

Azure Logic Apps (Learning Story)

Ik wil onderzoeken hoe ik mijn eigen afspraken uit outlook kan halen, met als doel dit later te gebruiken in een PA.

Ik begin daarom met te kijken of dit met en logic app kan.

Creating our first simple email response logic app.

My first step was to read through this Microsoft page to get an understanding of how to use logic apps with outlook.

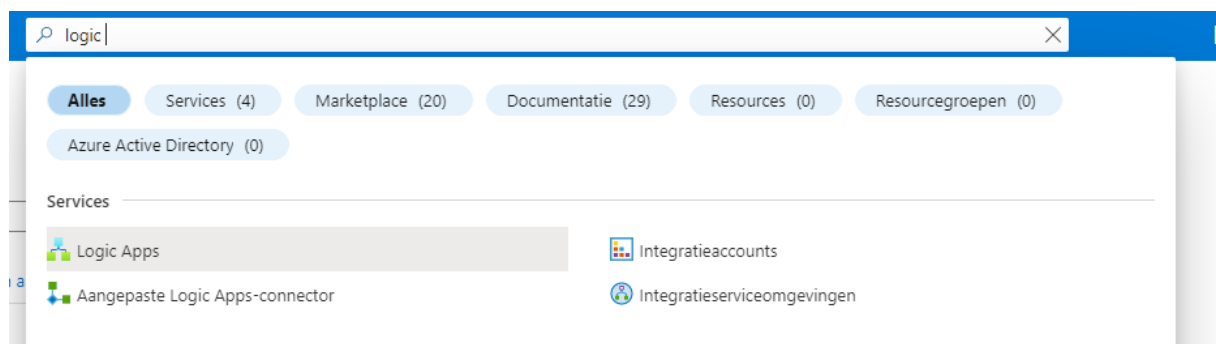
<https://docs.microsoft.com/en-us/azure/connectors/connectors-create-api-office365-outlook>

What is azure logic app.

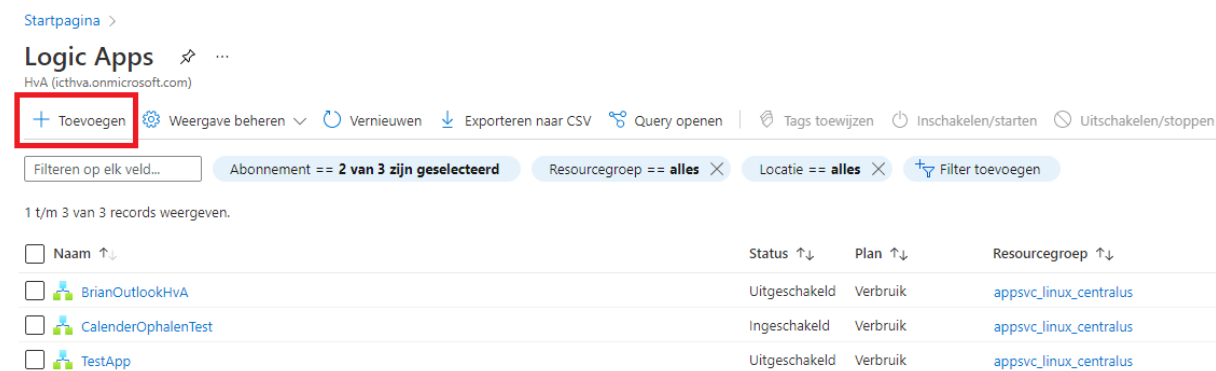
Azure Logic Apps is a cloud service. Using Logic Apps defines the workflow at ease consuming a range of APIs exposed as Connectors. These Logic App connectors will perform the sequence of actions defined in the workflow whenever the trigger gets fired.

Step by step what I did.

First I created a logic app in my azure portal.



When you select the logic app u it will direct you to a new page where u can see your exsisting logic apps and create new ones.



After selecting to create a new logic app a new window opens with options that need to be filled in to create a logic app. In the screenshot below u can see the default create form when I first open it apart from the plan I changed that from standard to consumption.

Create Logic App ...

Create a logic app, which lets you group workflows as a logical unit for easier management, deployment and sharing of resources. Workflows let you connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.


Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="Collab-ICTS-WB"/>
Resource Group *	<input type="text" value="(New) Resource group"/>
	Create new


Instance Details

Logic App name *	<input type="text" value="Logic App name"/>
Region *	<input type="text" value="North Central US"/>
Enable log analytics *	<input type="radio"/> Yes <input checked="" type="radio"/> No

 There are no log analytics workspace resources in the selected subscription. In order to enable log analytics, either create a new log analytics workspace resource or switch to a subscription which already has one.

Plan

The plan type you choose dictates how your app scales, what features are enabled, and how it is priced.

Plan type *	<input checked="" type="radio"/> Consumption: Best for entry-level. Pay only as much as your workflow runs.
	<input type="radio"/> Standard: Best for enterprise-level, serverless applications, with event-based scaling and networking isolation.
	 Looking for the classic consumption create experience? Click here

[Review + create](#)

[< Previous](#)

[Next : Tags >](#)

Explanation of options.

Subscription/Resource group: Here you select what azure subscription you want to use and which resource group that is a part of that subscription.

App name: The name of the App does not matter in way the app will function so I suggest to pick something that is relevant to the functionality of the app.

Region: Region you want to host your logic app.

Enable log analytics: Log analytics allows you to save data

Plan Type:

- **Consumption** : Consumption is a per use based plan so you pay every time the logic app is triggered. The first 4000 triggers are free.
- **Standard** : Standard is a pay per hour the logic app is online based plan.

Standard Plan (Single-tenant)

The Logic Apps Standard plan supports local development on Windows, Linux and Mac, offers a new layout engine that supports complex workflows, and enables custom connector extensions. Leverage the containerized runtime to run Logic Apps locally, in the cloud, or on premises with virtual network capabilities.

	Price
vCPU	€0.177434/hour
Memory	€0.012687/hour

Consumption Plan (Multi-tenant)

A pay-as-you-go pricing model that charges based on triggers and actions that are specified in a logic app. Running in a multi-tenant environment, this plan also leverages connectors, which provide users with access to SaaS applications and on premise data sources. Visit the [documentation](#) for our growing list of Standard and Enterprise connectors. Connectors make it easy to unlock data and applications behind the firewall, securely connect to cloud-based services and take advantage of rich integration leveraging on-premises systems you're already invested in.

Every time a Logic App definition runs the triggers, action and connector executions are metered.

	Price Per Execution
Actions	€0.000023 First 4,000 actions free
Standard Connector	€0.000113
Enterprise Connector	€0.000900

Now that we know what options we have when creating a logic app we can fill out the form. In the next screen shot you can see the filled out options I used when creating my logic apps.

[Home](#) >

Create Logic App ...

Basics Tags Review + create

Create a logic app, which lets you group workflows as a logical unit for easier management, deployment and sharing of resources. Workflows let you connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.


Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<div>Azure for Students</div>
Resource Group *	<div>appsvc_linux_centralus</div> <div>Create new</div>


Instance Details

Logic App name *	<div>BrianTestApp</div>
Region *	<div>West Europe</div>
Enable log analytics *	<div><input type="radio"/> Yes <input checked="" type="radio"/> No</div>


 There are no log analytics workspace resources in the selected subscription. In order to enable log analytics, either create a new log analytics workspace resource or switch to a subscription which already has one.

Plan

The plan type you choose dictates how your app scales, what features are enabled, and how it is priced.

Plan type *	<div><input checked="" type="radio"/> Consumption: Best for entry-level. Pay only as much as your workflow runs.</div> <div><input type="radio"/> Standard: Best for enterprise-level, serverless applications, with event-based scaling and networking isolation.</div> <div> Looking for the classic consumption create experience? Click here</div>
-------------	---

The reason I used the Azure for students subscription this time is because the HBO-ICT subscription gets a error saying it has no permission to create a logic app.

Subscription *	<div>Self-HBO_ICT</div> <div> Resource provider(s): Microsoft.Web, Microsoft.Logic are not registered for subscription Self-HBO_ICT and you don't have permissions to register a resource provider for subscription Self-HBO_ICT</div>
Resource Group *	<div>Mixit1-rg</div> <div>Create new</div>

Now we can review and create our logic app.

Create Logic App ...

Basics

Tags

Review + create

Summary



Logic App

by Microsoft

Details

Subscription	a27f2f09-058f-41f4-978c-6f278ee74f7e
Resource Group	appsvc_linux_centralus
Name	BrianTestApp
Region	westeurope
Log analytics	Disabled

After creating our logic app it will give you the option to go to the recourse.



