor onderstreepte tekst. **Dit werkt echter alleen in de meldingenmodule, en alleen in het verzoek- of actieveld!**

In de wijzigingsmodule is het niet mogelijk om opmaak-opties mee te geven. De enige uitzondering daarop is een nieuwe regel toevoegen. Daarvoor gebruik je \n:

#### Nieuwe regel toevoegen, in een verzoek- of actieveld

Meldingenmodule	 
Wijzigingsmodule	\n

In de wijzigingsmodule kan dat er zo uitzien:

#### JSON-body

```
[
{
  "op": "add",
  "path": "/progressTrail",
  "value": "Dit zijn de actie-updates: \n<#list _progresstrail.actions as action> \n${action.operator.name!}\n${action.plaintext!}</#list>"
}
]
```

## Waarde invoegen: Reactiegegevens

### Tweedelijns melding

HUDIGE WAARDEN

OUDE WAARDEN

VARIABELEN

PROGRESS TRAIL

**REACTIEGEGEVENS**

Het laatste tabblad "Reactiegegevens" is alleen zichtbaar vanaf stap 2 in een actiereeks. Je kunt hiermee de JSON-responsedata van een eerdere stap ophalen.

### Voorbeeld

Wil je de naam van de eerste categorie in de lijst met categorieën ophalen? Dan kies je "Reactiebody als JSON", en vervolgens typ je `[0].name`.

#### Stap die de reactie gaf

getcategories

#### Reactiewaarde

Reactiestatuscode

Reactiebody als tekst

Reactiebody als JSON

Specificeer welke JSON-waarde u wilt ontvangen van de reactiebody:

`[0].name`

#### Waardetoewijzing

Geen waardetoewijzing

#### Voorbeeld

`${_responses.getcategories.body[0].name}`

### Resultaat:



#### AUTOMATION,

De eerste categorie is Besturingssystemen

Hoe dit precies werkt, en welke syntax je moet gebruiken, dat is gedocumenteerd in het hoofdstuk 4 "Werken met stappen".

## Waardetoewijzingen

Het laatste onderwerp is hoe je geautomatiseerd bronwaarden vervangt met doelwaarden. Dat regel je op het tweede tabblad in de actiereeks, "Waardetoewijzingen".

Onderstaand is het voorbeeld dat op de volgende pagina verder wordt uitgewerkt:

### Actiereeks **Categorieën mappen**

ALGEMEEN **WAARDETOEWIJZINGEN** UITVOERINGSLOGS

#### mapping

Verwijderen

Naam

mapping

#### Toewijzingen

Invoerwaarde

Besturingssystemen



Uitvoerwaarde

BS



Applicaties



AP



Netwerk hardware



NW



Werkplek hardware



WH



#### Andere invoerwaarden

##### ☒ Niets doen

Iedere andere invoerwaarde



Uitvoerwaarde is gelijk aan invoerwaarde

##### ☐ Een standaard uitvoerwaarde gebruiken

Iedere andere invoerwaarde



Een set toevoegen

## Voorbeeld

Het idee is dat TOPdesk het veld "Categorie" afkort volgens een bepaalde sleutel. Vindt TOPdesk de waarde "Netwerk hardware"? Dan moet de actiereeks de waarde "NW" gaan gebruiken. Links staan dus de bronwaarden, en rechts de doelwaarden.

De naam van de categorie haal ik op in een eerdere stap, getmelding, en specifiek in het element category.name. Via "Waarde invoegen", tabblad "Reactiegegevens" vul ik die gegevens in:

### Stap die de reactie gaf

getmelding

### Reactiewaarde

Reactiestatuscode

Reactiebody als tekst

Reactiebody als JSON

Specificeer welke JSON-waarde u wilt ontvangen van de reactiebody:

.category.name

### Waardetoewijzing

Geen waardetoewijzing

### Voorbeeld

`${_responses.getmelding.body.category.name}`

De truc nu is om de waardetoewijzing hierop toe te laten passen:

### Waardetoewijzing

Geen waardetoewijzing

Geen waardetoewijzing

mapping

`${_responses.getmelding.body.category.name}`

Resultaat:

`${_map(_responses.getmelding.body.category.name, "mapping")}`

Die string komt vervolgens in de JSON-body:

#### JSON-body

```
{  
  "action": "De afgekorte categorie is ${_map(_responses.getmelding.body.category.name,  
"mapping")}"  
}
```

Resultaat:



#### AUTOMATION,

De afgekorte categorie is NW

### Andere waarden

Heb je nu een bronwaarde die niet gemapped is naar een doelwaarde? Dan kun je daar ofwel een standaardwaarde invullen, ofwel niets doen. Dat regel je onderaan de pagina voor "Waardetoewijzingen":

#### Andere invoerwaarden

☒ **Niets doen**

Iedere andere invoerwaarde

→ Uitvoerwaarde is gelijk aan invoerwaarde

☐ **Een standaard uitvoerwaarde gebruiken**

Iedere andere invoerwaarde

→

## Hoofdstuk 6

# If-then-constructies in Freemarker

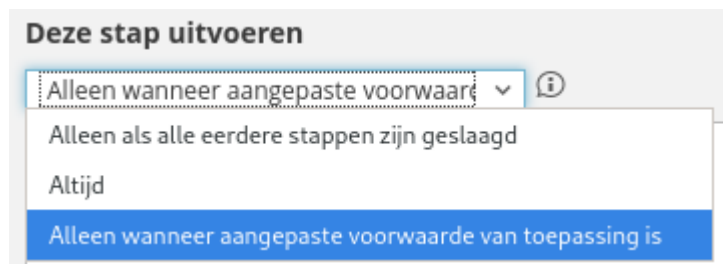
In hoofdstuk 3 "Je eerste GET-call" en hoofdstuk 4 "Werken met stappen" is uitgelegd hoe je een GET-call kunt doen op TOPdesk, en hoe je de responsedata daarvan kunt gebruiken in vervolgstappen binnen de actiereeks:

In dit hoofdstuk gaan we een stapje verder, en kijken we hoe we logische regels kunnen opstellen binnen een actiereeks. Dit doen we met Freemarker.

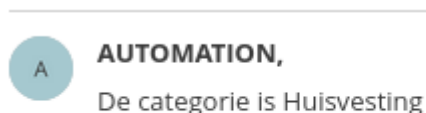
Freemarker maakt het mogelijk om actiereeksen nog specifiek te maken, zodat ze in nog meer situaties taken kunnen automatiseren. Enige programmeerervaring helpt wel bij het werken met Freemarker.

## If-then-constructie in "Aangepaste voorwaarde"

Bij het maken van stappen binnen een actiereeks zie je linksonder het veld "Deze stap uitvoeren". In dit opzoeklijstje staat de optie "Alleen wanneer aangepaste voorwaarde van toepassing is":



Met deze optie kun je regelen wanneer de stap uitgevoerd of overgeslagen moet worden. Hier kun je bijvoorbeeld instellen dat de volgende actie-update verschijnt, wanneer de categorie gelijk is aan "Huisvesting":



Is de categorie niet gelijk aan "Huisvesting"? Dan moet deze stap in de actiereeks niet worden uitgevoerd. Bovenstaande actie-update verschijnt in dat geval dan niet. Dit is een zogeheten If-then-constructie.

De actiereeks die hiervoor zorgt is als volgt opgebouwd:

1. een GET-call om de data van de melding op te halen
2. een PUT-call om de actie-update te plaatsen
  - Aangepaste voorwaarde: alleen uitvoeren wanneer de categorie van de melding (in de responsedata van stap 1) gelijk is aan "Huisvesting"

De syntax zorgt voor de aangepaste voorwaarde:

```
<#if _responses.getincident.body.category.name == "Huisvesting">true</#if>
```

Er is ook een alternatieve syntax. In de rest van dit hoofdstuk zet ik de twee verschillende manieren direct onder elkaar.

```
${_responses.getincident.body.category.name == "Huisvesting"}
```

Zie de volgende pagina voor een screenshot van de gehele actiereeks.



## Actiereeks met If-then-constructie

**Stap 1** ↑ ↓ Verwijderen

Naam

HTTP-methode

URL  ⓘ

**Headers**

ⓘ ⓘ

+

**Deze stap uitvoeren**

**Stap 2** ↑ ↓ Verwijderen

Naam

HTTP-methode

URL  ⓘ

**Headers**

ⓘ

ⓘ ⓘ

+

**Body**

☒ JSON-escaping toepassen op ingevoegde waarden ⓘ ⓘ

```
{  
  "action" : "De categorie is Huisvesting"  
}
```

**Deze stap uitvoeren**

ⓘ ⓘ

```
<#if _responses.getincident.body.category.name == "Huisvesting">true</#if>
```

## Logische functies

De actiereeks op de vorige pagina controleert of een bepaald element gelijk is aan een bepaalde waarde (element `_responses.getincident.body.category.name`, waarde Huisvesting). Deze "Gelijk aan" functie roep je aan met de syntax `==` .

