

## **Nadere informatie API koppeling**

Ten behoeve van



**Aloysius Stichting**

5 januari 2021  
TenderNed-kenmerk: 288477

© januari 2021, Telecomvisie

## Uitleg ten aanzien van gewenste functionaliteit

### **Portal**

In het geval dat medewerkers toegang krijgen tot een portal van de telecom provider (bijvoorbeeld voor verbruik gegevens en opvragen van een e-sim, puk code etc), dan is het noodzakelijk dat de aanbieder een SSO connectie kan maken met de Azure AD omgeving van Aloysius. Medewerkers kunnen dan op basis van hun Aloysius eigen account en wachtwoord inloggen in de omgeving van de aanbieder.

### **Archipel**

PepperByte maakt het voor Aloysius mogelijk om koppeldiensten als services op een snelle, veilige, betrouwbare én betaalbare manier beschikbaar te stellen. We vermenigvuldigen de waarde van (bedrijfs)data en functies door deze veilig, op ieder moment en vanaf elk device toegankelijk te maken. Onze kracht ligt in het in staat stellen van zowel bedrijven als overheden om samen te werken en digitale processen en diensten richting eindgebruikers te verbeteren.

Om dit mogelijk te maken hebben wij een geavanceerde integratie-oplossingen gecreëerd, aangevuld met specialistische, professionele diensten. Dit portfolio stelt onze klanten in staat om gegevens en functies vanuit een centrale plek in de organisatie beschikbaar te stellen aan haar aanbieders en afnemers. Op deze manier hoeven deze partijen zelf geen energie (tijd en geld) meer te steken in de ontwikkeling en beveiliging van hun koppelvlakken, terwijl ze toch op het hoogste niveau kunnen integreren.

## Voorbeelden processen die geautomatiseerd kunnen worden

### **Onboarden van een medewerker**

In het geval dat er een nieuwe medewerker start zal er op basis van rol, functie of status (geen externe of stagiair) een 06 abonnement aangevraagd worden. De Archipel verwerkt de nieuwe medewerker, zal een API call doen naar de telecom provider (bijvoorbeeld voornaam, achternaam, locatie) en vraagt (via de API) bij de telecom provider een nieuwe abonent aan. De response van de telecom provider is het contract nummer en een 06 nummer. De Archipel verwerkt het 06 nummer in alle systemen die het 06 nummer nodig hebben, tevens zal het 06 nummer verwerkt worden in de Office 365 omgeving van Aloysius. Dit zal dan de centrale plek voor medewerkers zijn om de collega's te kunnen bereiken.

### **Offboarden van een medewerker**

Een medewerker gaan uit dienst. De Archipel verwerkt de nieuwe medewerker, zal een API call doen naar de telecom provider (op basis van contract of 06 nummer) en de API vragen het abonnement stop te zetten op basis van de datum uitdienst.

### **Locatieslijst**

Locatie lijst voor facturering en n.a.w. gegevens. Ook dit kan geautomatiseerd worden via een API. De Archipel zal controleren of de locaties nog up-to-date zijn, wanneer dit niet het geval is zullen wij de locatie geautomatiseerd updaten.

### **Proces**

Wij voeren in grote lijnen de volgende stappen uit:

1. Ophalen van de count (CurrentCount) en vergelijken met voorgaande count (LastCount);
2. Wanneer de tellingen overeen komen dan halen wij alle items op van het endpoint;
3. We doen een check op de array count ten opzichte van de CurrentCount;
4. Wij lopen alle items na en controleren op wijzigingen;
5. We verwerken alle wijzigingen (nieuwe of verwijderde items en gewijzigde values).
6. Het dient een REST API te betreffen of dat er gebruik gemaakt wordt van Swagger.

## API SPEC Voorbeeld

### **voorbeeld**

Authenticatie

OAuth Token

Omgevingen

- Production
- Acceptance
- Development

Endpoints:

- POST /api/<versie\_nummer>/counts

Een manier om per <type\_naam> de counts op te halen, met een response zoals:

```
{
  "type_naam_1": 100,
  "type_naam_2": 200
}
```

- POST /api/<versie\_nummer>/reset

Een manier om de volledige database te resetten voor development doeleinden

Endpoints PER object/type:

- POST /api/<versie\_nummer>/<type\_naam>
- PATCH /api/<versie\_nummer>/<type\_naam>/<id>
- DELETE /api/<versie\_nummer>/<type\_naam>/<id>

Ophalen van Medewerkers (GET)

Request:

GET /api/v1/Medewerkers HTTP/1.1

Response:

HTTP/1.1 200 OK

```
[
  {
    "total_results": 100,
    "total_pages": 10,
    "page_size": 10,
    "current_page": 1
  }
  {
    "id": 1,
    "Voornaam": "Pietje",
    "Achternaam": "Puk"
  },
  {
    "id": 2,
    "Voornaam": "Klaas",
    "Achternaam": "Vaak"
  }
]
```

]

#### Toelichting

Bij de initiële GET (pagina 1) geef je naast de data ook een pagination object mee welke het totaal aantal objecten als count bevat en het aantal pagina's (bereken dit door de query parameter `?page_size` af te lezen)

Een voorbeeld: ophalen van alle Medewerkers:

- 1) GET `/api/v1/Medewerkers?page_size=10` (de response bevat data van de eerste pagina + het aantal pagina's)
- 2) GET `/api/v1/Medewerkers?page_size=10&page=2`
- 3) GET `/api/v1/Medewerkers?page_size=10&page=3`
- 4) GET `/api/v1/Medewerkers?page_size=10&page=4`

#### Aanmaken van een Medewerker (POST)

Request:

POST `/api/v1/Medewerkers` HTTP/1.1

```
{  
  "Voornaam": "Pietje",  
  "Achternaam": "Puk"  
}
```

Response:

HTTP/1.1 201 Created

Location: `<domein>/api/v1/Medewerkers/<id>`

#### Toelichting

Bij het succesvol aanmaken moet er aan de API kant een numeriek ID worden gegenereerd. Dit ID moet worden meegegeven in de header 'Location' zodat de aan onze kant deze opgeslagen kan worden en waar we later op zullen koppelen.

#### Bijwerken van een Medewerker (PATCH)

Request:

PATCH `/api/v1/Medewerkers/<id>` HTTP/1.1

```
{  
  "Voornaam": "Pietje",  
  "Achternaam": "Puk"  
}
```

Response:

HTTP/1.1 204 No Content