

# Actiereeksen: Freemarker en variabelen

Auteur: Rogier Visser (Laansloot IT)

Datum: 25 februari 2020

Naam: Laansloot-Actiereeksen-Freemarker-variabelen.pdf

Versie: 0.1

Doelgroep: TOPdesk-applicatiebeheerders

## Introductie

Actiereeksen binnen TOPdesk worden nog krachtiger wanneer je gebruik maakt van Freemarker. Met Freemarker kun je een extra laag logica inbouwen, waardoor je eindeloze mogelijkheden krijgt bij het maken van actiereeksen. Enige programmeerervaring is overigens wel gewenst.

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.

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?

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	QVVUT01BVEIPTjpob3FwYy10ejV6ci16NWpvdY12dTV5Zi1vYnUzaw==

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

## Inhoud

- p.3: Een variabele toewijzen met `<#assign>`
- p.4: Meerdere `<#assign>`
- p.5: Gereserveerde variabelen vanuit TOPdesk
- p.6: Variabele zoeken met een If-then-constructie
- p.7: Variabele zoeken met een lijst
- p.9: `<#assign>` binnen een URL

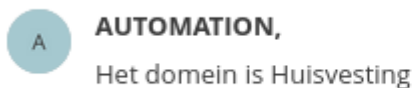
## Een variabele toewijzen met <#assign>

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.