De belangrijkste functies, met syntax, op een rij:

### Gelijk aan: `==`

```
<#if _responses.getincident.body.category.name == "Huisvesting">true</#if>
${_responses.getincident.body.category.name == "Huisvesting"}
```

### Ongelijk aan: `!=`

```
<#if _responses.getincident.body.category.name != "Huisvesting">true</#if>
${_responses.getincident.body.category.name != "Huisvesting"}
```

### Gelijk aan A of B: `||`

```
<#if _responses.getincident.body.category.name == "Huisvesting" ||
_responses.getincident.body.category.name == "Diensten">true</#if>
${_responses.getincident.body.category.name == "Huisvesting" ||
_responses.getincident.body.category.name == "Diensten"}
```

### Gelijk aan A en B: `&&`

```
<#if _responses.getincident.body.category.name == "Huisvesting" &&
_responses.getincident.body.subcategory.name == "Lift installaties">true</#if>
${_responses.getincident.body.category.name == "Huisvesting" &&
_responses.getincident.body.subcategory.name == "Lift installaties"}
```

### Als element bestaat: `??`

```
<#if _responses.getincident.body.category.name??>true</#if>
${_responses.getincident.body.category.name??}
```

### Als element bestaat en is ingevuld: `?has_content`

```
<#if _responses.getincident.body.category.name?has_content>true</#if>
${_responses.getincident.body.category.name?has_content}
```

## EN-operator && en OF-operator || combineren met haakjes

De syntax van Freemarker maakt het ook mogelijk om EN && en OF || te combineren. Stel dat de stap alleen uitgevoerd moet worden wanneer:

- Categorie is gelijk aan "Huisvesting" EN
- Subcategorie is gelijk aan
  - "Lift Installaties" OF
  - "Sanitair"

Dit kun je instellen door de twee OF-voorwaardes te bundelen binnen haakjes:

```
<#if _responses.getincident.body.category.name == "Huisvesting" &&  
    (_responses.getincident.body.subcategory.name == "Lift installaties" ||  
    _responses.getincident.body.subcategory.name == "Sanitair")>true</#if>  
${  
    _responses.getincident.body.category.name == "Huisvesting" &&  
    (_responses.getincident.body.subcategory.name == "Lift installaties" ||  
    _responses.getincident.body.subcategory.name == "Sanitair")}
```

Deze functionaliteit is waardevol, want dit is niet goed mogelijk wanneer je vergelijkbare logica probeert te bouwen in de gebeurtenisverkenner:

### Voorwaarden

Toepassen



Gebeurtenis treedt op als



Alle voorwaarden zijn van toepassing (AN



Kenmerk

Waarde

Categorie is gelijk aan



Huisvesting



[Verwijderen](#)

en

Subcategorie is gelijk aan



Huisvesting - Lift installaties



[Verwijderen](#)

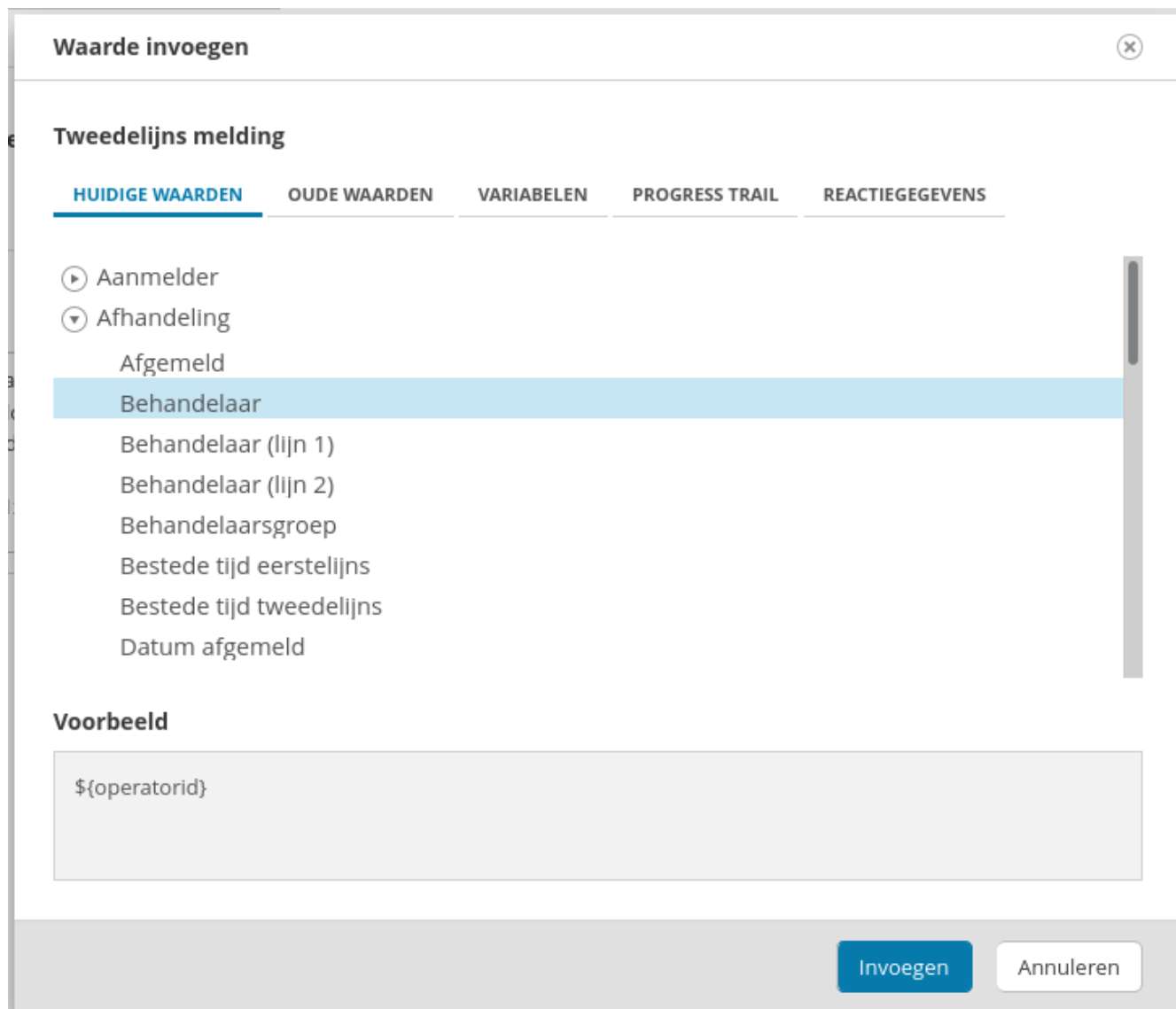
[Toevoegen](#)

Hier loop je vast, omdat je niet Subcategorie A *OF* Subcategorie B kunt kiezen, in combinatie met een bepaalde categorie. De workaround binnen de gebeurtenisverkenner is om hier twee losse gebeurtenissen voor aan te maken, voor beide subcategorieën.

## TOPdesk-velden in aangepaste voorwaarde

Het invoerveld voor "Aangepaste voorwaarde" heeft ook de mogelijkheid om direct een waarde uit de TOPdesk-database in te voeren, met het knopje "[ ] -Waarde invoegen- rechtsboven.

Je kunt bijvoorbeeld de code `${operatorid}` gebruiken om direct de behandelaar van de kaart op te zoeken:



**Waarde invoegen**

**Tweedelijns melding**

**HUIDIGE WAARDEN**   OUDE WAARDEN   VARIABLEN   PROGRESS TRAIL   REACTIEGEGEVENS

- ▶ Aanmelder
- ▼ Afhandeling
  - Afgemeld
  - Behandelaar**
  - Behandelaar (lijn 1)
  - Behandelaar (lijn 2)
  - Behandelaarsgroep
  - Bestede tijd eerstelijns
  - Bestede tijd tweedelijns
  - Datum afgemeld

**Voorbeeld**

`${operatorid}`

**Invoegen**   **Annuleren**

Maar let op! Met deze variabelen roep je de UNID op, en niet de naam van de kaart.

Toch kun je ook met de UNID in een aantal scenarios uit de voeten. Je kunt bijvoorbeeld controleren of er een behandelaar is ingevuld.

Let trouwens op de syntax! Binnen de accolades {} of haken <> gebruik je niet nogmaals de accolades. Dus daarom wordt \${operatorid} dan operatorid in onderstaande voorbeelden.

### Als veld is ingevuld: ?has\_content

```
<#if operatorid?has_content>true</#if>
${operatorid?has_content}
```

Daarnaast kun je ook logische regels opstellen als je de UNID al weet. In hoofdstuk 2 "Tips & tricks" staat hoe je de UNID van een kaart kunt achterhalen.

### Gelijk aan: ==

```
<#if operatorid == "a74e5723-6143-564d-836f-8965c6099c38">true</#if>
${operatorid == "a74e5723-6143-564d-836f-8965c6099c38"}
```

### Ongelijk aan: !=

```
<#if operatorid != "a74e5723-6143-564d-836f-8965c6099c38">true</#if>
${operatorid != "a74e5723-6143-564d-836f-8965c6099c38"}
```

Verdere syntax is gelijk aan wat eerder is uitgelegd, met de EN-operator &&, de OF-operator ||, en de haakjes.

