Nieuwe Loket Refresh Token Handleiding

Vanaf Q1 2025 gaat de Loket autorisatie veranderen, dus refresh tokens zullen niet meer permanent zijn, en maar één keer geldig, dus met elke nieuwe access token gaan wij voor de volgende token request een nieuwe refresh token krijgen

Op Triggre applicaties is het aanpaken niet zo ingewikkeld, met een Automation Flow dat de refresh token ten minste één keer per dag ophaalt, zorgen we dat de refresh token altijd geldig is, tegen een minimale kost qua actions.

Dovetail integraties presenteren een grotere uitdaging, aangezien flow variables (settings) kan niet aangepast worden vanbinnen en flow. In plaats van settings, het is nodig om een Tenant Variable te gebruiken.

Let op! Zorg dat jij het proces om een Loket access en refresh token voor de eerste keer via browser en Postman op te halen al kent.

Stap 1: Analyseren

Eerst moeten wij kijken naar de frequentie naar de API calls dat we naar Loket gaan sturen, als iets moet meer dan één keer per dag gebeuren, kunnen we in principe alles binnen dezelfde flow bouwen. Voor dit voorbeeld gaan wij employer en salarisdata één keer per maand ophalen. Voor gevallen zoals dit, moeten wij zoveel mogelijk een filter opzetten zodat alles binnen één flow kan doen. De integratie in twee flows opsplitsen, een "opdrachtsflow", en één refresh token flow die de refresh token vers houdt, is niet de meest gewenste oplossing.

Stap 2: Tenant Variable aanmaken

Voordat jij een token kan ophalen, moet jij eerst een Tenant Variable ervoor hebben, dus ga naar Tenant Variables en maak er een nieuwe met de + knop aan. Tenant Variables zijn beschikbaar voor alle flows binnen een Tenant, dus gebruik nooit een generieke naam zoals "loket token". Geef de token variable een unieke naam die verbonden aan de opdracht is, volgens het formaat *Loket_Refresh_Token_KLANT*.

Tenant Variables						
Test Productio	on					
	_	V I				
Name	_	value	=	Create =	Create =	Las ↓ ≡
oauth2_774bfc70-a1	 144-11ed	0	=	Create =	Create = System	Las ↓ ≡

Stap 3: Nieuwe flow aanmaken

Nadat er een variable bestaat, kan jij een nieuwe flow binnen de tenant aanmaken. Deze flow moet ook een naam die idereen met de de klant makkelijk kan verbinden, vb *Loket_Koppeling_KLANT*.

Stap 4: Copy-Paste

Er is al een werkende flow die de stappen voor het ophalen van de nieuwe refresh token doorloopt. Ga naar de Dovetail demo omgeving en zoek de volgende flow template

S Name ≡ Vers ≡ Trac ≡ Upt	
	ime
LoketRefreshTokenNoCompo 8 1 Week(s) 2 ho	ours 53
 NewLoketTokens 1 Off 1 da 	ıy 3 ho
 TestAPICallTenantVariable 6 1 Week(s) 2 hc 	ours 55

Open de flow en doe een copy-paste van deze 8 nodes naar de flow van stap 3



Stap 5: Tenant Variable Configureren

Nadat jij de nodes op de nieuwe flow staan, moet jij de Tenant Variable configureren.

GetTenantVariableComponent 🛛	×
Header name 🕐 🔇 Specify a specific header name to change.	
loket_refresh_token	
Tenant variable () (3)	
Loket_Refresh_Token_TEST]
Current values	- 11
Test	
RGvCq4b2re_ytrIkGPEYnl6FiTrdEELndh1_yrTRN4u3y6ngL4VBbz8axdBZaBQYwkVbRC8i9V	
Production	

Open de GetTenantVariableComponent node en vervang de tekst met de naam van de eigen Tenant variable van de flow. De header name moet als in de foto ook zijn.

SetTenantVariableComponent 😆	×
Tenant variable () (3) Tenant variable you want to set.	
Expression type Specify the type of expression you are using.	l
Simple ~	J.
Value 💿 🔇 The value you want to set on the variable.	l
\${header.new_refresh_token}	
Encrypt? Whether or not the tenant variable should be stored encrypted.	
Validate expression	1
Current values	
Test	
RGvCq4b2re_ytrIkGPEYnl6FiTrdEELndh1_yrTRN4u3y6ngL4VBbz8axdBZaBQYwkVbRC8i9	V

Doe hetzelfde met de SetTenantVariableComponent, inclusief de Simple expression voor de naam van de header.

