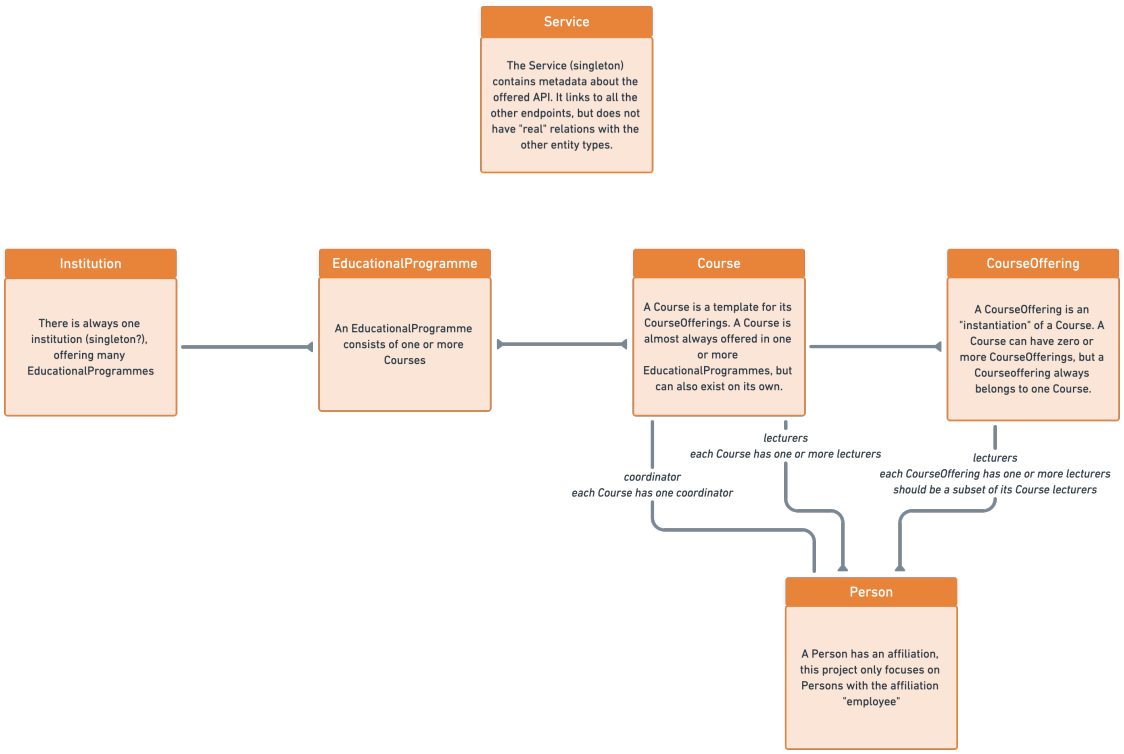
Datamodel demodata

Table of contents

- Table of contents
- Entity-entity relations
- Service
- Institution
- EducationalProgramme
- Course
- CourseOffering
- Person

Entity-entity relations



Service

Attribute (OOAPI v3)	Description (OOAPI v3)	Notes	Depends on
owner	Name of the service owner	• Should be the same as Institution/name	• Institution /name

logo	URL of the service logo (image dimensions should be 300x300 pixels)	• Would be great if we could randomly generate a logo, but obviously a nice to have. • For now just link to a static image somewhere.	• Institution /name
specification	URL of the API specification (YAML or JSON, compliant with Open API Specification v3)	• The API should also host the Open API spec so that we can link to it.	
documentation	URL of the API documentation, including general terms and privacy statement	• Link to the Github repo for this project.	
courseLevels	Set of course levels for the institution	• Enum: "Bachelor" or "Master. • All EducationalProgrammes and Courses should have one of these levels.	
roomTypes	Set of room types for the institution	• We won't use these, so we will just use the list from the example in the API documentation: • ["General purpose", "Lecture hall", "PC lab"]	
_links to endpoints	Links to the service (collection) endpoints	• A list of all the endpoints we will implement, so " /persons", "/educational-programmes", etc.	

Institution

Attribute (OOAPI v3)	Description (OOAPI v3)	Notes	Depends on	Notes for the OOAPI workgroup
Institution Id	id of the institution	• Random id		
brin	Brincode of the institution	• Brincodes zijn strings met het volgende formaat: #"\d{2}[A-Z]{2}"		
name	Name of the institution	• Name of the institition, a string that we will have to generate consistently. • Make it dependent on a hidden "City" atttribute (that we will also use in Institution /address). • Then generate a string, either: "Hogeschool van <City Name>" or "Universiteit <City Name>"	• Hidden attribute Institution/city • Maybe a hidden attribute for whether this is an hbo or a university (such an attribute could be useful elsewhere)	

description	Any general description of the institution should clearly mention the type of higher education institution, especially in the case of a binary system. In Dutch; universiteit (university) or hogeschool (university of applied sciences).	• Lorem ipsum text for now.		
academic Calendar	links to academic calenders per year	• For now let's just link to some random url.		
address	the full street address including house number and street name of main building	• Full address of the main building, let's make it so that it is at least in the correct city if you look at the Institute name.	• Hidden attribute Institution/city	
logo	uri to logo of the institution	• Should be the same as Service/logo		• Not sure why this attribute occurs twice in the spec. Discuss.
_links to educational-programmes	Link to the list of educational programmes at this institution	• Should link to /educational-programmes.		