### Bronnen combineren

Je kunt ook de responsedata uit een eerdere stap combineren met een variabele uit de TOPdesk-database, binnen de aangepaste voorwaarde:

```
<#if _responses.getincident.body.category.name == "Huisvesting" &&
operatorid == "a74e5723-6143-564d-836f-8965c6099c38">true</#if>
${_responses.getincident.body.category.name == "Huisvesting" &&
operatorid == "a74e5723-6143-564d-836f-8965c6099c38"}
```

## If-then-constructie in de JSON-body

Met If-then-constructies in de JSON-body kun je heel fijnmazig instellen welke data in de body terechtkomt. Hieronder is als voorbeeld een actie-update met de categorie en de subcategorie:



### AUTOMATION,

De categorie is Huisvesting en de subcategorie is Lift installaties

Maar als de subcategorie niet is ingevuld op de kaart, dan willen we alleen de naam van de categorie:



### AUTOMATION,

De categorie is Huisvesting

Dit kan met een If-then-constructie in de JSON-body, met de optie ?has\_content:

**Body**

☒ JSON-escaping toepassen op ingevoegde waarden ⓘ ⓘ

```
{
  "action" : "De categorie is ${_responses.getincident.body.category.name}<#if
_responses.getincident.body.subcategory?has_content> en de subcategorie is
${_responses.getincident.body.subcategory.name}</#if>"
}
```

De gele gehighlightte code is de If-then-constructie, en de blauw gehighlightte code is de waarde die we invoegen wanneer de expressie Waar is.

```
}
"action" : "De categorie is ${_responses.getincident.body.category.name}<#if
_responses.getincident.body.subcategory?has_content> en de subcategorie is
${_responses.getincident.body.subcategory.name}</#if>"
}
```

Je kunt ook hele elementen binnen- of buitensluiten met een If-then-constructie. In onderstaande JSON-body wordt een behandelaar ingevuld, maar alleen wanneer de subcategorie gelijk is aan "Sanitair":

**Body**

☒ JSON-escaping toepassen op ingevoegde waarden ⓘ ⓘ

```
{
  "action" : "De categorie is ${_responses.getincident.body.category.name}"<#if
_responses.getincident.body.subcategory.name == "Sanitair">,
  "operator" : { "id" : "a74e5723-6143-564d-836f-8965c6099c38" }</#if>
}
```

```
{
  "action" : "De categorie is ${_responses.getincident.body.category.name}"<#if
_responses.getincident.body.subcategory.name == "Sanitair">,
  "operator" : { "id" : "a74e5723-6143-564d-836f-8965c6099c38" }</#if>
}
```

Is de subcategorie gelijk aan "Sanitair", dan wordt de volgende JSON gegenereerd:

```
Request body: {
  "action" : "De categorie is Huisvesting",
  "operator" : { "id" : "a74e5723-6143-564d-836f-8965c6099c38" }
}
```

Is de subcategorie ongelijk aan "Sanitair", dan is de JSON korter:

```
Request body: {
  "action" : "De categorie is Huisvesting"
}
```

### Let op de komma!

De komma die als eerste blauw gehighlight is in de screenshot, is heel belangrijk. Staat die komma er niet, dan is de JSON niet geldig en geeft TOPdesk een error:

```
Request body: {
  "action" : "De categorie is Huisvesting"
  "operator" : { "id" : "a74e5723-6143-564d-836f-8965c6099c38" }
}
```

[2020-02-15T21:08:20.712918Z] Response status: 400 Bad Request  
Response headers: {Date=[Sat, 15 Feb 2020 21:08:20 GMT], X-Content-Type-Options=[nosniff], X-XSS-Protection=[0], Cache-Control=[no-store], Content-Type=[application/json; charset=utf-8], Content-Length=[100]}  
Response body: [{"message":"Request cannot be parsed."}]

## Uitgebreide If-then-constructies

Onderstaande JSON-body maakt gebruik van drie verschillende If-then-constructies die door elkaar heen lopen. Dit zijn geneste If-then-constructies:

### JSON-body

```
{
  "action": "De categorie is ${_responses.getincident.body.category.name}"
  <#if _responses.getincident.body.category.name == "Huisvesting">
  <#if _responses.getincident.body.subcategory.name == "Lift installaties">,
  "subcategory": { "name": "Sanitair" } </#if>
  <#if _responses.getincident.body.subcategory.name == "Sanitair">,
  "subcategory": { "name": "Lift installaties" } </#if> </#if>
}
```

Hier staat eigenlijk:

1. Plaats een actie-update
2. Is de categorie gelijk aan "Huisvesting"?
  - Is de subcategorie gelijk aan "Lift installaties"?
    - Verander dan de subcategorie naar "Sanitair"
  - Is de subcategorie gelijk aan "Sanitair"?
    - Verander dan de subcategorie naar "Lift installaties"
3. Is de categorie ongelijk aan "Huisvesting"? Verander dan de subcategorie *niet*.

Met geneste If-then-constructies kun je dus heel precies regelen wat je aanpast in TOPdesk en onder welke voorwaarden.

### If-elseif

In bovenstaande code gebruiken we drie If-then-constructies om de subcategorie aan te wijzen. Hier kun je ook twee If-then-constructies van maken, door gebruik te maken van de optie <#if><#elseif></#if> :

### JSON-body

```
{
  "action": "De categorie is ${_responses.getincident.body.category.name}"
  <#if _responses.getincident.body.category.name == "Huisvesting">
  <#if _responses.getincident.body.subcategory.name == "Lift installaties">,
  "subcategory": { "name": "Sanitair" }
  <#elseif _responses.getincident.body.subcategory.name == "Sanitair">,
  "subcategory": { "name": "Lift installaties" } </#if> </#if>
}
```

Dit is ook een geneste If-then-constructie, en de uitwerking is precies gelijk als in de eerdere variant. Maar als je goed kijkt zie je dat er maar twee <#if>-tags zijn.



## If-else

Iets vergelijkbaars is aan de hand in de volgende JSON. Hier is echter de `<#elseif>`-tag vervangen door een `<#else>`-tag:

### JSON-body

```
{
  "action": "De categorie is ${_responses.getincident.body.category.name}"
  <#if _responses.getincident.body.category.name == "Huisvesting">
  <#if _responses.getincident.body.subcategory.name == "Lift installaties">,
  "subcategory": { "name": "Sanitair" }
  <#else>,
  "subcategory": { "name": "Lift installaties" } </#if> </#if>
}
```

Deze actiereeks verandert de subcategorie **altijd** in "Lift installaties". In de vorige twee voorbeelden deed TOPdesk dat alleen in het geval van subcategorie "Sanitair".

## If-elseif-else

Als je te maken hebt met drie of meerdere variabelen op basis waarvan je vervolgstappen wilt uitzetten, dan gebruik je `<#if>`, `<#elseif>` en `<#else>`. In onderstaand voorbeeld is sprake van drie subcategorieën.

### JSON-body

```
{
  "action": "De categorie is ${_responses.getincident.body.category.name}"
  <#if _responses.getincident.body.category.name == "Huisvesting">
  <#if _responses.getincident.body.subcategory.name == "Lift installaties">,
  "subcategory": { "name": "Sanitair" }
  <#elseif _responses.getincident.body.subcategory.name == "Sanitair">,
  "subcategory": { "name": "Lift installaties" }
  <#else>,
  "subcategory": { "name": "Verlichting" } </#if> </#if>
}
```

De logica in deze actiereeks is een combinatie van de vorige voorbeelden:

1. Plaats een actie-update
2. Is de categorie gelijk aan "Huisvesting"?
  - Is de subcategorie gelijk aan "Lift installaties"?
    - Verander dan de subcategorie naar "Sanitair"
  - Is de subcategorie gelijk aan "Sanitair"?
    - Verander dan de subcategorie naar "Lift installaties"
  - Is de subcategorie niet "Lift installaties" en ook niet "Sanitair"?
    - Verander dan de subcategorie naar "Verlichting"
3. Is de categorie ongelijk aan "Huisvesting"? Verander dan de subcategorie *niet*.

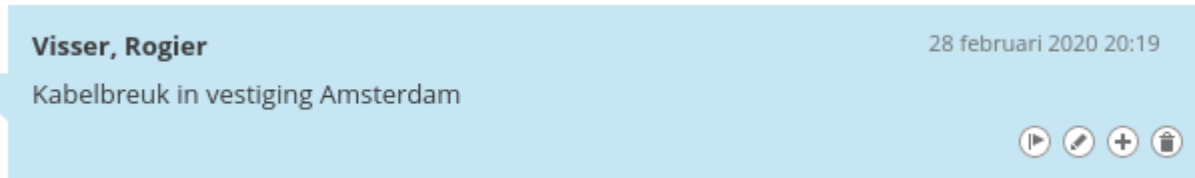
Wil je meer dan 3 variabelen opgeven? Dan kun je meerdere `<#elseif>`-expressies inbouwen, op dezelfde manier als hierboven.

## ?contains-functie

Het laatste kopje in dit hoofdstuk gaat over de functie ?contains. Met deze functie kun je sleutelwoorden definiëren en daarop functionaliteit bouwen.