Your deployment is complete



Deployment name: Microsoft.Web-LogicAppConsumption-Portal-6...

Subscription: [Azure for Students](#)

Resource group: [appsvc_linux_centralus](#)

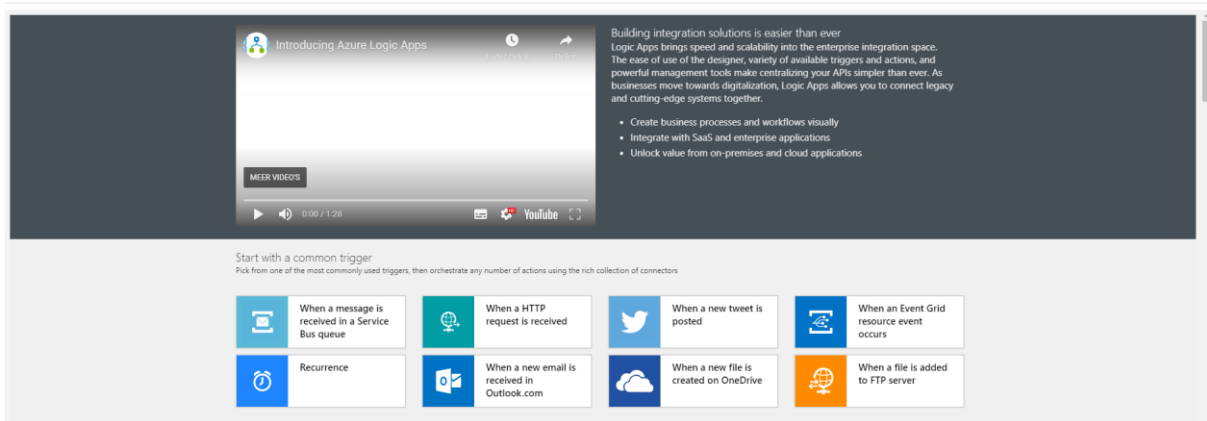
✓ **Deployment details** [\(Download\)](#)

^ **Next steps**

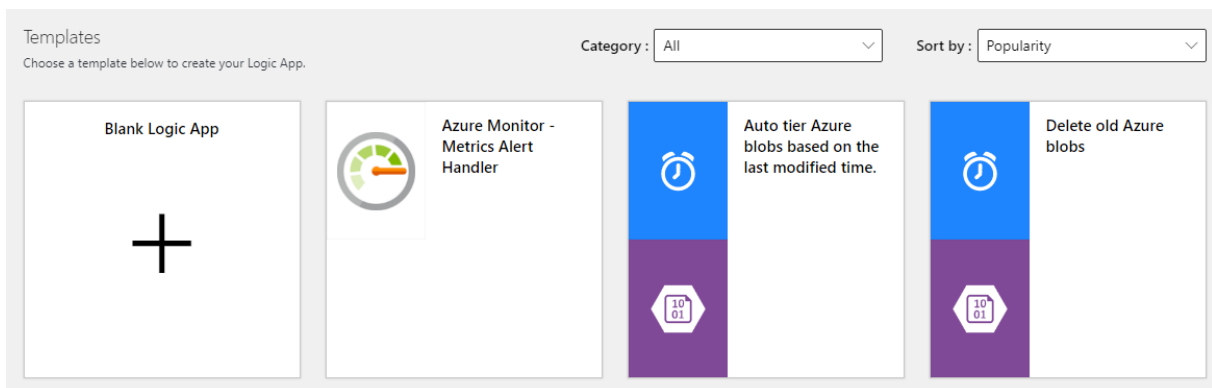
[Setup log analytics for your app.](#) Recommended

[Go to resource](#)

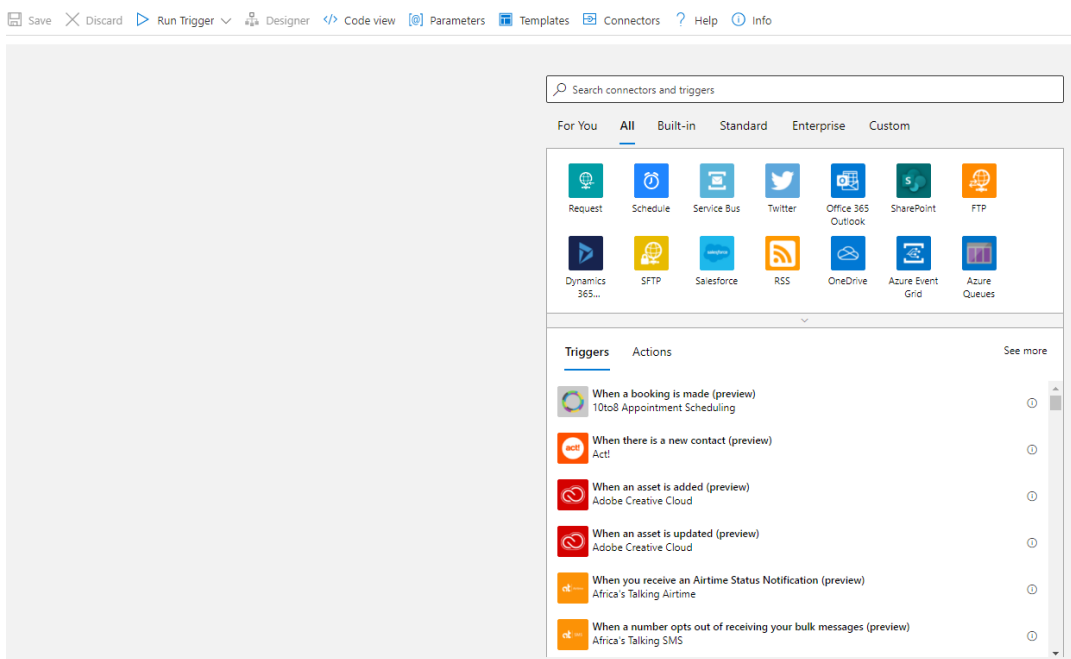
When going to the recourse you will see a small introduction video and a few common trigger option you can choose. I chose the blank logic app option



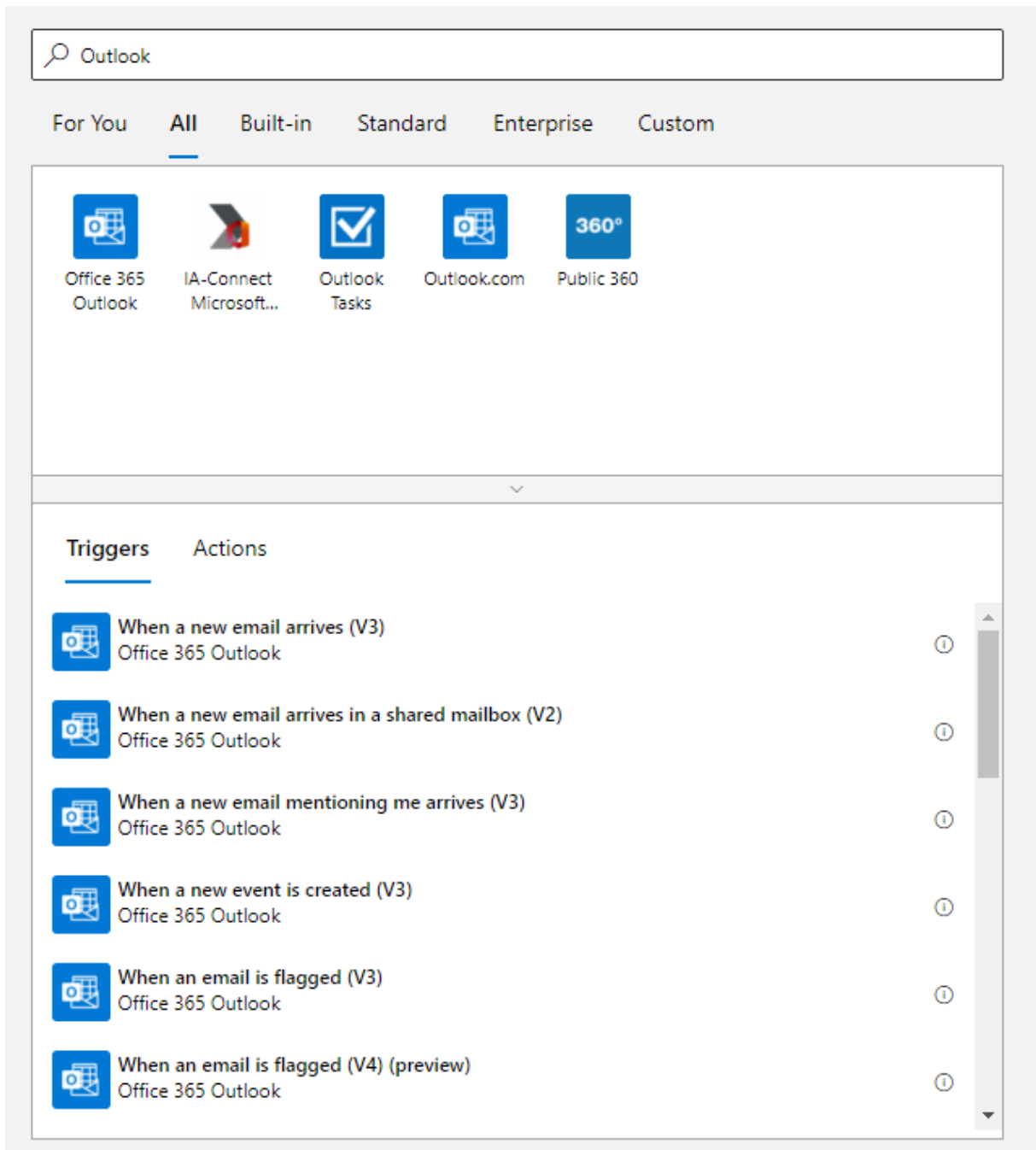
If you scroll down a little bit you can see the option to create a blank logic app I chose that one for my first logic app.



