

OOAPI voor roostering

Implementatie Eveoh MyTimeTable op basis van OOAPl bij de UL

Opdracht

- Implementeer MyTimeTable voor het tonen van
 - Persoonlijk rooster (student, medewerker)
 - Opleidings- en vakrooster
- UL gebruikt geen roostersysteem, de informatie staat in uSis
- Vereiste vanuit UL: gebruik de OOAPI
 - Nieuw voor Eveoh
 - Eerste koppeling met uSis -> veel spraakverwarring

Welke services zijn er geïmplementeerd

- faculty
 - Misbruikt, bevat de opleidingen
 - uSis heeft en MyTimeTable vraagt een andere gelaagdheid dan OOAPI ondersteund
 - type veld gebruikt om aan te geven dat het om studieprogramma's gaat
- educational-programmes
 - Bevat de studiedeellijsten, die als opleiding beschouwd worden
 - Waarom geen course-groups ...
 - er is geen relatie tussen educational-programme en course-group, dus je moet ergens afwijken van de standaard

Welke services zijn er geïmplementeerd

- course-offerings
 - Overzicht van het vakaanbod
 - OOAPI is vanuit een beperkte invalshoek ontworpen en houdt geen rekening met andere behoeften dan die van de student of een andere view op de data
 - Uitbreiding met extra kenmerken voor filtering: mtt_lecturer, mtt_educationalProgramme
- schedules
 - Het rooster zelf
 - OOAPI is vanuit een beperkte invalshoek ontworpen en houdt geen rekening met andere behoeften dan die van de student of een andere view op de data
 - Uitbreiding met extra kenmerken voor filtering: mtt_lecturer, mtt_educationalProgramme

Welke services zijn er geïmplementeerd

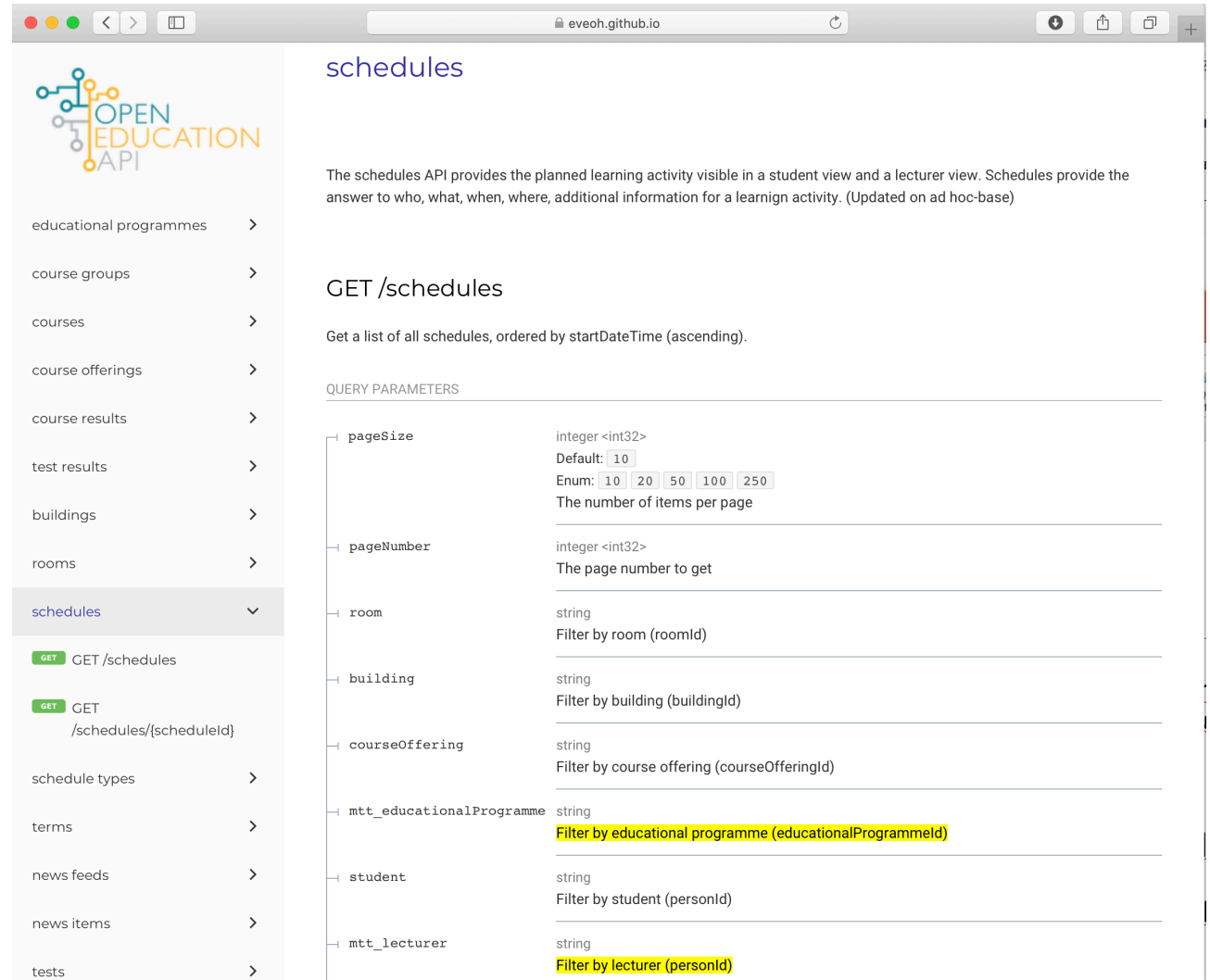
- Maar er bleken er meer nodig ...
 - MyTimeTable is puur en houdt geen data vast (bijv. referentiegegevens)
 - OOAPI ondersteunt niet echt het uitwisselen van referentiedata
 - Metadata is hiervoor geen oplossing
- mtt-terms
 - Ophalen van de aanbodperiodes en hun start- en einddatum tbv selecties
- mtt-schedule-types
 - De activiteiten, gegroepeerd naar types inclusief kleurindicaties