- Is de zoekterm gevonden? Dan voert TOPdesk de stap in de actiereeks uit
- Is de zoekterm niet gevonden? Dan voert TOPdesk de stap in de actiereeks *niet* uit

Onderstaande melding gaat over een kabelbreuk:



"Kabelbreuk" is de zoekterm die we opgeven in de "Aangepaste voorwaarde":

### Aangepaste voorwaarde

```
<#if _responses.getincident.body.request?contains("Kabelbreuk")>true</#if>
```

Als de zoekterm is gevonden in het zoekveld, dan wordt de volgende JSON-body verstuurd:

### JSON-body

```
{  
  "action": "Het sleutelwoord is gevonden!"  
}
```

Resultaat:



In dit voorbeeld wordt een actie-update geplaatst, maar op dezelfde manier kun je ook een melding routeren naar een bepaalde behandelaarsgroep, of een prioriteit invullen, op basis van de zoekterm. De mogelijkheden zijn eindeloos.

NB1: de ?contains-functie zoekt altijd hoofdlettergevoelig.

## Hoofdstuk 7

# Freemarker en variabelen

Binnen actiereeksen zijn er verschillende manieren om variabelen op te roepen. Dat doe je met de syntax `${bronveld}`. Die variabelen kunnen verschillende bronnen hebben:

- Uit de responsedata van een eerdere stap
  - `${_responses.getincident.body.category.name}` roept het veld `category, name` op in de body van de stap “getincident” .
- Direct uit de TOPdesk-database.
  - `${operatorid}` roept de behandelaar op in de meldingenmodule.
- Vanuit de “Variabelen” bovenaan in de actiereeks
  - `${_variables.TOPdeskLoginZichtbaar}` in onderstaande screenshot:

Naam	Waarde
TOPdeskURL	https://locatietopdesk.net.com
TOPdeskLoginZichtbaar	QVVUT01BVEIPTjpbob3FwYy10ejV6ci16NWpvdy12dTV5Zi1vYnUzaw==

- Of je maakt ze helemaal zelf in Freemarker, met de syntax `<#assign>`. Dit hoofdstuk beschrijft hoe je dat doet, en wat je ermee kunt bereiken.

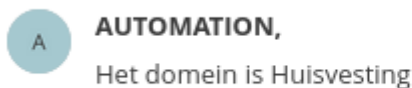
## Een variabele toewijzen met <#assign>

In onderstaand voorbeeld maken we een variabele domein, met de waarde "Huisvesting". Dit zetten we bovenaan in de JSON-body van de stap in de actiereeks:

### JSON-body

```
<#assign domein = "Huisvesting">
{
  "action": "Het domein is ${domein}"
}
```

Bij het uitvoeren vindt TOPdesk de waarde "Huisvesting" bij de variabele \${domein}:



De syntax met <#assign> kun je zowel in de JSON-body, als in de "Aangepaste voorwaarde" gebruiken:

### Aangepaste voorwaarde

```
<#assign domein = "Huisvesting">
<#if _responses.getincident.body.category.name == domein>true</#if>
<#assign domein = "Huisvesting">
${_responses.getincident.body.category.name == domein}
```

## Let op de dubbele aanhalingstekens

Dubbele aanhalingstekens binnen de Freemarker-expressies zijn belangrijk! Dat is alles binnen de <> en binnen de {}.

- Geen dubbele aanhalingstekens betekent: TOPdesk interpreteert het woord als een variabele (hier: domein)
- Wel dubbele aanhalingstekens betekent: TOPdesk interpreteert het woord als een waarde (hier: "Huisvesting")

## Meerdere <#assign>

Onderstaande "Aangepaste voorwaarde" bevat logica die controleert of een behandelaarsaccount wel dezelfde loginnaam heeft als een persoonskaart:

### Aangepaste voorwaarde

```
<#assign personloginname = tasloginnaam>  
<#assign operatorloginname = _responses.getoperator.body[0].loginName>  
<#if personloginname == operatorloginname>true</#if>  
  
<#assign personloginname = tasloginnaam>  
<#assign operatorloginname = _responses.getoperator.body[0].loginName>  
${personloginname == operatorloginname}
```

Je ziet dat er twee variabelen staan, personloginname en operatorloginname. Komen de twee variabelen overeen? Dan mag de stap uitgevoerd worden. Je kunt dus prima meerdere variabelen met <#assign> aanmaken.

Bovenstaande Freemarker-expressie kun je overigens eenvoudiger maken. In de eerste plaats kun je de variabele personloginname weglaten:

### Aangepaste voorwaarde

```
<#assign operatorloginname = _responses.getoperator.body[0].loginName>  
<#if tasloginnaam == operatorloginname>true</#if>  
  
<#assign operatorloginname = _responses.getoperator.body[0].loginName>  
${tasloginnaam == operatorloginname}
```

Vervolgens kun je ook de variabele operatorloginname weglaten, zodat je helemaal geen syntax met <#assign> hebt:

### Aangepaste voorwaarde

```
<#if tasloginnaam == _responses.getoperator.body[0].loginName>true</#if>  
  
${tasloginnaam == _responses.getoperator.body[0].loginName}
```

## Gereserveerde variabelen vanuit TOPdesk

Het voorbeeld op voorgaande pagina bevatte een check of een behandelaarsaccount dezelfde loginnaam heeft als een persoonskaart:

### Aangepaste voorwaarde

```
<#if tasloginnaam == _responses.getoperator.body[0].loginName>true</#if>
```

```
${tasloginnaam == _responses.getoperator.body[0].loginName}
```

De eerste variabele, tasloginnaam, komt direct uit de TOPdesk-database, en daar gaat het nu even om. Met het knopje "[ ]" (Waarde invoegen) kun je namelijk dezelfde variabele tasloginnaam invoegen:

**Waarde invoegen**

**Persoon**

**HUIDIGE WAARDEN** | OUDE WAARDEN | VARIABELEN | REACTIEGEGEVENS

- Personenkaart
- ▶ Informatie
- ▼ Inlognaam
  - Inlognaam mainframe
  - Inlognaam netwerk
  - Inlognaam selfserviceportal**
- ▶ Kaartgegevens
- ▶ Leaseauto
- ▶ Locatie
- ▶ Manager
- ▶ Manager mag aanvragen autoriseren van

**Voorbeeld**

`${tasloginnaam}`

Ga je zelf variabelen toevoegen met de syntax `<#assign>`? Zorg er dan voor dat de naam die je kiest niet al gereserveerd is door TOPdesk. Met andere woorden: gebruik geen namen die je ook via het knopje "Waarde invoegen" kunt invoegen.

## Variabele zoeken met een If-then-constructie

Onderstaande JSON-body komt uit een actiereeks die een emailadres wegschrijft naar een vrij veld. De If-then-constructie erin bepaalt welke waarde precies wordt toegewezen aan de variabele `${sender}`:

### JSON-body

```
<#if _responses.getprogresstrail.body[0].sender?has_content>
<#assign sender = _responses.getprogresstrail.body[0].sender>
<#else>
<#assign sender = _responses.getprogresstrail.body[1].sender> </#if>

{
  "optionalFields2" : { "text1" : "${sender}" }
}
```

De logica is als volgt:

- Heeft de variabele `_responses.getprogresstrail.body[0].sender` een waarde?
  - `${sender}` wordt gelijk aan `_responses.getprogresstrail.body[0].sender`.
- Heeft de variabele `_responses.getprogresstrail.body[0].sender` geen waarde?
  - `${sender}` wordt gelijk aan `_responses.getprogresstrail.body[1].sender`.

Met deze oplossing kun je dus één variabele `${sender}` toewijzen, terwijl je op twee plekken zoekt naar de juiste waarde.

### Aangepaste voorwaarde

Zorg er wel voor dat je in de "Aangepaste voorwaarden" afdwingt dat ofwel `_responses.getprogresstrail.body[0].sender` ofwel `_responses.getprogresstrail.body[1].sender` daadwerkelijk bestaat. Anders is de logica in de JSON-body niet correct en werkt de actiereeks ook niet correct.

### Aangepaste voorwaarde

```
<#if _responses.getprogresstrail.body[0].sender?has_content ||
_responses.getprogresstrail.body[1].sender?has_content>true</#if>

${
  _responses.getprogresstrail.body[0].sender?has_content ||
  _responses.getprogresstrail.body[1].sender?has_content}
```

## Variabele zoeken met een lijst

Onderstaande JSON-body komt uit een actiereeks die een melding automatisch sluit.

### JSON-body

```
<#list _responses.getstatuses.body as status><#if status.name == "Gesloten">
<#assign nieuwestatus = status.id></#if></#list>

{
  "processingStatus" : { "id" : "${nieuwestatus}" }
}
```

De code met `<#list>`, `<#if>` en `<#assign>` in de JSON-body vertaalt de naam van de status "Gesloten" naar de UNID "935bd773-fcc8-5edb-b037-99ad32ce62b4". Die UNID komt vervolgens in de variabele `${nieuwestatus}`:

Request body:

```
{
  "processingStatus" : { "id" : "935bd773-fcc8-5edb-b037-99ad32ce62b4" }
}
```

### De logica stap voor stap

De variabele `${nieuwestatus}` wordt gezet op de volgende manier:

1. In een eerdere stap (een GET-call op het endpoint `/tas/api/incidents/statuses`) worden alle mogelijke statuses van een melding opgeroepen: `getstatuses`.
2. `<#list>`: De responsedata van de stap `getstatuses` is een lijst. TOPdesk loopt de resultaten in de lijst één voor één na.
  - `<#if>`: Is het element name gelijk aan "Gesloten"?
    - `<#assign>`: `${nieuwestatus}` wordt gemaakt met waarde `id`.
  - `<#if>`: Is het element name niet gelijk aan "Gesloten"?
    - TOPdesk doet niets
3. De variabele `${nieuwestatus}` wordt in de actiereeks opgeroepen.

