

Actiereeksen: Werken met stappen

Auteur: Rogier Visser (Laansloot IT)

Datum: 7 maart 2020

Naam: Laansloot-Actiereeksen-werken-met-stappen.pdf

Versie: 0.2

Doelgroep: TOPdesk-applicatiebeheerders

Introductie

De kracht van actiereeksen ligt erin dat je kunt werken met opeenvolgende stappen. Met die stappen kun je informatie verzamelen en gebruiken in vervolgstappen. Zo kun je hele complexe functionaliteit maken die niet standaard in TOPdesk zit.

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?

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

Inhoud

- p.3: De actiereeks in TOPdesk (screenshot)
- p.4: Koppeling tussen stap 1 en stap 2
- p.5: JSON-responsedata
- p.6: JSON-responsedata verbeterd weergegeven
- p.7: UNID bij direct velden invoegen
- p.8: Voorbeeld met stappen uit de praktijk
- p.9: Lijsten: één resultaat gebruiken
- p.10: Lijsten: meerdere resultaten gebruiken
- p.11: Lijsten: een waarde opzoeken in een lijst

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.getincident.body.category.name}</code>	Applicaties

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 responsdata 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

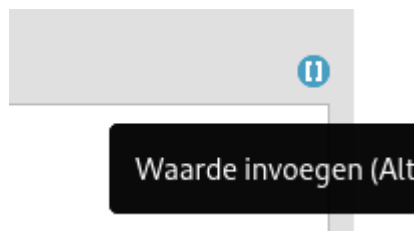
```
Response body: {
  "id": "798e0389-28e1-5b8a-9333-9df8a3e610da",
  "status": "secondLine",
  "number": "M 1707 002",
  "request": "MARCELO: Wil graag werken met de nieuwste versie van TOPdesk",
  "requests": "/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/requests",
  "action": "/tas/api/incidents/id/798e0389-28e1-5b8a-9333-9df8a3e610da/actions",
  "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
<code>\$_responses.getincident.body.category.name</code>	Applicaties

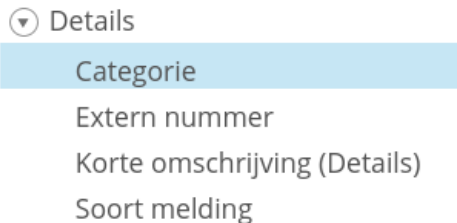
UNID bij direct velden invoegen

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

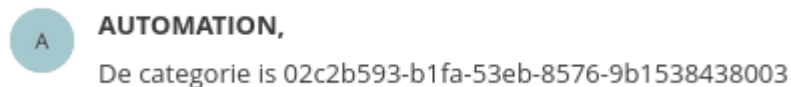
`${incident_domeinid}`

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 *

Behandelaarsgroep *

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 statussen 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 statussen 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 document niet verder behandeld. Daarvoor is het document "Laansloot-Actiereeks-Freemarker-if-then.pdf".