Andere uitdagingen

- Meertaligheid
 - Huh dat zit er niet in ... we voegen wat extra attributen toe
 - `mtt_name/nl` en `mtt_name/en` -> laten we het dan maar direct uitbreidbaar maken
 - `mtt_remark/nl` en `mtt_remark/en`
 - We vullen de name en remark op basis van de languageHeader
- Paginering – mooi concept maar moeten we dat wel doen
 - MyTimeTable is puur en houdt geen data vast
 - Bijv. zoeken naar studentrooster haalt 30.000 studenten op (elke keer) en gaat dan de zoek van MyTimeTable in. Dat zijn best veel calls.

Resultaat na 3 maanden

- Opgeleverd voor gebruikersacceptatie
- Implementatie gedocumenteerd
 - Afwijkingen van de standaard goed aangegeven
- 4 OOAPI services geïmplementeerd die gebruik maken van 13 verschillende uSis services
 - Mapping en routing via inQdo integratieplatform
- 2 eigen 'OOAPI-services' toegevoegd



The screenshot shows a web browser window displaying the documentation for the 'schedules' API endpoint. The browser's address bar shows 'eveoh.github.io'. On the left, there is a sidebar with a logo for 'OPEN EDUCATION API' and a list of API endpoints: 'educational programmes', 'course groups', 'courses', 'course offerings', 'course results', 'test results', 'buildings', 'rooms', 'schedules' (highlighted), 'schedule types', 'terms', 'news feeds', 'news items', and 'tests'. The main content area is titled 'schedules' and contains a description: 'The schedules API provides the planned learning activity visible in a student view and a lecturer view. Schedules provide the answer to who, what, when, where, additional information for a learnign activity. (Updated on ad hoc-base)'. Below this, the endpoint 'GET /schedules' is listed with the description 'Get a list of all schedules, ordered by startDateTime (ascending)'. A section titled 'QUERY PARAMETERS' lists several filters: 'pageSize' (integer <int32>, default 10, enum 10, 20, 50, 100, 250), 'pageNumber' (integer <int32>), 'room' (string, filter by room (roomId)), 'building' (string, filter by building (buildingId)), 'courseOffering' (string, filter by course offering (courseOfferingId)), 'mtt_educationalProgramme' (string, filter by educational programme (educationalProgrammeId)), 'student' (string, filter by student (personId)), and 'mtt_lecturer' (string, filter by lecturer (personId)).

educational programmes >

course groups >

courses >

course offerings >

course results >

test results >

buildings >

rooms >

schedules v

GET GET /schedules

GET GET /schedules/{scheduleId}

schedule types >

terms >

news feeds >

news items >

tests >

schedules

The schedules API provides the planned learning activity visible in a student view and a lecturer view. Schedules provide the answer to who, what, when, where, additional information for a learnign activity. (Updated on ad hoc-base)

GET /schedules

Get a list of all schedules, ordered by startDateTime (ascending).

QUERY PARAMETERS

pageSize	integer <int32> Default: 10 Enum: 10 20 50 100 250 The number of items per page
pageNumber	integer <int32> The page number to get
room	string Filter by room (roomId)
building	string Filter by building (buildingId)
courseOffering	string Filter by course offering (courseOfferingId)
mtt_educationalProgramme	string Filter by educational programme (educationalProgrammeId)
student	string Filter by student (personId)
mtt_lecturer	string Filter by lecturer (personId)

Vragen / opmerkingen