Met deze oplossing kun je dus een waarde opzoeken in een lijst met resultaten, en die waarde vervolgens toewijzen aan een variabele `${nieuwestatus}`.

### Let op unieke zoektermen

In het voorbeeld hierboven zochten we op de waarde "Gesloten", en gebruikten we vervolgens de UNID ervan. Dit werkt natuurlijk alleen goed wanneer er maar één status met naam "Gesloten" is. Ga je dus met de syntax `<#list>`, `<#if>` en `<#assign>` een waarde vertalen naar een UNID? Zorg er dan voor dat je zoekterm uniek is.



## Nog een voorbeeld met <#list>, <#if> en <#assign>

Onderstaande JSON-body bevat code die de UNID van een status vertaalt naar de naam:

### JSON-body

```
<#list _responses.getstatuses.body as status><#if status.id == afhandelingstatusid>
<#assign huidigestatus = status.name></#if></#list>

{
  "action": "De huidige status is ${huidigestatus}"
}
```

## De logica stap voor stap

De variabele \${huidigestatus} wordt gezet op de volgende manier:

1. In een eerdere stap (een GET-call op het endpoint /tas/api/incidents/statuses) worden alle mogelijke statuses van een melding opgeroepen: getstatuses.
2. <#list>: De responsedata van de stap getstatuses is een lijst. TOPdesk loopt de resultaten in de lijst één voor één na.
  - <#if>: Is het element id gelijk aan de waarde van afhandelingstatusid?
    - <#assign>: \${huidigestatus} wordt gemaakt met waarde name.
  - <#if>: Is het element id niet gelijk aan de waarde van afhandelingstatusid?
    - TOPdesk doet niets
3. De variabele \${huidigestatus} wordt in de actiereeks opgeroepen:

Request body:

```
{
  "action": "De huidige status is Gesloten"
}
```

NB: de variabele afhandelingstatusid is een gereserveerde variabele vanuit TOPdesk. Die bestaat dus al standaard in actiereeksen binnen de meldingenmodule.

## <#assign> binnen een URL

Het laatste scenario waar je een <#assign> kunt gebruiken, is binnen de URL. Onderstaande is een GET-call op het endpoint /tas/api/persons, met als doel om een persoonskaart op te halen met emailadres "rogier@laansloot.nl":

The screenshot shows a REST client interface with the following fields:

- Stap 1**
- Naam:** getperson
- HTTP-methode:** GET
- URL:** `${_variables.TOPdeskURL?no_esc}/tas/api/persons?email=<#assign email = "rog`
- Headers:**
  - Authorization:** BASIC `${_variables.TOPdeskLoginZichtbaar}`
- Deze stap uitvoeren:** Alleen als alle eerdere stappen zijn ges

De URL is in de screenshot niet helemaal zichtbaar, maar is als volgt:

### URL

```
${_variables.TOPdeskURL?no_esc}/tas/api/persons?email=<#assign email = "rogier@laansloot.nl">
${email}&page_size=1
```

## Uitgebreide logica in de URL

Het invoerveldje voor URL is maar klein, maar je kunt alle Freemarker-logica erin invoeren die je wil. Je kunt bijvoorbeeld ook de constructie met <#list>, <#if> en <#assign> in een URL opnemen:

### URL

```
${_variables.TOPdeskURL?no_esc}/tas/api/persons?email=<#list _responses.getpersons.body as
person><#if person.id == "9010c3f4-ab7f-4084-a1d0-3c9874fbe17b"><#assign correctemail =
person.email></#if></#list>${correctemail}&page_size=1
```

Hier zoekt TOPdesk het emailadres op van de persoon met UNID "9010c3f4-ab7f-4084-a1d0-3c9874fbe17b". Dat emailadres wordt gezet in de variabele `${correctemail}` en vervolgens opgeroepen.

## Hoofdstuk 8

# Tekstmanipulatie

In hoofdstuk 7 "Variabelen" is uitgelegd hoe je een variabele kunt aanmaken, en hoe je een bepaalde waarde toekent aan een variabele.

Dit hoofdstuk gaat één stap hierop verder. Je kunt namelijk met Freemarker ook extra bewerkingen laten toepassen op je variabele. Je kunt dus de waarde van een variabele aanpassen bij het aanroepen ervan.

Deze extra bewerkingen bewerkstellig je met de syntax `${variabele?extrabewerking}`. Dit hoofdstuk geeft een overzicht van de meest handige extra bewerkingen voor actiereeksen binnen TOPdesk.

## ?replace

De meest basale extra bewerking is de ?replace-functie. De beginsituatie is één variabele die we simpelweg aanroepen:

### JSON-body

```
<#assign omschr = "Kabelbreuk in Amsterdam">

{
  "action": "Korte omschrijving is ${omschr}"
}
```

Resultaat:



**AUTOMATION,**

Korte omschrijving is Kabelbreuk in Amsterdam

Met de ?replace-functie kunnen we waarde X laten vervangen door waarde Y:

### JSON-body

```
<#assign omschr = "Kabelbreuk in Amsterdam">

{
  "action": "Korte omschrijving is ${omschr?replace("Amsterdam", "Rotterdam")}"
}
```

Resultaat:



**AUTOMATION,**

Korte omschrijving is Kabelbreuk in Rotterdam

De functies voor extra bewerkingen plaats je dus binnen de accolades {} van de variabele:

```
${omschr?replace("Amsterdam", "Rotterdam")}
```

## Derde argument

De optie `?replace` vervangt standaard *alle* gevallen waar TOPdesk de oude waarde vindt in de string:

### JSON-body

```
<#assign omschr = "Kabelbreuk in Amsterdam">

{
  "action": "Korte omschrijving is ${omschr?replace("a", "_")}"
}
```

Resultaat:



**AUTOMATION,**

Korte omschrijving is K\_belbreuk in Amsterd\_m

Wil je alleen in het eerste geval de oude waarde vervangen door de nieuwe waarde? Dan geef je als derde argument "f" op in de `?replace`-functie:

### JSON-body

```
<#assign omschr = "Kabelbreuk in Amsterdam">

{
  "action": "Korte omschrijving is ${omschr?replace("a", "_", "f")}"
}
```

"f" betekent First only. Resultaat:



**AUTOMATION,**

Korte omschrijving is K\_belbreuk in Amsterdam

Een ander derde argument binnen de `?replace`-functie is "i", waarmee je geen onderscheid maakt tussen hoofdletters en kleine letters. De syntax `${omschr?replace("a", "_", "i")}` wordt dan "K\_belbreuk in \_msterd\_m".

In een overzichtje:

Syntax	Resultaat
<code>\${omschr?replace("a", "_")}</code>	K_belbreuk in Amsterd_m
<code>\${omschr?replace("a", "_", "f")}</code>	K_belbreuk in Amsterdam
<code>\${omschr?replace("a", "_", "i")}</code>	K_belbreuk in _msterd_m

## Zoekwaarde verwijderen

Naast vervangen kun je met de `?replace`-functie ook tekst verwijderen, namelijk door de zoekwaarde te vervangen met een lege waarde. Je kunt op die manier bijvoorbeeld de streepjes in een UNID verwijderen:

### JSON-body

```
<#assign omschr = unid>
```

```
{  
  "action": "https://locatietopdesk.net.com/tas/secure/incident?unid=${omschr?replace("-", "")}"  
}
```

### Resultaat:



**AUTOMATION,**

[https://locatietopdesk.net.com/tas/secure/incident?  
unid=db0057e430964cb1841a23f1abd7c922](https://locatietopdesk.net.com/tas/secure/incident?unid=db0057e430964cb1841a23f1abd7c922)

Overigens is `${unid}` een gereserveerde variabele in TOPdesk, en daarom kan `${omschr}` weggelaten worden:

### JSON-body

```
{  
  "action": "https://locatietopdesk.net.com/tas/secure/incident?unid=${unid?replace("-", "")}"  
}
```

## ?keep\_before, ?keep\_after, ?keep\_before\_last, ?keep\_after\_last

De functies ?keep\_before, ?keep\_before\_last, ?keep\_after, ?keep\_after\_last gebruik je wanneer je een substring uit de variabele wil halen. In onderstaand voorbeeld scheiden we de substring op basis van een komma (dat we als karakter opgeven in de functie):

### JSON-body

```
<#assign info = "Naam: Jan, Achternaam: Visboer, Leeftijd: 30, Beroep: Koopman ">
{
  "action": "Eerste deel is ${info?keep_before(",")}<br>Laatste deel is ${info?keep_after(",")}"
}
```

### Resultaat:



#### AUTOMATION,

Eerste deel is Naam: Jan

Laatste deel is Achternaam: Visboer, Leeftijd: 30, Beroep: Koopman

De functies ?keep\_before en ?keep\_after scheiden zo dus de variabele \${info} bij de eerste komma die ze tegenkomen. De functies ?keep\_before\_last en ?keep\_after\_last scheiden juist bij de laatste komma:

### JSON-body

```
<#assign info = "Naam: Jan, Achternaam: Visboer, Leeftijd: 30, Beroep: Koopman ">
{
  "action": "Eerste deel is ${info?keep_before_last(",")}<br>Laatste deel is ${info?keep_after_last(",")}"
}
```

### Resultaat:



#### AUTOMATION,

Eerste deel is Naam: Jan, Achternaam: Visboer, Leeftijd: 30

Laatste deel is Beroep: Koopman

De komma is hier slechts een voorbeeld. Je kunt scheiden op elke waarde die je kiest. Wat zou er gebeuren wanneer we scheiden op "Visboer"?

```
${info?keep_before("Visboer")}
${info?keep_after("Visboer")}
```

## ?remove\_beginning, ?remove\_ending

Wil je specifiek waarden aan het begin of aan het eind van een variabele verwijderen? Dan gebruik je daarvoor de functies ?remove\_beginning of ?remove\_ending.