After selection the blank logic app it will open the editing screen for your logic app that is the screen where u can add triggers and action for the logic app.

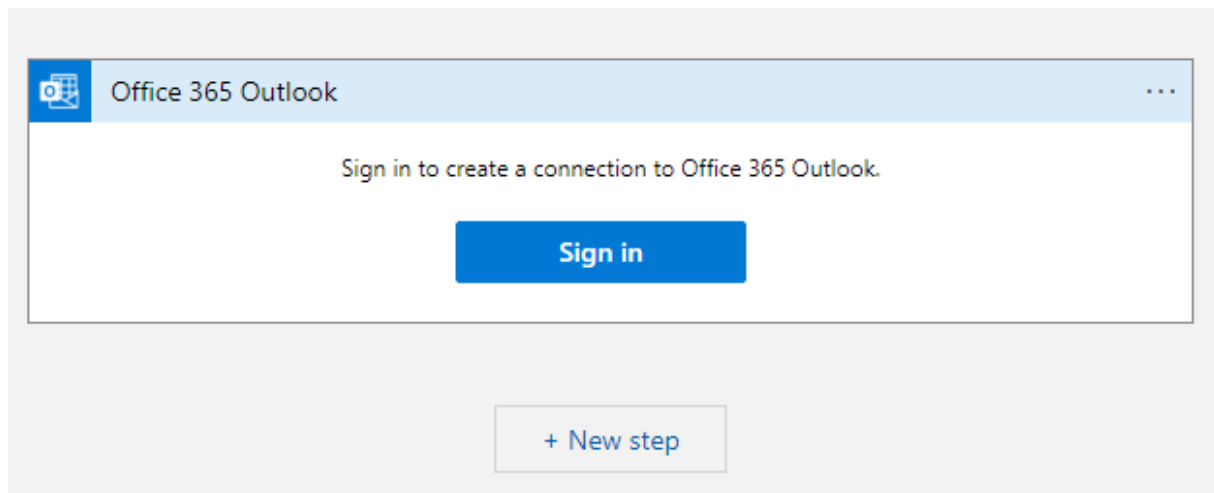


Now I want to create a logic app that uses outlook so the first thing I did was type outlook into the search bar.

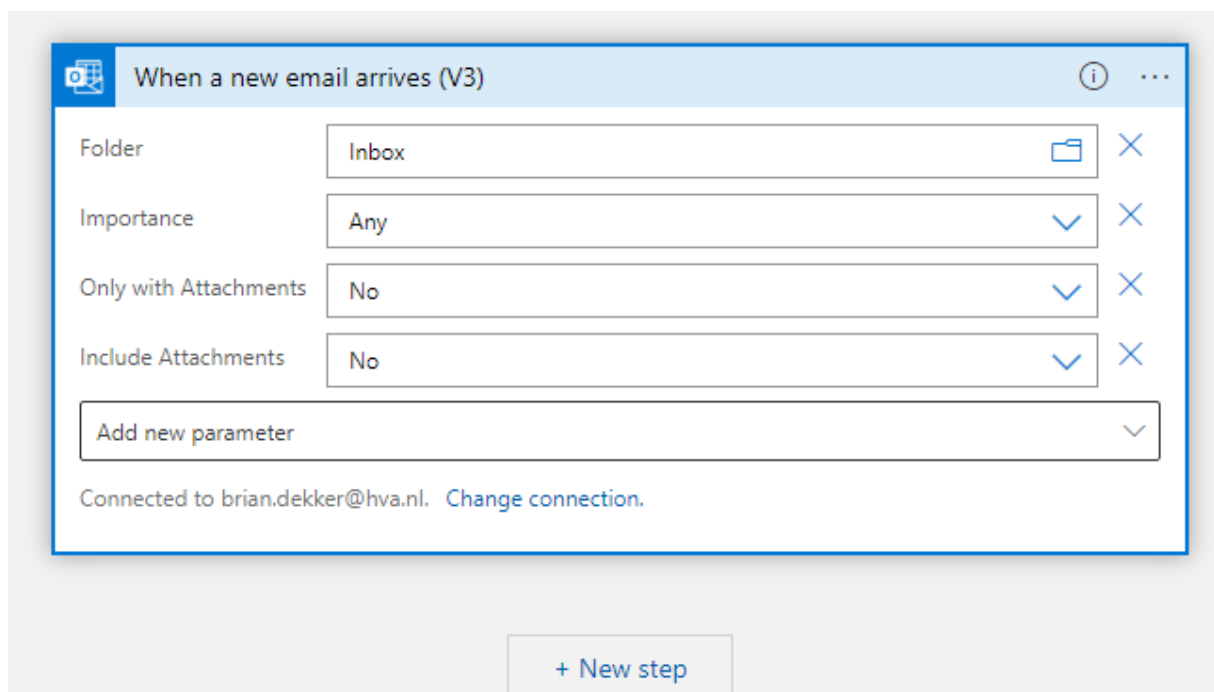


Now we can see all the triggers that can be used based on office 365 Outlook. For my first logic app I choose the top option trigger when a new email arrives (V3).

When u select that trigger it will ask you to login to a office 365 account in our case that would be our HvA account so in my case brian.dekker@hva.nl.

