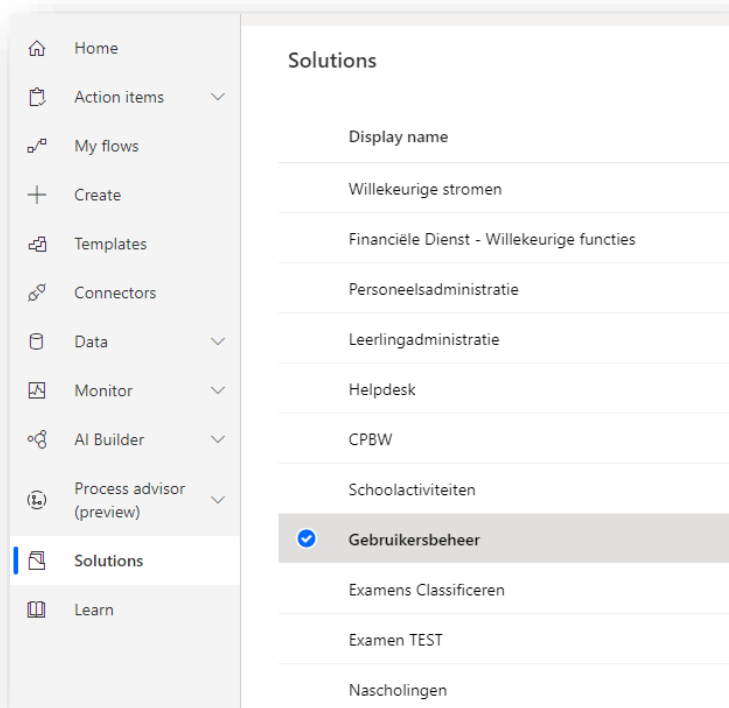


1. Solution

Alle Flows + de PowerApp werden aangemaakt in een aparte Solution. Dit om het overzicht te blijven bewaren.



Uiteindelijk had ik nood aan 1 PowerApp en 3 Flows (de overige Flows waren als test en zijn dus niet nuttig).

Display name	Name	Type
Smartschool co-accounts	crdc7_smartschoolcoaccounts_fe0c1	Canvas app
PowerApp -> HTTP, Variabele initialiseren - Reactie, Vari...	PowerApp -> HTTP, Variabele initialiseren - Reactie, Variabele ir	Cloud flow
PowerApp -> HTTP, Variabele initialiseren - Reeks, Varia...	PowerApp -> HTTP, Variabele initialiseren - Reeks, Variabele init	Cloud flow
Smartschool - getUserDetailByUsername v2	Smartschool - getUserDetailByUsername v2	Cloud flow
Smartschool - saveInfoCoAccounts	Smartschool - saveInfoCoAccounts	Cloud flow
Smartschool - saveUserParameter	Smartschool - saveUserParameter	Cloud flow

2. Powerapp

Co-account gegevens opvragen

Voornaam leerling

Naam leerling

E-mailadres leerling @student.tsaam.be

Haal gegevens op **Kier e ki were**

Intern nummer

Co-account 1

Voornaam

Naam

E-mailadres

Type

Status

Laatst ingelogd

Nieuw wachtwoord

Wachtwoord co-account 1 resetten

Co-account 2

Voornaam

Naam

E-mailadres

Type

Status

Laatst ingelogd

Nieuw wachtwoord

Wachtwoord co-account 2 resetten

Vraagt naam en voornaam van de leerling waarvan je de co-accounts wil wijzigen. De knop 'Haal gegevens op' roept de Flow 'GetUserDetailByUsername v2' op met als parameter het emailadres van de leerling. Het is aan de gebruiker van de Flow om de naam correct en zonder accenten te schrijven.

OnSelect van de knop:

```
OnSelect = fx ClearCollect(UserDetails, 'Smartschool-getUserDetailByUsernamev2'.Run('Label-E-mailadresLLNIngevuld'.Text))
```

Het antwoord van de Flow wordt gebruikt om de onderste velden in te vullen. Deze zitten vervat in een Gallery. Deze Gallery krijgt als Items de collectie 'UserDetails' (zie hierboven, het antwoord van de Flow wordt opgeslagen in deze collectie). Hieruit worden dan de nodige velden gelezen:

De Gallery:

Items

UserDetails

Veld 'Voornaam' van co-account 1.