Binnen de functie geef je de zoekwaardes op die je wil weghalen, zoals "in: " of "!":

### JSON-body

```
<#assign info = "in: 10, out: 20!">
```

```
{  
  "action": "${info?remove_beginning("in: ")}<br>${info?remove_ending("!")}"  
}
```

Resultaat:

---

**A** **AUTOMATION,**  
10, out: 20!  
in: 10, out: 20

Als de zoekwaardes niet voorkomen, dan doen de functies niets:

### JSON-body

```
<#assign info = "in: 10, out: 20!">
```

```
{  
  "action": "${info?remove_beginning("X")}<br>${info?remove_ending("Y")}"  
}
```

Resultaat:

---

**A** **AUTOMATION,**  
in: 10, out: 20!  
in: 10, out: 20!



## ?ensure\_starts\_with, ?ensure\_ends\_with

Met deze functies dwing je af dat de ingevoegde waarde begint of eindigt met een bepaalde string. In onderstaand voorbeeld dwingen we de string "Naam: " af


### JSON-body

```
<#assign info = "Pietje Bel">

{
  "action": "${info?ensure_starts_with("Naam: ")}"
}
```

Resultaat:

---

 **AUTOMATION,**  
Naam: Pietje Bel

Als de opgegeven string al in de variabele zit, dan doet de functie in feite niets:


### JSON-body

```
<#assign info = "Naam: Pietje Bel">

{
  "action": "${info?ensure_starts_with("Naam: ")}"
}
```

Het resultaat is dan precies hetzelfde:

---

 **AUTOMATION,**  
Naam: Pietje Bel

De functie ?ensure\_ends\_with werkt op exact dezelfde manier, alleen dan met het einde van de variabele.

## Slicing


Slicing is een techniek die eerder bij lijsten ter sprake is gekomen. Met vierkante haken kun je aangeven welke letters en karakters je precies uit een groter geheel wil plukken. Wil je alleen de eerste letter van een variabele? Dan gebruik je daarvoor [0]:

### JSON-body

```
<#assign plaats = "Sintjohannesga">

{
  "action": "Substring: ${plaats[0]}"
}
```

Resultaat:

 **AUTOMATION,**  
Substring: S


Wil je de eerste tien letters? Dan gebruik je [0..10]:

### JSON-body

```
<#assign plaats = "Sintjohannesga">

{
  "action": "Substring: ${plaats[0..10]}"
}
```

Resultaat:

 **AUTOMATION,**  
Substring: Sintjohanne

Een cheatsheet:

Variabele	Resultaat (vanuit waarde "Sintjohannesga")
\${plaats[0..1]}	Si
\${plaats[1..3]}	int
\${plaats[2..<5]}	ntj
\${plaats[2..5]}	ntjo
\${plaats[10..*3]}	esg
\${plaats[2..]}	ntjohannesga

## De plus + operator

Een vervolg op slicing is de + operator, waarmee je waardes aan elkaar plakt:

### JSON-body

```
<#assign plaats = "Sintjohannesga">

{
  "action": "Substring: ${plaats[0..2]} + plaats[4..5]}"
}
```

Resultaat:



**AUTOMATION,**

Substring: Sinjo

De samengevoegde waardes zijn niet noodzakelijkerwijs altijd variabelen. Je kunt ook platte tekst erbij plaatsen, mits de tussen dubbele aanhalingstekens staan:

### JSON-body

```
<#assign plaats = "Sintjohannesga">

{
  "action": "Substring: ${plaats[0..2]} + "ABCD" + plaats[4..5]}"
}
```

Resultaat:



**AUTOMATION,**

Substring: SinABCDjo

Hetzelfde kun je overigens ook bereiken door de variabele `${plaats}` tweemaal aan te roepen:

### JSON-body

```
<#assign plaats = "Sintjohannesga">

{
  "action": "Substring: ${plaats[0..2]} ABCD ${plaats[4..5]}"
}
```

De + operator heeft daarna ook een mathematische functie; je kunt er daadwerkelijk getallen mee optellen. Dat is echter buiten de scope van dit document.

## Funcities combineren

Je kunt meerdere functies toepassen op één variabele. Je plaatst ze simpelweg achter elkaar, maar nog wel binnen de accolades {}.

### JSON-body

```
<#assign info = "Pietje Bel">

{
  "action": "${info?ensure_starts_with("Naam: ")?ensure_ends_with(" (gebruiker)}"
}
```

Resultaat:



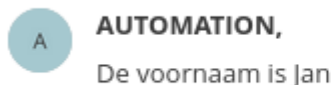
Een ander voorbeeld:

### JSON-body

```
<#assign info = "Naam: Jan, Achternaam: Visboer, Leeftijd: 30, Beroep: Koopman ">

{
  "action": "De voornaam is ${info?keep_after(": ")?keep_before(",")}"
}
```

Resultaat:



Ook slicing kun je met andere functies combineren. Wat zou het resultaat van onderstaande JSON-body zijn?

### JSON-body

```
<#assign info = "Naam: Jan, Achternaam: Visboer, Leeftijd: 30, Beroep: Koopman ">

{
  "action": "De voornaam is ${info[8..20]?keep_after("a", "i")?keep_before_last("a", "i")}"
}
```

Tip: check het kopje "Derde argument" onder het hoofdstukje ?replace.

## Uit de praktijk: data uit een ingediend formulier opzoeken

Onderstaande screenshot komt uit een formulier dat geautomatiseerd behandelaarsaccounts aanmaakt op basis van een emailadres:

### Benodigde informatie

Emailadres *	<input type="text" value="jantje@internet.nl"/>
Behandelaarsgroep *	<input type="text" value="Servicedesk"/>

Met een GET-call kun je het verzoekveld uitlezen, bijvoorbeeld:

#### JSON-responsedata van stap getchange

```
Response body: {"results":[{"id":"fef66b7a-b9c3-404d-a2d9-fa6499eab18a","memoText":"Emailadres<br/>- jantje@internet.nl<br/><br/>Behandelaarsgroep<br/>- Servicedesk","entryDate":"2019-07-14T09:06:47+0000","operator":null,"person":null}]}
```

Het veld memoText bevat de ingevulde gegevens, en dat veld is hierboven gehighlight. De memoText haal je als volgt uit de responsebody:

```
<#assign verzoek = _responses.getchange.results[0].memoText>
```

Als we goed kijken, dan hebben we de data nodig na "Emailadres<br/>- ", en voor "<br/><br/>Behandelaarsgroep<br/>":

```
verzoek?keep_after("Emailadres<br/>- ")
```

```
"Emailadres<br/>- jantje@internet.nl<br/><br/>Behandelaarsgroep<br/>- Servicedesk"
```

```
verzoek?keep_before("<br/><br/>Behandelaarsgroep<br/>")
```

```
"Emailadres<br/>- jantje@internet.nl<br/><br/>Behandelaarsgroep<br/>- Servicedesk"
```

De twee functies ?keep\_before en ?keep\_after kun je vervolgens combineren. In twee stappen heb je dan het ingevulde emailadres:

```
<#assign verzoek = _responses.getchange.results[0].memoText>
<#assign mailadres =
verzoek?keep_after("Emailadres<br/>- ")?keep_before("<br/><br/>Behandelaarsgroep<br/>")>
```

Hiermee ontsluit je dus allerlei functionaliteit: je kunt nu geautomatiseerd acties uitvoeren op basis van data uit ingevulde formulieren. De mogelijkheden zijn eindeloos!