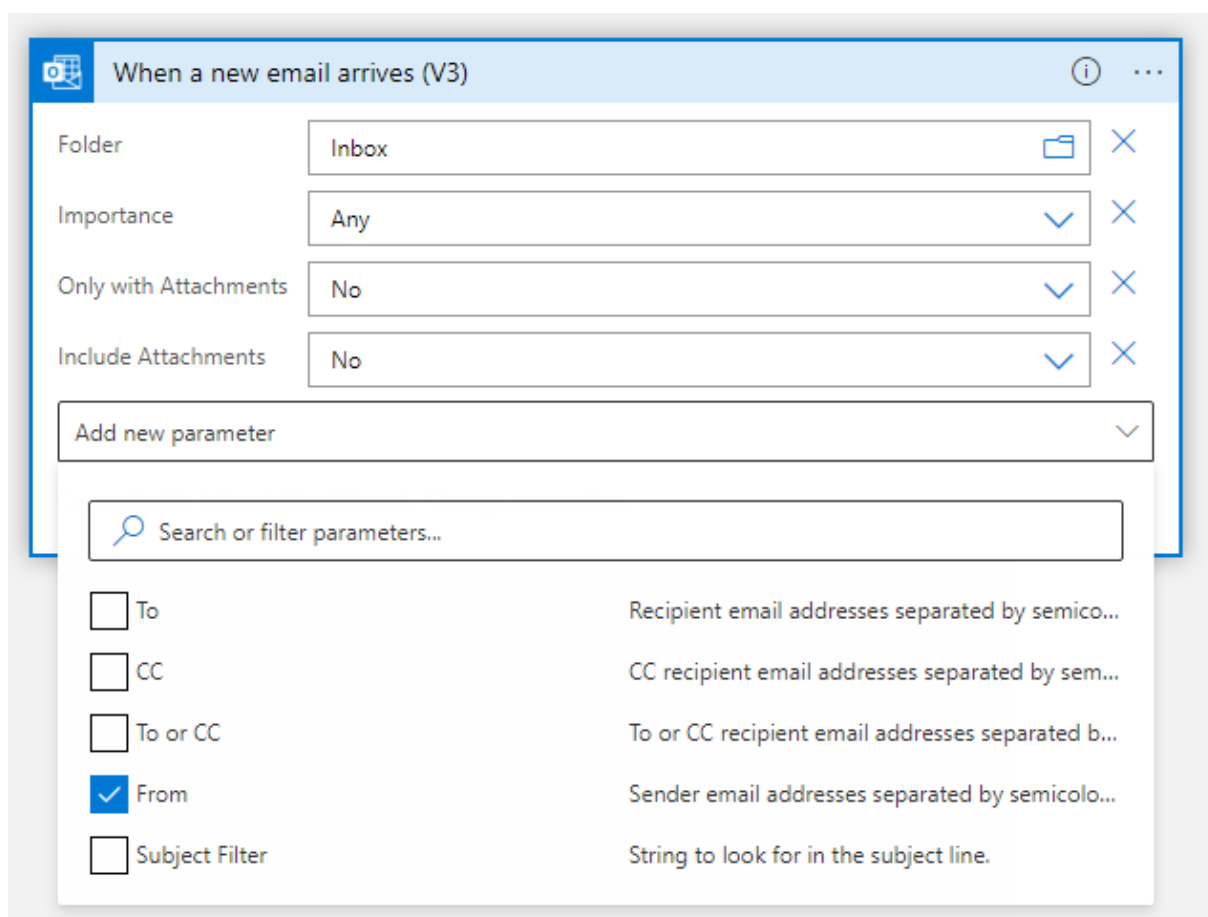


After login in you will see the option that are used in this trigger.



Now I kept it very simple I want to look at my inbox and when a new email arrives from my Gmail account I want to automatically respond.

In order to only trigger when the new email is from my gmail account we need to add a FROM parameter.



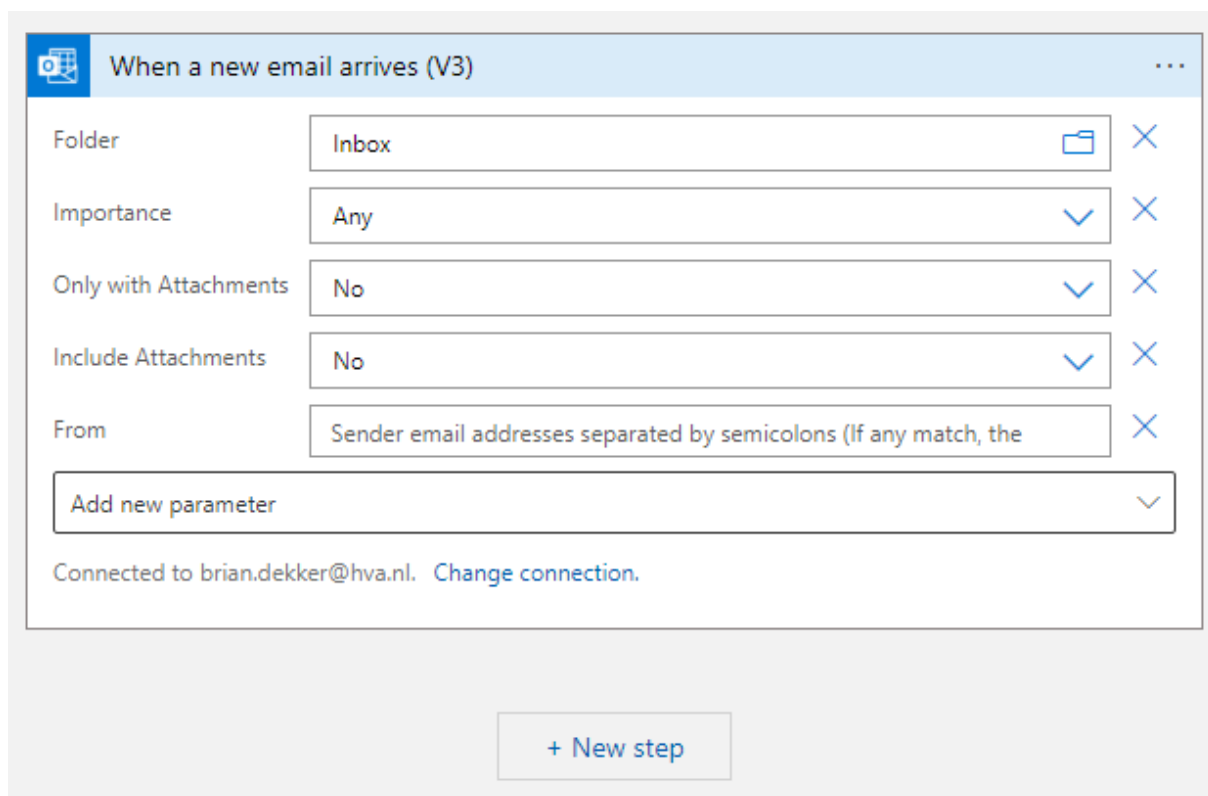
When a new email arrives (V3) ⓘ ...

Folder	Inbox	✕
Importance	Any	✕
Only with Attachments	No	✕
Include Attachments	No	✕

Add new parameter ▾

<input type="checkbox"/> To	Recipient email addresses separated by semicolons
<input type="checkbox"/> CC	CC recipient email addresses separated by semicolons
<input type="checkbox"/> To or CC	To or CC recipient email addresses separated by semicolons
<input checked="" type="checkbox"/> From	Sender email addresses separated by semicolons
<input type="checkbox"/> Subject Filter	String to look for in the subject line.

Now that we have selected that parameter it will show up in our trigger options.



When a new email arrives (V3) ...

Folder	Inbox	✕
Importance	Any	✕
Only with Attachments	No	✕
Include Attachments	No	✕
From	Sender email addresses separated by semicolons (If any match, the	✕

Add new parameter ▾

Connected to brian.dekker@hva.nl. [Change connection.](#)

+ New step

All I need to do now is add my gmail to the from option.

When a new email arrives (V3)

Folder

Inbox

Importance

Any

Only with Attachments

No

Include Attachments

No

From

briandekk@gmail.com

Add new parameter

Connected to brian.dekker@hva.nl. [Change connection.](#)

+ New step

Now our trigger step is done but a trigger alone is not enough we need to tell it what to do when the trigger is executed.

When you click new step it opens a new selection menu where u can search and select what u want the app to do after it is triggered.

When a new email arrives (V3)