DATA

Default

ThisItem.voornaam_coaccount1

De velden 'Voornaam', 'Naam', 'E-mailadres' en 'Nieuw wachtwoord' kunnen aangepast worden. Door te drukken op de knop 'Wachtwoord co-account 1 / 2 resetten' worden de gegevens in Smartschool aangepast via de Flow 'SaveInfoCoAccounts'.

Indien het veld '*Nieuw wachtwoord*' werd leeggelaten, wordt er automatisch een nieuw wachtwoord gegenereerd dat doorgestuurd wordt.

OnSelect van de knoppen:

```
OnSelect = fx 14(TextInput1.Text = "",Set(wachwoord1,TextInput2.Text&"19-20*"),
Set(wachwoord1,TextInput2.Text));Set(Antwoord,"Smartschool-saveInfoCoAccounts".Run(1,ThisItem.Internnummer,TextInput2.Text,TextInput2_1.Text,TextInput2_2.Text,wachwoord1,ThisItem.gebruikersnaam));Set(Reset1,true)
```

3. GetUserDetailByUsername

The screenshot shows a PowerApps flow with the following steps:

- HTTP Action:**
 - Method: POST
 - URI: `https://tsaam.smartschool.be/Webservices/V3`
 - Headers:
 - Content-Type: `text/xml; charset=UTF-8`
 - SOAPAction: `https://tsaam.smartschool.be/Webservices/V3#getUserDetailsByUsername`
 - Body: XML SOAP request for `v3:getUserDetailsByUsername` with parameters `accesscode` and `username`.
 - Cookie: Enter HTTP cookie
- Variable initialization - UserDetail:**
 - Name: UserDetail
 - Type: String
 - Value: `replace(...)`
- Variable initialization - JSON:**
 - Name: JSON
 - Type: String
 - Value: `xpath(...)`
- Reactie (Response):**
 - Status Code: 200
 - Headers:
 - Content-Type: `application/json`
 - Body: `replace(...)`

De Flow wordt opgeroepen vanaf een PowerApp en krijgt 1 parameter mee, namelijk het e-mailadres van de leerling. Dit e-mailadres is binnen Smartschool ook zijn gebruikersnaam.

We doen een HTTP aanvraag richting Smartschool. [Zie de handleiding voor uitleg rond de mogelijke aanvragen en hun parameters.](#)

In bovenstaand geval geven we enkel ons Webserviceswachtwoord en de gebruikersnaam van de leerling mee.

De eerste variabele 'UserDetail' haalt de hoofding weg van het antwoord van Smartschool via volgende functie:

```
replace(replace(body('HTTP'), '<?xml version="1.0" encoding="utf-8"?>', ''), '<?xml version="1.0" encoding="UTF-8"?>', '')
```