## Regex (geavanceerd)

Tenslotte nog een korte opmerking over regex. Regex maakt het mogelijk om strings heel minutieus te vergelijken met patronen. Heb je geen ervaring met regex? Dan kun je dit kopje overslaan

Regex-patronen zijn op het eerste gezicht erg complex. Onderstaand is het regex-patroon voor alleen letters:

```
^([a-zA-Z]+)$
```

Met het regex-patroon beoordeelt een computer of een waarde geldig is (TRUE) of ongeldig is (FALSE):

Pietje	TRUE
Pietje Bell	FALSE (er zit een spatie in)
ABCD123	FALSE (er zitten cijfers in)
ABCdef	TRUE

### Binnen actiereeksen

Je kunt regex gebruiken in de "Aangepaste voorwaarde", om te controleren op geldige waarden. Dat doe je met de `?matches`-functie:

#### Aangepaste voorwaarde

```
<#assign info = "ABCdef">  
<#if info?matches("^([a-zA-Z]+)$")>true</#if>
```

```
<#assign info = "ABCdef">  
${info?matches("^([a-zA-Z]+)$")}
```

Daarnaast kun je ook regex gebruiken met de `?replace`, `?keep_before`, `?keep_before_last`, `?keep_after`, `?keep_after_last` functies. Dat doe je door het extra argument "r" mee te geven. Voorbeeld:

#### JSON-body

```
<#assign info = "xxxxxxxAABB">  
  
{  
  "action": "Van ${info} naar ${info?replace("^([x]+)", "y", "r")}"  
}
```

Resultaat:



**AUTOMATION,**

Van xxxxxxxAABB naar yAABB

## Hoofdstuk 9

# Datumvelden

Werken met datumvelden en actiereeksen is erg ingewikkeld. Dat komt omdat computers hele precieze eisen stellen aan een datum. De tijdzone, de taal- en landinstellingen, en de verschillende manieren van weergeven van een tijdstip maken dit nog complexer.

Moet je met datumvelden werken? Dan moet je drie punten in orde hebben:

### **De tijdzone**

Zorg ervoor dat de tijdzone van TOPdesk goed is ingesteld.

### **Datumobjecten**

Begrijp het verschil tussen een datumobject en een datumstring, en gebruik ze op de juiste manier.

### **Syntax**

Gebruik de juiste syntax in je actiereeksen. Dit luistert wederom heel nauw.

Dit hoofdstuk loodst je door deze onderwerpen heen.

## Platte datumvelden en tijdzones

Hoe TOPdesk omgaat met datumvelden kan het beste geïllustreerd worden aan de hand van een melding:



De datum/tijd van aanmaak staat rechtsboven: "28 februari 2019 om 20:19". In de TOPdesk-database staat dat moment als volgt geregistreerd:

2020-02-28 20:19:19

Dit is een zogenoemde "platte" timestamp. Je kunt er niet aan aflezen in welke tijdzone de gebruiker was toen hij/zij de melding aanmaakte.

Alle data staan in de TOPdesk-database als "platte" datumvelden, zonder tijdzone-informatie. Ook in HTTP-requests gebruik je platte timestamps:

&action=edit&field0=vrijedatum1&value0=2019-05-14 16:37:50

&action=edit&field0=vrijedatum1&value0=2019-05-14 16:37

&action=edit&field0=vrijedatum1&value0=2019-05-14

Het voordeel van platte timestamps is dat het gebruik intuïtief is. Het grote nadeel van platte timestamps is natuurlijk dat je in de knoop raakt wanneer er gebruikers in meerdere tijdzones actief zijn in dezelfde TOPdesk.

### Per vestiging de tijdzone opgeven

TOPdesk kan gelukkig prima omgaan met gebruikers in meerdere vestigingen. In de Functionele instellingen, onder "Algemeen" heb je de optie "Tijdzones gebruiken". Als je die optie aanvinkt kun je vervolgens per vestiging de juiste tijdzone instellen, onder tabblad "Informatie":

#### Tijdzone van vestiging

Tijdzone

Deze setting heeft impact op alle personen en behandelaars die onder die vestiging hangen. Verander je de tijdzone van de vestiging? Dan zien *zij* (na opnieuw inloggen) dat alle timestamps in de interface van TOPdesk zijn gewijzigd.



Voorbeeld: als je de tijdzone van een bepaalde vestiging verandert in "Asia/Singapore", dan verandert de datum/tijd van aanmaak in de interface van "28 februari 2019 om 20:19" naar "29 februari 2020 om 03:19":



Dit is alleen zichtbaar voor behandelaars en aanmelders die vallen onder de vestiging die nu tijdzone "Asia/Singapore" heeft gekregen.

Deze verandering in tijdstippen is alleen visueel; in de database is niets veranderd. Daar staat de hele tijd dezelfde "platte" timestamp:

2020-02-28 20:19:19

TOPdesk rekent de datum uit de database dus om naar de doeltijdzone, en laat die tijd vervolgens zien in de interface.

### Tijdzone bekend

Maar als alleen de timestamp "2020-02-28 20:19:19" in de database staat, hoe weet TOPdesk dan of dat Nederlandse tijd is, of Chinese tijd, of Amerikaanse tijd? Dat leest TOPdesk uit van de server waarop het is geïnstalleerd. Op basis daarvan kan TOPdesk vervolgens tijdsverschillen uitrekenen.

Wil je zeker zijn dat TOPdesk de juiste tijdzone gebruikt? Of wil je een andere tijdzone gebruiken dan die van de server? Dan kun je ook een standaard tijdzone afdwingen in TOPdesk. Bij een on-premise TOPdesk doe je dat als volgt:

- Stop TOPdesk
- Open het bestand "topdesk.conf" in de installatiemap van TOPdesk
- Zet de gehyponde lijn onder het kopje # Java Additional Parameters, en zorg ervoor dat de nummertjes elkaar netjes opvolgen:

#### topdesk.conf

```
# Java Additional Parameters
wrapper.java.additional.1=-Xlog:gc*=debug:./log/gclog.txt
wrapper.java.additional.2=-Djava.locale.providers=JRE,SPI
wrapper.java.additional.3=-Djdk.tls.client.protocols=TLSv1.2
wrapper.java.additional.4=-Duser.language=nl
wrapper.java.additional.5=-Dcvs.export.enabled=true
wrapper.java.additional.6=-Duser.timezone=Europe/Amsterdam
```

- Sla het bestand op en start TOPdesk weer

Heb je TOPdesk draaien op SAAS, in de cloud? Dan is dit al ingeregeld.

## UTC-tijd via de TOPdesk API

Met de TOPdesk API worden datumvelden weer een stapje complexer. Als je een GET-call uitvoert op de genoemde melding, dan krijg je de volgende timestamp terug als datum/tijd van aanmaak:

```
"callDate" : "2020-02-28T19:19:00.000+0000"
```

Terwijl in de database dit staat:

```
2020-02-28 20:19:19
```

Een uur verschil dus!

De reden hiervan is dat de TOPdesk-API de tijd omrekent naar UTC-format. Dat is de standaard voor tijdzone-onafhankelijke tijdsweergave. UTC komt overeen met de tijd in Groot-Brittannië. De TOPdesk API geeft altijd de UTC-tijd terug, ook wanneer er vestigingen zijn met verschillende tijdzones.

Daarnaast bevat de datumweergave uit de TOPdesk API juist wel tijdzone-informatie:

```
"callDate" : "2020-02-28T19:19:00.000+0000"
```

De +0000 betekent: geen afwijking met UTC-tijd. +0200 betekent een afwijking van twee uur (later) ten opzichte van UTC-tijd.

Ga je werken met datumvelden en actiereeksen? Dan moet je hier dus alert op zijn. Er zullen dus misschien gevallen voorkomen waarbij je timestamps tegenkomt die niet overeenkomen met de tijd in je tijdzone.

### Datumveld wegschrijven naar een ander datumveld

Overigens kun je wel gewoon een ontvangen datumveld wegschrijven naar een ander datumveld. Onderstaande JSON-body kopieert het vrije datumveld 1 naar het vrije datumveld 2:

#### JSON-body

```
{
  "optionalFields1": { "date2" : "${_responses.getincident.body.optionalFields1.date1}" }
}
```

Resultaat:

Datum 1ste gesprek

28 februari 2020

17:42



Datum 2e gesprek

28 februari 2020

17:42



TOPdesk interpreteert en verwerkt hier dus wel de ontvangen timestamps goed, zonder uur verschil. Dat komt omdat TOPdesk hier datumobjecten herkent.

## Over datumobjecten en strings

### Datumobject

Een datumobject moet je zien als een datumvakje, inclusief een ingevulde datum/tijdstip:

Datum 1ste gesprek

28 februari 2020

17:42



Het datumvakje dwingt af dat je alleen maar een datum/tijdstip kunt invullen. Je kunt er dus niet tekst of nummers in kwijt. Computers rekenen altijd met datumobjecten.

Alle datumvelden uit TOPdesk zijn standaard al datumobjecten. Daarnaast heb je de speciale variabele `${.now}`, een datumobject dat het huidige tijdstip aangeeft.

### Datumstring of string

Een datumstring is niets anders dan een weergave van een bepaalde datum in tekst (letters en cijfers). Onderstaand is een uitgetype datum, in het actieveld:



AUTOMATION,

28 februari 2020, 17:42

Mensen begrijpen dit natuurlijk wel als een goede datum, maar computers in principe niet. Simpelweg geldt: computers werken met datumobjecten, en mensen met datumstrings.