Choose an operation

Search connectors and actions

For You

All

Built-in

Standard

Enterprise

Custom

Control

HTTP

Inline Code

Service Bus

SQL Server

Azure Functions

Office 365 Outlook

Triggers

Actions

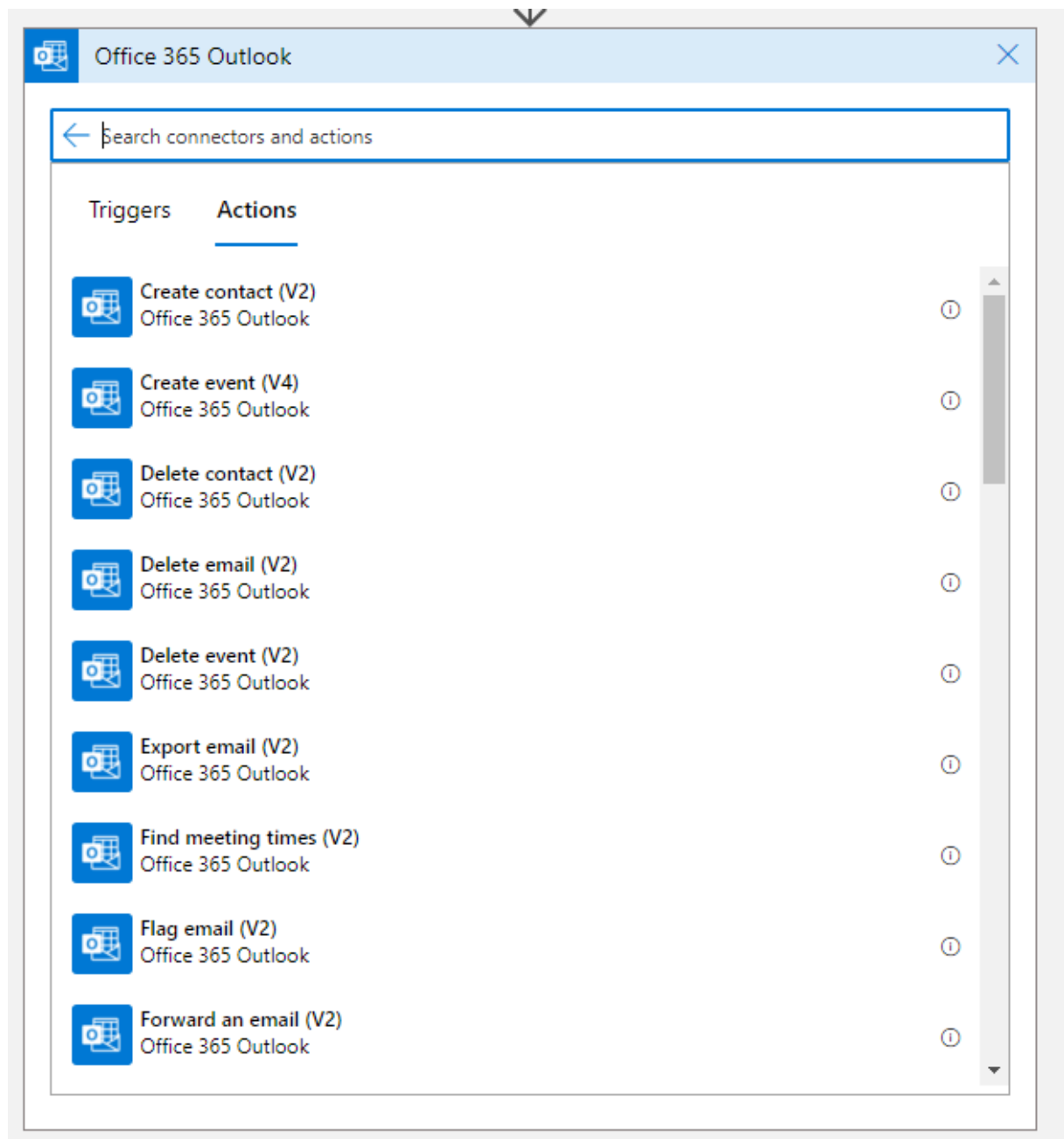
See more

Response Request

Extract document information (preview)
jexghts gen. Document & more

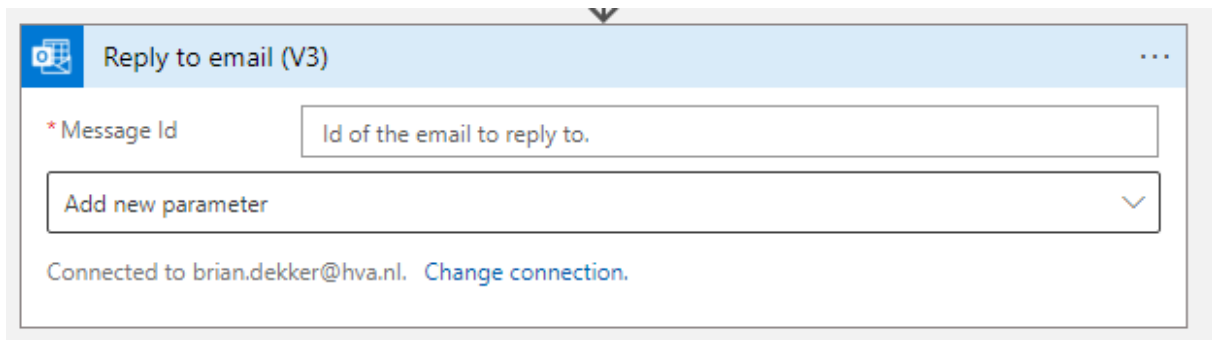
Generate document (preview)
jexghts gen. Document & more

Now we want to use office 365 outlook to respond to the email automatically so once again we search for outlook and select office outlook 365.

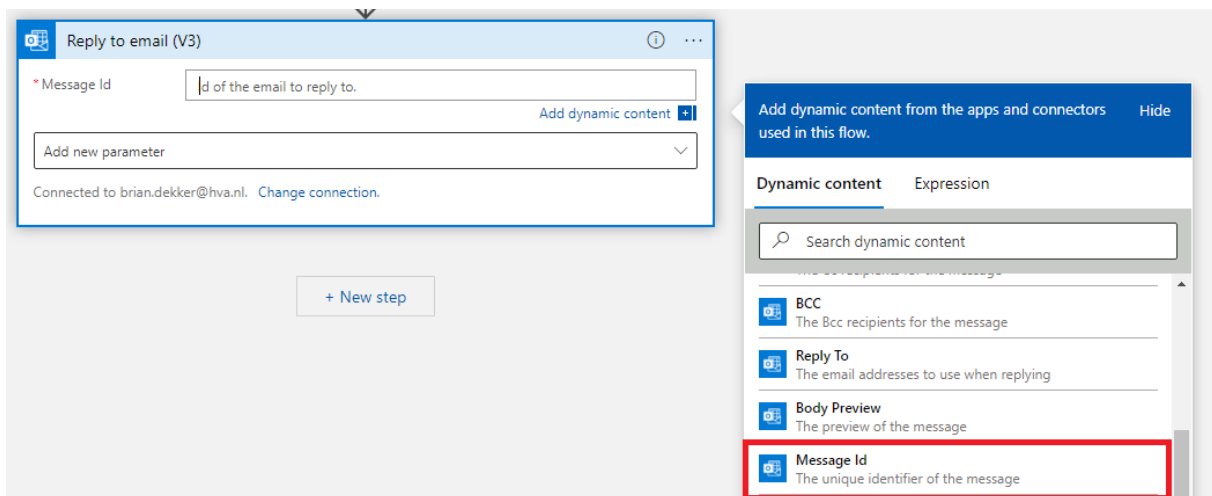


We now get a list of actions that are possible to use with outlook. The option I want is Reply to email (V3).