Dit wordt dan op zijn beurt als XML ingelezen via

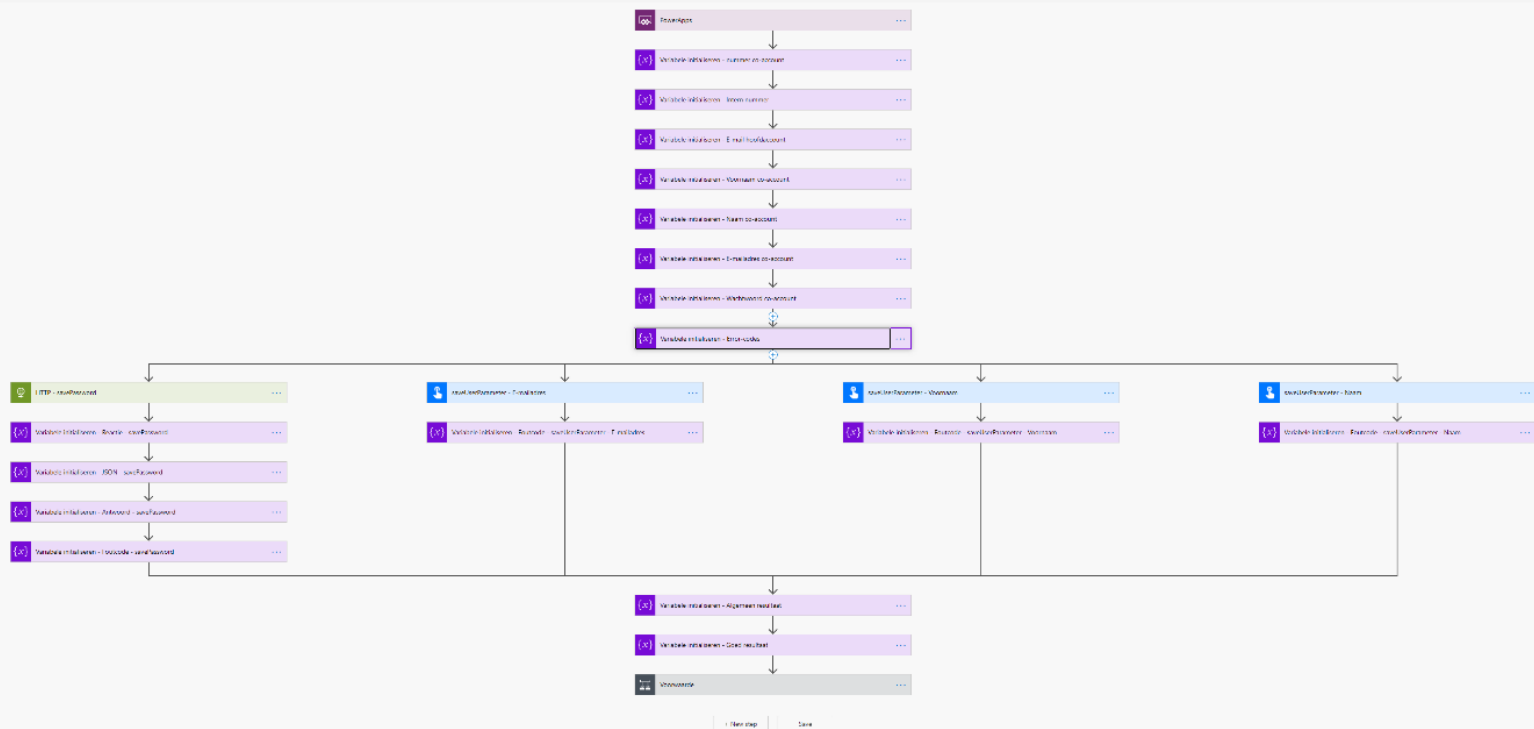
```
xpath(xml(variables('UserDetail')), '//*[local-name()="return"]/text()')
```

Die dan wordt teruggegeven aan de PowerApp na een kleine aanpassing:

```
replace(replace(replace(variables('Json'), '[' , '' ), ']' , '' ), '\"', '')
```

Het uiteindelijk resultaat is iets dat PowerApps kan uitlezen.

4. saveInfoCoAccounts



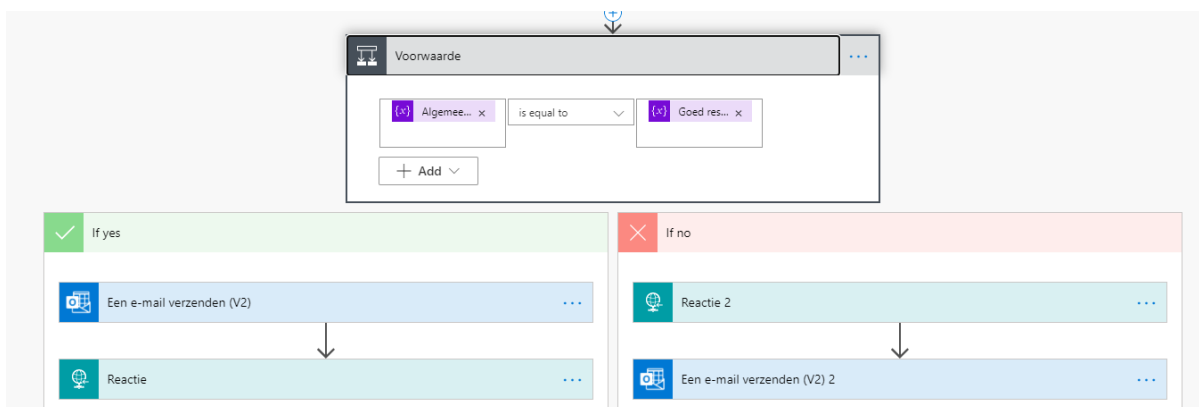
Deze flow gebruikt wat meer parameters, zie de eerste paar stappen.

