

# API endpoint voor Eduhub-RIO-mapper

Joost Diepenmaat, Remco van 't Veer

2022-07-06

## **Geschiedenis**

2022-07-01	Eerste versie
2022-07-05	POST prefix toegevoegd
2022-07-06	Program en Course verplicht met EducationSpecification

Dit document beschrijft de API van de Eduhub-RIO-mapper. Hierin is te vinden hoe updates doorgegeven worden en de status van deze updates opgevraagd kan worden.

## Standen en hoofdobjecten

Updates van gegevens van een opleider worden altijd als "stand" doorgegeven aan RIO. Een stand wordt altijd opgemaakt op basis van een hoofdobject. In deze koppeling worden de volgende typen objecten als hoofdobject aangemerkt:

- EducationSpecification (RIO OpleidingEenheid)  
Opmerkingen:
  - EducationSpecification kan genest worden
- Education (Program of Course) (RIO AangebodenOpleiding)  
Opmerkingen:
  - Program kan genest worden
  - Course kan aan meerdere Programs gekoppeld zijn
  - Program kan meerdere Courses bevatten
  - Program en Course moeten gekoppeld zijn aan een EducationSpecification

Bij updates van een Course of Program wordt dat object als hoofdobject genomen. Bij updates aan onderliggende objecten wordt het "direct gelinkte" Course of Program als hoofdobject van een update genomen. Bijvoorbeeld: als een ProgramOffering aangepast wordt, is het bijbehorende Program het hoofdobject.

Bij een update van een Program, worden geen updates aan de gekoppelde programs of gerelateerde Courses doorgegeven. Bij een update aan een Course worden geen updates aan de gerelateerde Programs doorgegeven. Voor deze situaties moeten aparte updates aangeboden worden.

Alleen voor Programs en Courses welke gekoppeld zijn aan een EducationSpecification kunnen updates aangeboden worden.

## API endpoints

Het doorgeven van updates verloopt via de onderstaande endpoints:

- POST /job/education-specifications/{educationSpecificationId}
- POST /job/programs/{programId}
- POST /job/courses/{courseId}

Het gebruikte endpoint is afhankelijk van het type hoofdobject.

Response voor bovenstaande endpoints is:

200 OK  
Content-Type: application/json

```
{"token": TOKEN}
```

Het token is een UUID string.

GET /status/{token}

200 OK  
Content-Type: application/json

```
{"status": STATUS}
```

Als het token niet herkend wordt:

404 Not Found  
Content-Type: application/json

```
{"status": "unknown"}
```

Status is één van:

- unknown
- pending
- in-progress
- time-out

- error
- done

De "error" status heeft extra properties:

- message

De foutmelding ter informatie zoals verkregen tijdens verwerking.

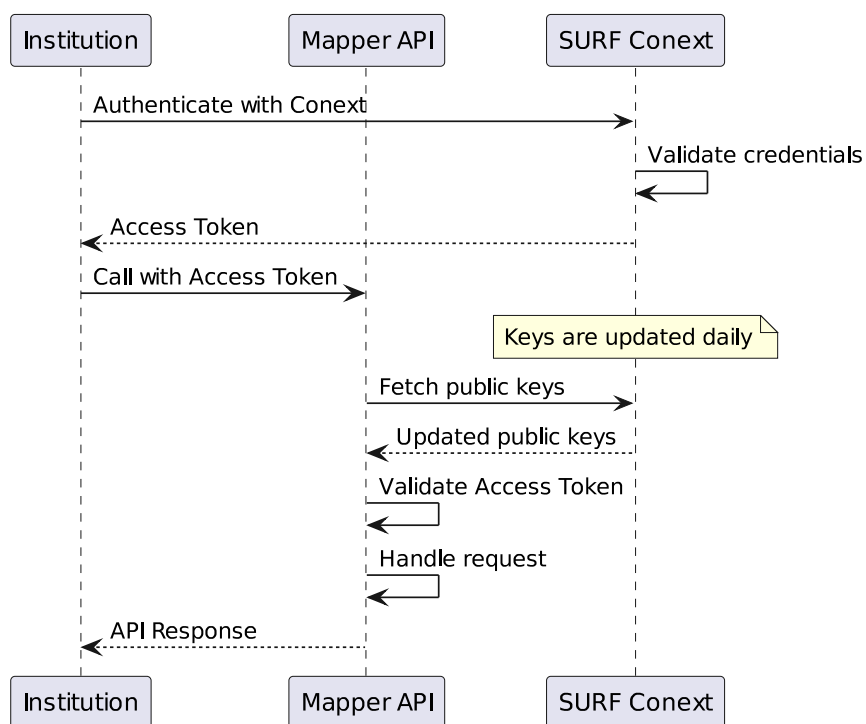
- phase

De fase waarin de fout voorgekomen is.

- fetching-ooapi
- resolving
- fetching-rio (misschien later geïmplementeerd)
- merging (misschien later geïmplementeerd)
- updating

## Authenticatie dmV OAuth2 client credentials

Authenticatie verloopt via OAuth2 Client Credentials gefaciliteerd door SURF Conext.



Figuur 1: OAuth2 Client Credentials flow met SURF Conext