Stap 6 Flow Settings

Open de Flow Settings en voeg deze setting variables toe (met dezelfde namen als in de foto).

loket_api_url	https://api.loket-acc.nl	No value
loket_redirect_uri	https://loket.nl	No value
loket_authentication_server_url	https://oauth.loket-acc.nl	No value
loket_client_id	ThirdPartiesTestClient	No value
loket_client_secret	Welkom01	No value

Stap 6: Scheduler configureren

De laatste stap is een Scheduler component als eerste stap van de flow te plaatsen. Een manier om deze te configureren is met en cronjob. De cronjob in de foto trapt de flow elke dag om 00:00 uur af. (Jij kan een cron expression aanmaken en ophalen via <u>cronmaker.com</u>)

SchedulerComponent 🛛	×
Time Zone Default Europe/Amsterdam is used.	
Europe/Amsterdam	
Method The method to determine when the scheduler should run	
Oron	
O Repeats	
Cron Define a cronjob for the scheduler Every day \lor at $0 \lor$: $0 \lor$ Clear	
000**? Validate	e
	/

Stap 7: De eerste token ophalen

Nu dat de flow klaar voor gebruik is, is een eerste refresh token nodig. Loop de OAuth flow via browser en Postman zoals bij het oude system door, maar sla de refresh token in een settings variable zoals eerder niet op, maar direct in de Tenant Variable van stap 1.

Tenant Variable	2S			
Test Production				
Name =	Value =	Create ≡	Create ≡	Las $\downarrow \equiv$
oauth2_774bfc70-a144-11ed	0	15/05/202	System	15/10/202
Loket_Refresh_Token_TEST	RGvCq4b2re_ytrl	11/10/202	victor@gro	15/10/202

Stap 8 (Optioneel): Een filter opzetten

Als de functionaliteit van de flow één keer per dag moet geëxecuteerd worden, kan jij API calls direct na stap 7 sturen, maar het kan zijn dat de dagelijkse token refresh en de echte funcionaliteit niet gesynchroniseerd zijn.

Volgens het voorbeeld, hebben wij een flow die maar één keer per maand Loket API calls stuurt. Elke laatste dag van de maand haalt de flow salarisdata binnen, maar de flow trapt elke dag af, zodat wij altijd een geldige refresh token hebben. Deze situatie lossen wij met een filter op.

Voor het geval dat een flow op een bepaalde dag van de maand moet vb. elke 25^e van de dag worden de salarissen betaald, de volgende Simple expressie los het op: **\${date:now:dd} == 25**.



8a Script en Filter componenten toevoegen

Plaats een Script en een Filter component tussen de laatste node van Stap 4 (waar de access token binnen de Authorization header gezet wordt) en de HTTP component die de API call gaat sturen. Goed opletten dat op Dovetail is Groovy gewenst in plaats van JavaScript.

8b Script schrijven

Wat belangrijk is, is dat de script gebruikt een header als een variable om een output op te slaan. Dit voorbeeld geeft true of false terug als vaandaag de laatste dag van de maand is, of niet.

ScriptComponent 🖲	
Upload method	
Enter script manually	
○ Upload file	
Script 🔘 🔇	
The script that will be executed for each message.	
import java.time.LocalDate	
import java.time.temporal.TemporalAdjusters	- 1
// Get today's date	- 1
LocalDate today = LocalDate.now()	- 1
// Get the last day of the current month	
LocalDate lastDayOfMonth = today.with(TemporalAdjusters.lastDayOfMonth())	
// Check if today is the last day of the month	
if (today.eguals(lastDayOfMonth)) {	- 1
request.setHeader('isLastDay',true)	
}else {	- 1
request.setHeader('isLastDay',false)	
}	T
Language	
Scripting language used to evaluate the script.	
Groovy	~

8c Filter opzetten

Nadat de script klaar is, moeten wij een filter bouwen, de filter component laat de flow verder gaan alleen als de regel true is. Meer informatie over de Filter component is hier beschikbaar: https://academy.dovetail.world/docs/components/routing/filter

FilterComponent	×
Expression Solution (1) Solution (2) S	
\${header.isLastDay} == 'true'	
Type Type of the expression.	
Simple	~
Validate expression	

Zorg dat jij de output header van de script hier binnen de expression gebruikt. Dit voorbeeld geeft true terug als vaandag de laatste dag van de maand is, en dus kunnen wij de salarisdata binnenhalen.

9 API call(s) sturen

Plaats en HTTP component direct na de filter, of de laaste SetHeaders component van stap 4 als jij geen filter gebruikt, en jij bent klaar om API calls naar Loket te sturen, configureer de request volgens de API reference, installer de flow en begin met testen.