De *error-codes* werden hardgecodeerd (kan je ook via de webservice ophalen, maar zag ik het nut niet van in).

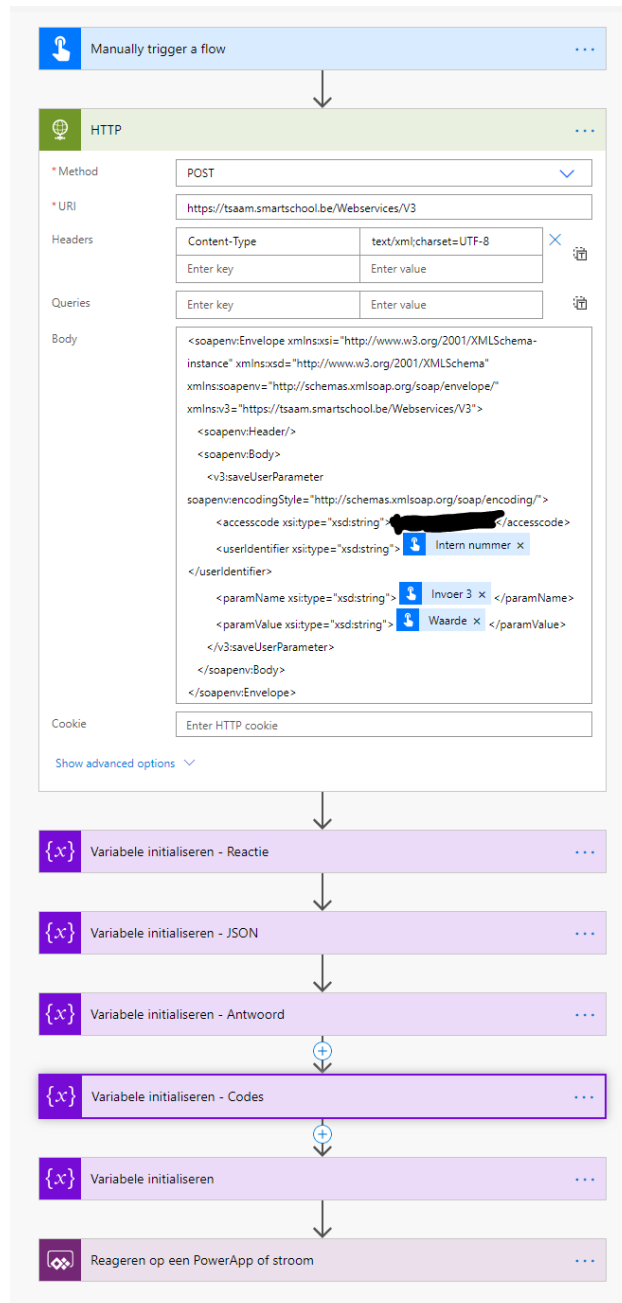
Voor iedere variabele die kan aangepast worden, wordt een HTTP aanvraag naar Smartschool verzonden. Dit gebeurt op dezelfde manier als bij de vorige Flow. Alleen gebruiken we nu de functie [savePassword](#) en de functie [saveUserParameter](#). Deze laatste wordt via een extra Flow opgeroepen (herbruikbaar, dus slechts 1x gemaakt en dan telkens opgeroepen indien nodig).

Resultaat van al deze Flows is telkens een getal. Dit kunnen we omzetten tot een string via de variabele *'Errorcodes'*. Indien het uiteindelijk resultaat 4x *'De bewerking is gelukt'* is, dan stuurt de Flow een mail naar het e-mailadres van het co-account dat werd aangepast en wordt er gereageerd naar de PowerApp met een positieve melding.

Indien er ergens iets foutliep, wordt een mail naar mij gestuurd en krijgt de gebruiker deze foutmelding ook te zien in hun PowerApp.



5. saveUserParameter



Gebruikt de functie 'saveUserParameter' om een waarde van een account aan te passen. Via deze functie kunnen bv. 'Voornaam' en 'Naam' van een co-account aangepast worden.