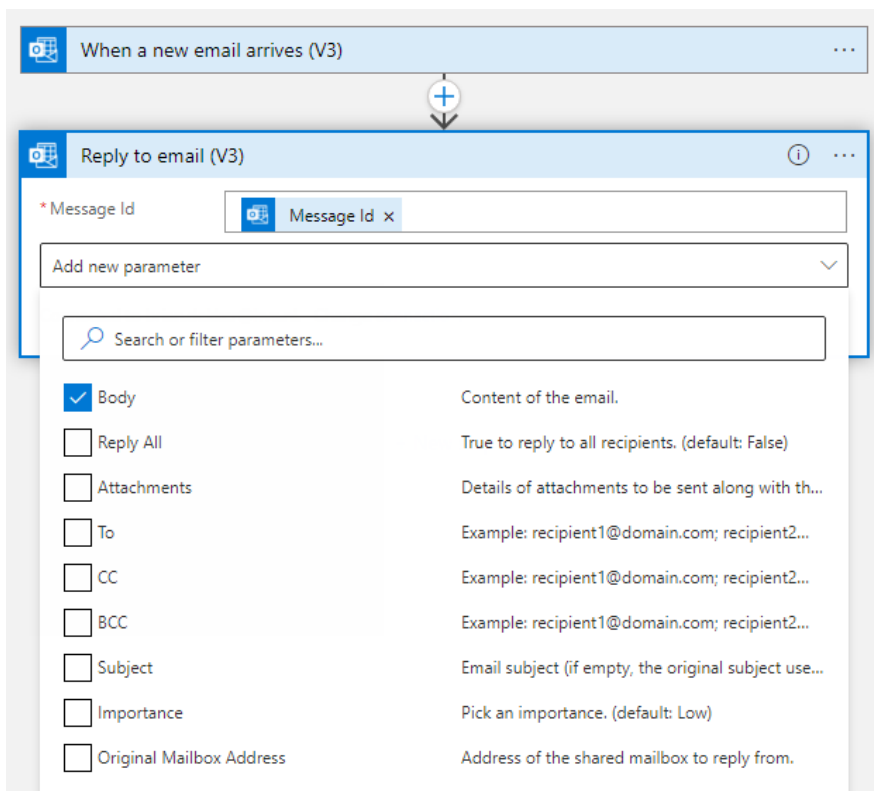




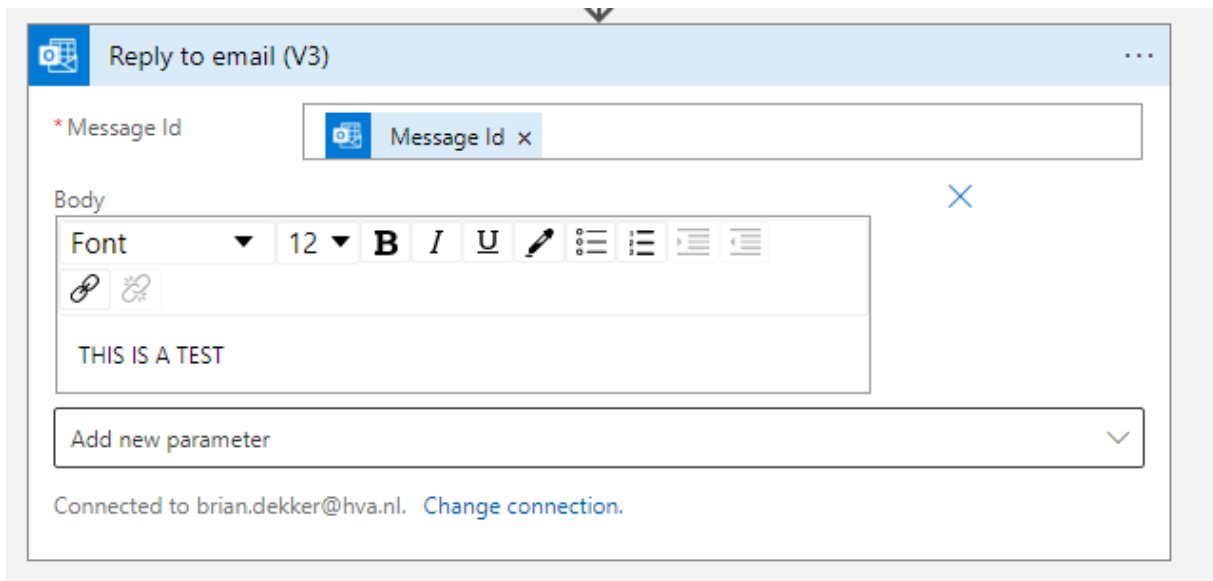
Now we get one option and that is to what email it should reply. If we click on the input box it will open a window on the side with option we are looking for the **messageId** option.



Now I also want to send something back so in the parameter menu I selected to add a body.

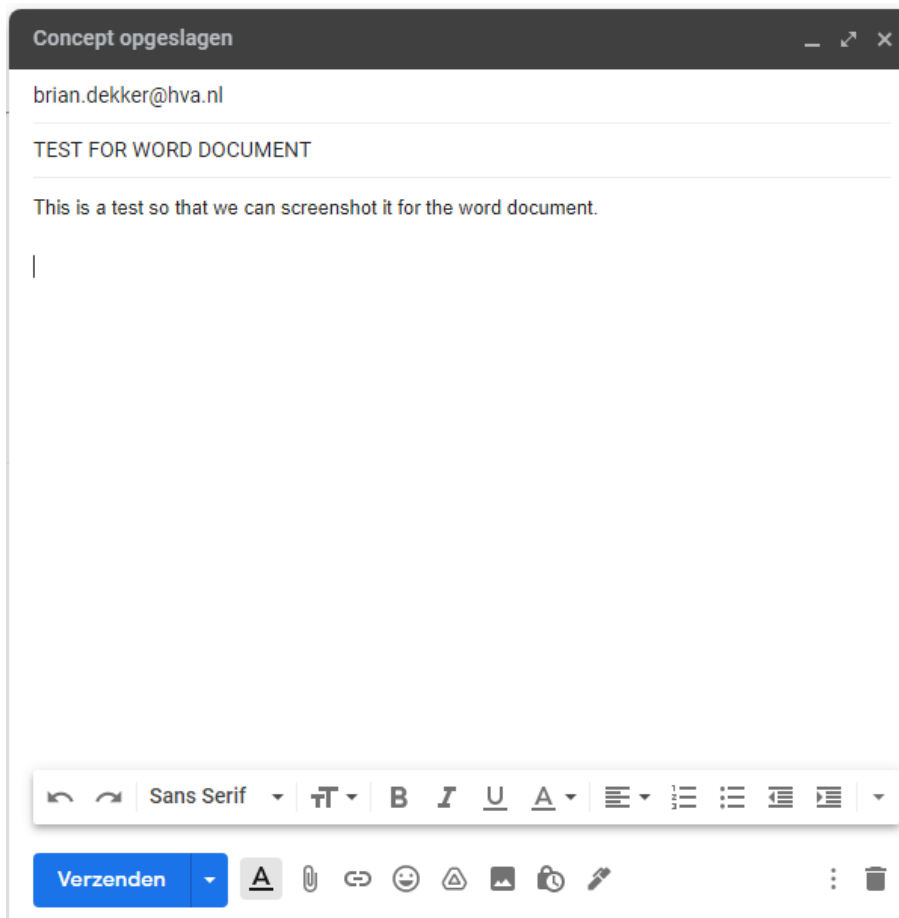


In this body you can put what ever text you want so for now I just put in THIS IS A TEST.

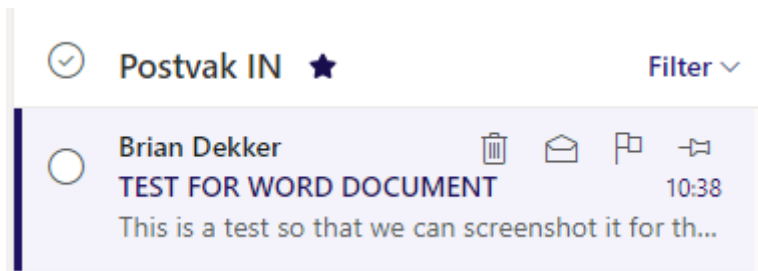


We can now click save at the top right and It should work so let's test it.

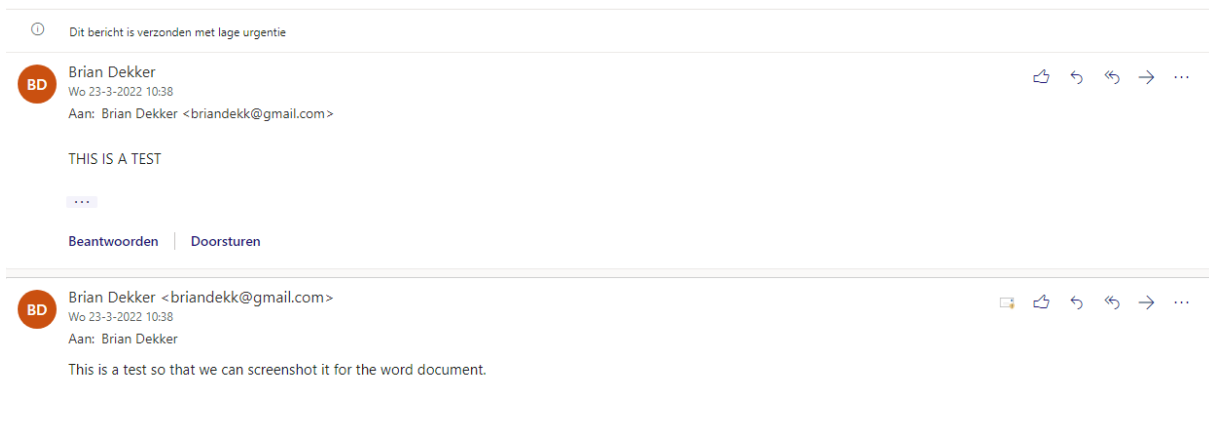
So we will send this email to my HvA outlook account.



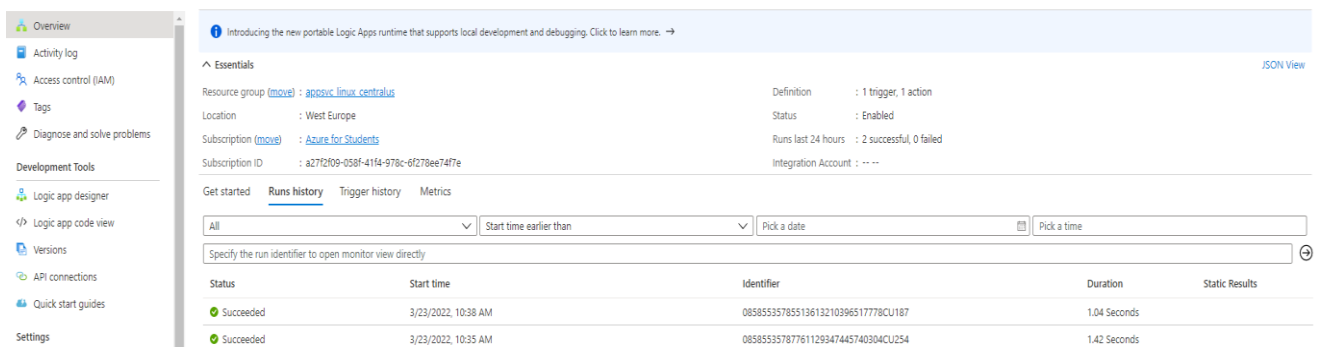
And I expect a reply within a minute.