### Strings vanuit externe APIs

Vanuit de TOPdesk API krijg je ook datumvelden terug, zoals de datum/tijd van aanmaak van een melding:

```
"callDate" : "2020-02-28T19:19:00.000+0000"
```

**Maar let op! Dit is ook geen datumobject!** Wij weten dat dit een timestamp in UTC-tijd is, maar TOPdesk weet dat niet. TOPdesk ziet hier een string, een rij letters en cijfers in het element `callDate`. Schrijf je dit weg naar een ander veld dan een datumveld? Dan krijg je de hele string 1-op-1 terug:



AUTOMATION,

2020-02-28T19:19:00.000+0000

**Wil je dus "2020-02-28T19:19:00.000+0000" weergeven als "28 februari 2020, 20:19"? Of wil je andere bewerkingen of berekeningen met die datum uitvoeren? Dan moet je de ontvangen timestamp eerst omzetten naar een datumobject.**

Dit is van toepassing op alle timestamps die je vanuit externe APIs terugkrijgt.

## String omzetten naar datumobject

Zoals aangegeven, TOPdesk herkent niet zomaar een timestamp als een datumobject. Dat moet je TOPdesk expliciet vertellen. Dat doe je met de functie `?datetime`.

In onderstaande voorbeeld zit de timestamp in het element `_responses.getincident.body.optionalFields1.date1`. Die waarde wordt in twee stappen omgezet naar een variabele `${datumobject}` (dat dus een datumobject is):

```
<#assign string = _responses.getincident.body.optionalFields1.date1?replace("+0000", "+00:00")>
<#assign datumobject = string?datetime.iso>
```

**Stap 1:** De `?replace`-functie repareert de syntax van de datumstring. De output van de TOPdesk API voldoet namelijk niet aan de ISO-norm voor tijdsweergave.

**Stap 2:** De `?datetime.iso`-functie zet de gecorrigeerde timestamp in ISO-weergave om in een datumobject.

Je kunt de twee stappen ook combineren op één regel:

```
<#assign datumobject = _responses.getincident.body.optionalFields1.date1?replace("+0000", "+00:00")?datetime.iso>
```

Daarnaast is er een alternatieve syntax, waarbij je precies de opmaak van de timestamp opgeeft:

```
<#assign datumobject = _responses.getincident.body.optionalFields1.date1?datetime("yyyy-MM-dd'T'HH:mm:ss.SSSZ")>
```

De betekenis van de letters yyyy en MM etc staat op de volgende pagina.

Het resultaat in alle drie de gevallen is hetzelfde:

---

 **AUTOMATION,**  
2020-02-28T16:42:00Z

In TOPdesk zie je nauwelijks verschil (er staat nu een Z in plaats van +0000), maar onderwater heb je nu een datumobject `${datumobject}`, waarop je allerlei functionaliteit kunt loslaten.

## Datumobject omzetten naar string

Heb je eenmaal een datumobject? Dan kun je met de `?string`-functie precies bepalen hoe je de timestamp wilt weergeven. Het volgende is een vrije datumveld:

Datum 1ste gesprek

6 maart 2020

17:42



Dat veld roepen we aan en schrijven we weg naar het actieveld met de `?string`-functie:

### JSON-body

```
<#assign string = _responses.getincident.body.optionalFields1.date1?replace("+0000", "+00:00")>
<#assign datumobject = string?datetime.iso>

{
  "action": "${datumobject?string["dd MMMM yyyy, HH:mm"]}"
}
```

Resultaat:



AUTOMATION,

06 March 2020, 17:42

Het onderstaand overzicht laat zien hoe je de `?string`-functie gebruikt:

Syntax	Resultaat
<code>\${datumobject?string["dddd MMMM yyyy, HH:mm"]}</code>	06 March 2020, 17:42
<code>\${datumobject?string["MMMM dd, yyyy, HH:mm 'z'"]}</code>	March 06, 2020, 17:42 (CET)

De gehighlighte letters hebben allemaal een betekenis. Dit zijn de meest gebruikte:

Syntax	Voorbeeld	Betekenis
yyyy, yy	2020, 20	Jaar
MMMM, MM	March, 03	Maand
dd	06	Dag
EEEE, EE	Friday, Fri	Dag uitgeschreven
HH	17	Uur (24h)
hh	05	Uur (AM/PM)
mm	42	Minuut
z, zzzz	CET, Central European Time	Tijdzone
Z	+0100	Afwijking van UTC
u	Genummerde dag in de week (1 = maandag, ..., 7 = zondag)	

De `?string`-functie past ook tijdzone-correcties toe. De UTC-tijd die de TOPdesk API teruggeeft is dus automatisch aangepast aan de lokale tijd.

## Taal instellen

De standaard taalinstelling voor de namen van dagen en maanden is Engels. Wil je dat veranderen in Nederlands? Verander dan de locale-setting in "nl":

### JSON-body

```
<#setting locale = "nl">
<#assign string = _responses.getincident.body.optionalFields1.date1?replace("+0000", "+00:00")>
<#assign datumobject = string?datetime.iso>

{
  "action": "${datumobject?string["EEEE dd MMMM"]}"
}
```

### Resultaat:



Deze aanpassing verandert niet alleen de taal, maar ook bijvoorbeeld de weergaves van tijden (24u ipv AM/PM).

## Standaard weergave-opties

Je kunt met de ?string-functie ook een aantal standaardweergaves kiezen:

Syntax	Resultaat	Locale
\${datumobject?string.short}	3/6/20 5:42 PM	en
\${datumobject?string.medium}	Mar 6, 2020 5:42:00 PM	en
\${datumobject?string.long}	March 6, 2020 5:42:00 PM CET	en
\${datumobject?string.full}	Friday, March 6, 2020 5:42:00 PM CET	en
\${datumobject?string.short}	6-3-20 17:42	nl
\${datumobject?string.medium}	6-mrt-2020 17:42:00	nl
\${datumobject?string.long}	6 maart 2020 17:42:00 CET	nl
\${datumobject?string.full}	vrijdag 6 maart 2020 17:42:00 uur CET	nl
\${datumobject?string.iso}	2020-03-06T17:42:00+01:00	en + nl
\${datumobject?string.iso_nz}	2020-03-06T17:42:00	en + nl
\${datumobject?string.iso_m}	2020-03-06T17:42+01:00	en + nl
\${datumobject?string.iso_m_nz}	2020-03-06T17:42	en + nl

## Later dan / eerder dan

Heb je twee datumobjecten? Dan kun je ze met Freemarker vergelijken en vervolgacties uitzetten. Die functionaliteit was nooit eerder mogelijk in TOPdesk. Zie onderstaande JSON-body, met een If-then-constructie:


### JSON-body

```
<#assign string = _responses.getincident.body.optionalFields1.date1?replace("+0000", "+00:00")>
<#assign datumobject = string?datetime.iso>


<#if datumobject gte dataaanmk>
<#assign bericht = "Datumobject is later dan de aanmelddatum">
<#else>
<#assign bericht = "Datumobject is eerder dan de aanmelddatum">
</#if>

{
  "action": "${bericht}"
}
```

Resultaat:

 **AUTOMATION,**  
Datumobject is later dan de aanmelddatum

Nu zetten we het vrije datumveld eerder dan de datum/tijd van aanmaak:

 **AUTOMATION,**  
Datumobject is eerder dan de aanmelddatum

De syntax die deze functionaliteit mogelijk maakt is gte in de If-expressie (greater than or equals, groter of gelijk aan). Een overzichtje:

Syntax	Betekenis
datum1 gte datum2	datum1 is later of gelijk aan datum2
datum1 gt datum2	datum1 is later dan datum2
datum1 lte datum2	datum1 is eerder of gelijk aan datum2
datum1 lt datum2	datum1 is eerder dan datum2
datum1 == datum2	datum1 is gelijk aan datum2
datum1 != datum2	datum1 is ongelijk aan datum2

## Dezelfde dag

De syntax `datum1 == datum2` zul je in de praktijk niet snel tegenkomen, omdat een datumobject ook uren, minuten en seconden bevat. "Gelijk aan" betekent dus in feite op de seconde nauwkeurig hetzelfde tijdstip.

Wil je controleren of twee datumobjecten dezelfde dag zijn? Zet dan de datumobjecten weer om in een string, en vergelijk die strings:

### JSON-body

```
<#assign string = _responses.getincident.body.optionalFields1.date1?replace("+0000", "+00:00")>
<#assign datumobject = string?datetime.iso>

<#if datumobject?string["yyyy-MM-dd"] == dataanmk?string["yyyy-MM-dd"]>
<#assign bericht = "Het is dezelfde dag!">
<#else>
<#assign bericht = "Het is niet dezelfde dag!">
</#if>

{
  "action": "${bericht}"
}
```

Resultaat:



**AUTOMATION,**

Het is dezelfde dag!

## Huidige datum/tijd ophalen met `.now`

Het huidige tijdstip kun je ophalen en wegschrijven met de speciale variabele `.now`:

### JSON-body

```
{
  "optionalFields1": { "date2" : "${.now}" }
}
```

Resultaat:

Datum 2e gesprek

6 maart 2020

11:56



`${.now}` is ook een datumobject. Je kunt daarom ook alle `?string`-functies toepassen.