We can see the email arrived and when we open it we can see that the reply has already been sent back so it's very fast. If you don't see that a reply has been send back yet just wait a little bit.



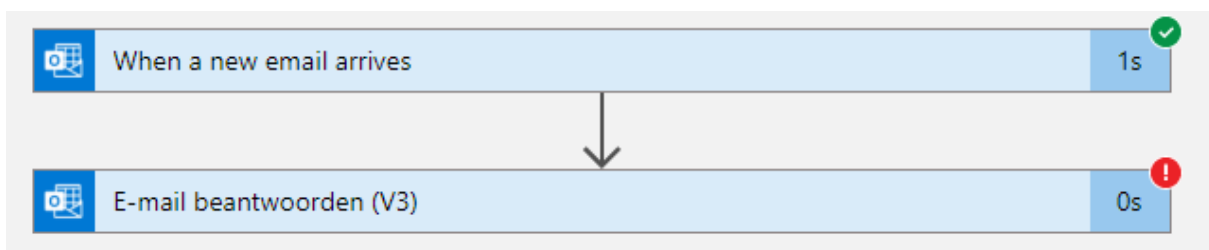
If the reply has been send then your logic app works correctly if the reply was not send that means something went wrong u can check what went wrong by going to your logic app overview and selecting the execution that has an error.



When a execution failed it should look something like this.

Failed	3/21/2022, 11:09 AM	08585537495073330427435278144CU31
Failed	3/21/2022, 11:06 AM	08585537496717461487336853729CU10

When you select your failed execution it will take you to your workflow and that lets you see where in the workflow the logic app failed.



You can then open that step and see what the error message says and hopefully solve that problem.

The screenshot displays a REST client interface with two main sections: **INPUTS** and **OUTPUTS**.

INPUTS

- Message Id**: `briandekk@gmail.com`
- Body**: `<p>Dit is een test</p>`

OUTPUTS

- Status code**: `400`
- Headers**:

Key	Value
Pragma	no-cache
x-ms-request-id	442cdecb-42d4-7fba-6006-af...
Strict-Transport-Security	max-age=31536000; includeS...
- Body**:

```
{
  "status": 400,
  "message": "De id heeft een onjuiste notatie.\r\nclientRequestId:",
  "error": {
    "message": "De id heeft een onjuiste notatie.",
    "code": "ErrorInvalidIdMalformed",
    "originalMessage": "Id is malformed."
  }
}
```

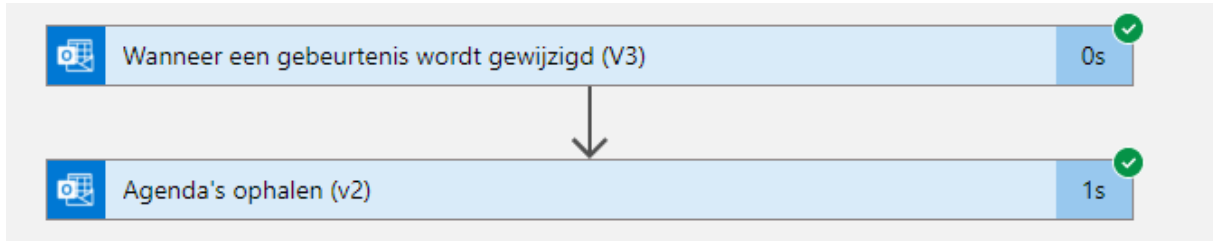
So in this case the message id was incorrect because I didn't know what variable it expected me to use when I did it for the very first time.

Creating a logic app that uses outlook calendar.

So now we want to create a logic app that is a more relevant to our learning story.

My first attempt.

My first attempt was if something changes in my calendar retrieve my calendar.



Now this works but it does not give me the output I want I lreilary gives you back the name of your calendars.

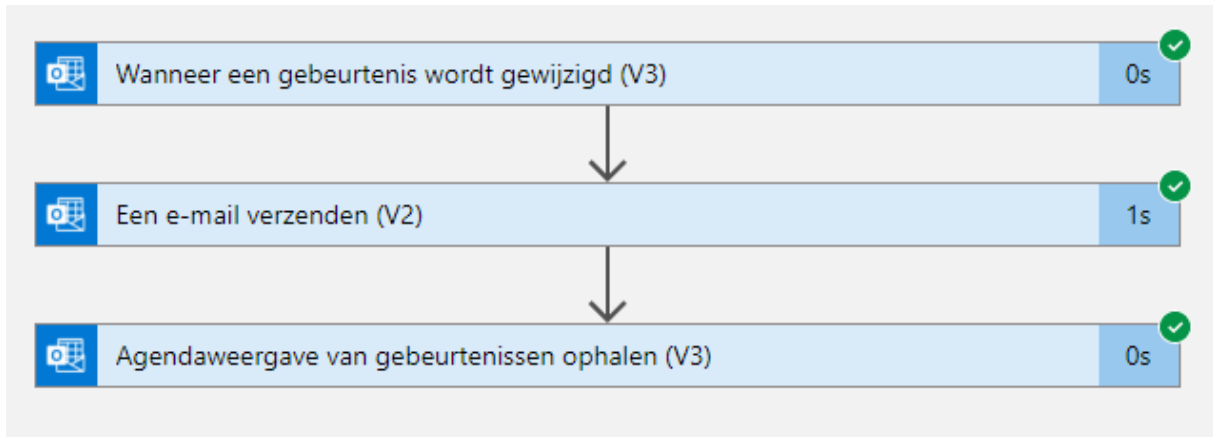
OUTPUT:

```
},
"body": {
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users('7b6e9e90-6494-4bc0-92df-198a0601e5ea')/calendars",
  "value": [
    {
      "id": "AAMkAGNhNM0Yzg0LTkxNmItNDc1Yi1iYWJhLWJhMzBlYjM3YzAyMgBGAAAAAAc7Fd32L6kR5POWFRaRV9_BwDr1c54YndvTL9GGqDod-TPAAAAAEGAADr1c54YndvTL9GGqDod-TPAABpeRU3AAA=",
      "name": "Agenda",
      "color": "auto",
      "hexColor": "",
      "isDefaultCalendar": true,
      "changeKey": "69X0eGJ3b0y/Rhqg6Hf0zwAAaX1Azw==",
      "canShare": true,
      "canViewPrivateItems": true,
      "canEdit": true,
      "allowedOnlineMeetingProviders": [
        "teamsForBusiness"
      ],
      "defaultOnlineMeetingProvider": "teamsForBusiness",
      "isTallyingResponses": true,
      "isRemovable": false,
      "owner": {
        "name": "Brian Dekker",
        "address": "brian.dekker@hva.nl"
      }
    },
    {
      "id": "AAMkAGNhNM0Yzg0LTkxNmItNDc1Yi1iYWJhLWJhMzBlYjM3YzAyMgBGAAAAAAc7Fd32L6kR5POWFRaRV9_BwDr1c54YndvTL9GGqDod-TPAAAAAEGAADr1c54YndvTL9GGqDod-TPAABpeRU4AAA=",
      "name": "Feestdagen in Nederland",
      "color": "auto",
      "hexColor": "",
      "isDefaultCalendar": false,
      "changeKey": "69X0eGJ3b0y/Rhqg6Hf0zwAAaX1A0A==",
      "canShare": false,
      "canViewPrivateItems": true,
      "canEdit": false,
      "allowedOnlineMeetingProviders": [],
      "defaultOnlineMeetingProvider": "unknown",
      "isTallyingResponses": false,
      "isRemovable": true,
      "owner": {
```


So I needed something else.

My second attempt.

My second attempt was more what I wanted as a output it actually returns me the created/edited event back.



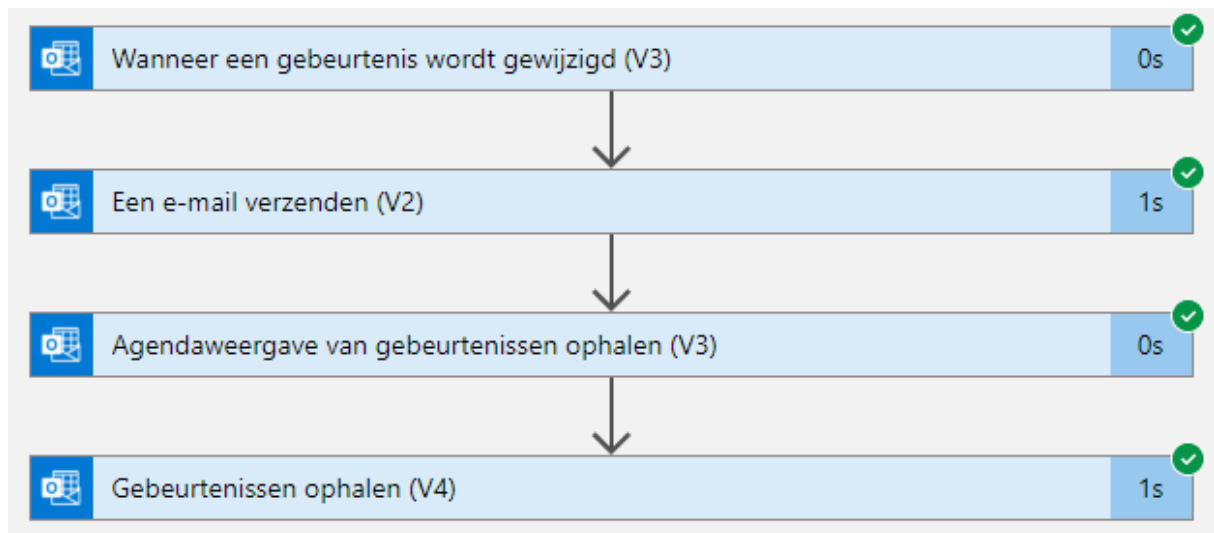
OUTPUT:

```
''
"body": {
  "value": [
    {
      "subject": "TEST 55",
      "start": "2022-03-24T07:00:00.0000000",
      "end": "2022-03-24T07:30:00.0000000",
      "startWithTimeZone": "2022-03-24T07:00:00+00:00",
      "endWithTimeZone": "2022-03-24T07:30:00+00:00",
      "body": "<html>\r\n<head>\r\n<meta http-equiv=\"Content-Type\" content=\"t",
      "isHtml": true,
      "responseType": "organizer",
      "responseTime": "0001-01-01T00:00:00+00:00",
      "id": "AAMkAGNhNWM0Yzg0LTkxNmItNDc1Yi1iYWJhLWJhMzBlYjM3YzAyMgBGAAAAAAAc7Fd",
      "createdDateTime": "2022-03-22T17:47:26.942382+00:00",
      "lastModifiedDateTime": "2022-03-22T18:14:32.8848433+00:00",
      "organizer": "brian.dekker@hva.nl",
```

The output is a lot bigger then this but this is basically the event that I created/edited.

Attempt number 3.

Now that I had the correct output I wanted to see if I could use it somehow unfortunately it was not as simple as I thought there is no option to use the output of a previous set in your next step or at least I have not found a way to do that yet. But I did try a couple different actions that return events on the calendar the most relevant one was Retrieve events (V4) this one when run return more then one event in the calendar when the logic app triggers this might be useful to retrieve schedules for an entire day.

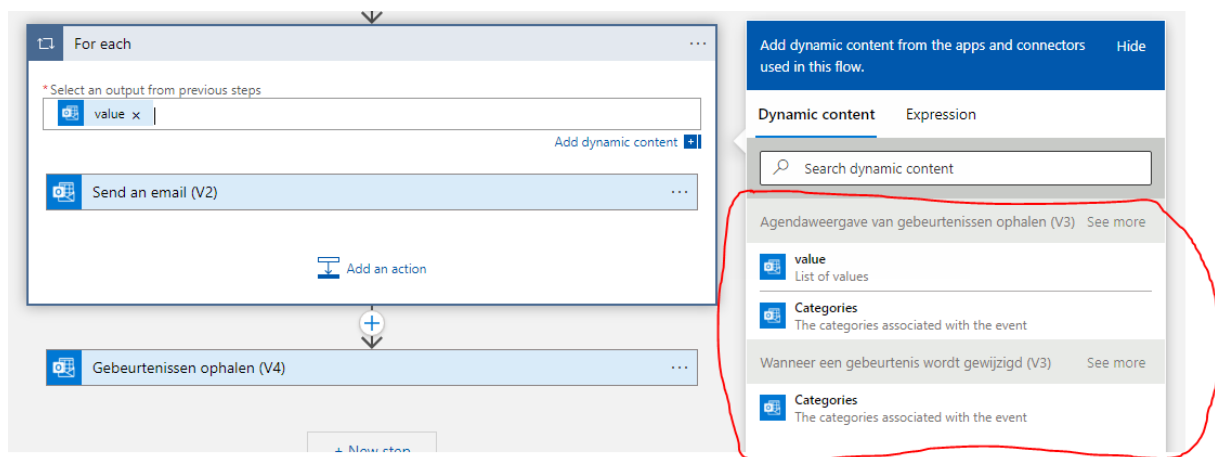


The output of this step is way to big to copy into word but its like attempt 2 but more then just the one created/edited event.

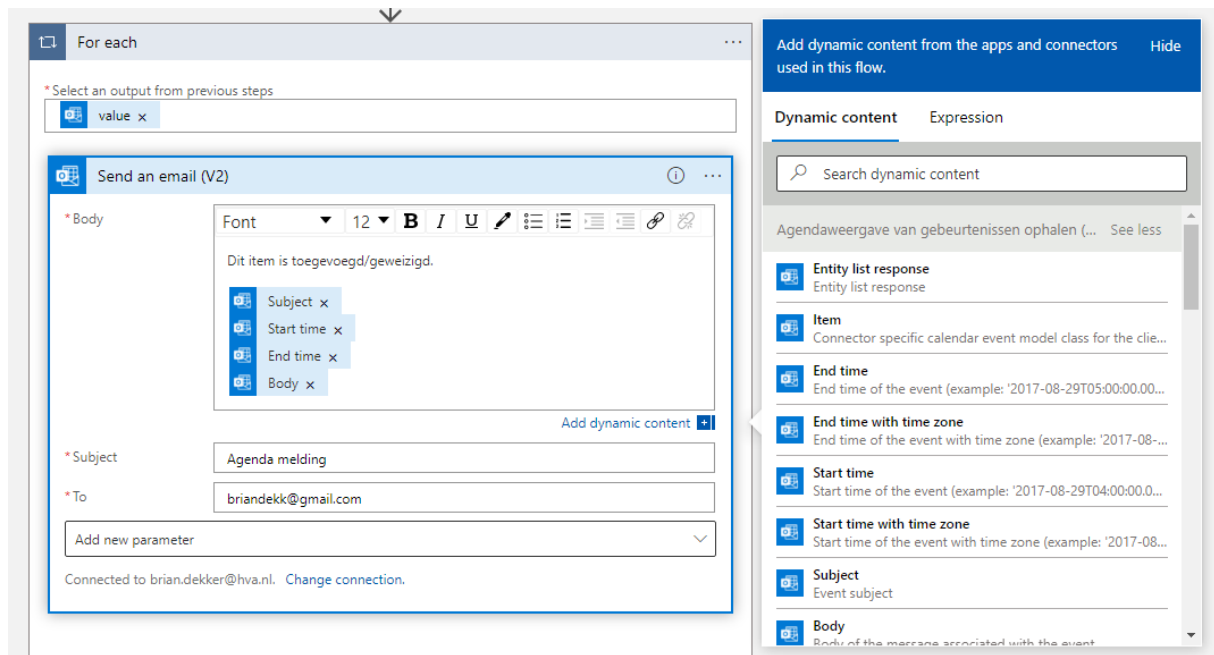
Attempt 4

I now wanted to attempt sending the output of attempt 2 in the email I send myself kind of like a notification system.

So I have now discovered that how to use values from previous steps (Microsoft can really use some interface improvements on this front.)



So in the red area you can see your previous steps and the values that u can use from those steps now the SEE MORE option is very important here even tough it is kind of faded into the background.



As you can see now I use several different variables that I extract from the output if the previous step in the body of my notification email whenever an event is created or changed.

Whenever I add or change an event this is the email I now get:

Agenda melding ➡ Inbox x

Brian Dekker

aan mij ▾

Dit item is toegevoegd/gewezigd.

TEST 6

2022-03-24T07:00:00.0000000

2022-03-24T09:30:00.0000000

Ik heb de titel en tijd en body aangepast.