EducationalProgramme

Attribute (OOAPI v3)	Description (OOAPI v3)	Notes	Depends on	Notes for the OOAPI workgroup
educationalProgrammeId	Unique id for this educational programme	• Random id		
name	The name of this educational programme	• Name should be picked from one of a set of lists. • Which list to use depends on the fieldOfStudy	• EducationalProgramme/fieldsOfStudy	
description	The description of this educational programme	• Lorem ipsum for now		

termStartDate	The day on which this educational programme starts, RFC3339 (full-date)	• Date on which this programme was offered for the first time. • Should be first monday of September or first monday of February of some year.		
termEndDate	The day on which this educational programme ends, RFC3339 (full-date)	• Date after which this programme will no longer be offered. • It is optional, for most programmes it is not yet known when (or if) they will ever cease to be offered. • If it exists, it should be (at least some years) after the termStartDate. • If it exists, should be last day of August or last day of January.	• EducationalProgramme /termStartDate	
ects	The number of EC's that can be archieved in this educational plan	• A number larger than 0. • To be realistic should lie between 60 and 240 in jumps of 30.	• To be really realistic, should be dependent on a hidden attribute which describes whether this is a Bachelor or Master programme. • Master programmes are usually shorter (= less ects) than Bachelor programmes.	• This is modelled as an integer, but should be a float. • Is called ects, but should be called ec.
mainLanguage	The main language in which the courses within this education plan are given, RFC3066	• Randomly choose between NL-nl and EN-gb	• If we want to be fancy we look at a the hidden attribute about whether this is an hbo or university and weigh the choice by that.	
qualificationAwarded	Type of qualificaton that can be obtained on finishing the programme	• We will generate a string based on the dependencies	• Depends on hidden attribute for Bachelor or Master. • Depends on EducationalProgramme /fieldsOfStudy.	

lenghtOfProgramme	The lenght of the programme specified in months	• Calculate the length based on the ects. • The norm is 60 ec = a year	• EducationalProgramme/ects	
levelOfQualification	Level of qualification according to the National Qualification Framework and the European Qualifications Framework	• Lookup list based on whether it is a Bachelor or Master	• Hidden attribute for Bachelor or Master.	

fieldsOfStudy	Field(s) of study (e.g. ISCED-F) http://uis.unesco.org/sites/default/files/documents/isced-fields-of-education-and-training-2013-en.pdf	• Enum, should be one of the following:
		  • Economie
		  • Gezondheidszorg
		  • Gedrag & Maatschappij
		  • Landbouw & Natuurlijke omgeving
		  • Natuur
		  • Onderwijs
		  • Taal & Cultuur
		  • Recht
		  • Techniek
		  • Sectoroverstijgend		

profileOfProgramme	brief description of the main focus of the programme.	• Lorem ipsum		
programmeLearningOutcomes	List the learning outcomes at programme level. It is advisable to limit the number of learning outcomes to approximately 20. It is also advisable to make sure that the programme learning outcomes in the course catalogue correspond with those on the Diploma Supplement.	• For now let's just do a Lorem ipsum.		

modeOfStudy	Indicate whether the programme is e.g. full-time, part-time, dual, e-learning.	• weighted enum:
		  • full-time
		  • part-time
		  • dual
		  • e-learning
		• Most will be full-time.		

_links to courses	Link to the list of courses that are part of this programme			

Course

Attribute (OOAPI v3)	Description (OOAPI v3)	Notes	Depends on	Notes for OOAPI workgroup

courseId	Unique id of this course	• Random id		
name	The name of this course (ECTS-Title)	• A name to be generated from (several) lists of names • Should depend on its (main) EducationalProgramme fieldsOfStudy sow we can generate approriate names.	• EducationalProgramme /fieldsOfStudy	
abbreviation	The abbreviation or internal code used to identify this course (ECTS-code)	• Abbreviation generated from the name. • Should be unique for this course.	• Course/name	
ects	The number of EC's that is earned when the course is completed successfully (ECTS-ects)	• Should be an integer somewhere between 2.5 and 60 in steps of 2.5. Lower values are more common. Most common should be values between 5 and 10.		• Should be a float.
description	The description of this course (ECTS-description)	• Lorem ipsum		
learningOutcomes	Statements that describe the knowledge or skills students should acquire by the end of a particular course (ECTS-learningoutcome)	• For now lorem ipsum		
goals	The description of the main goal of this course	• For now lorem ipsum		
requirements	The requirements needed to enter this course (ECTS-prerequisites)	• Would be great if we could somehow use information from already generated courses to generate this information.	• Course/level • Other already generated Courses?	
level	The level of this course (see service metadata for possible values) (ECTS-year of study if applicable)	• Enum whose possible values depend on service metadata	• Service /courseLevels	
format	The format in which this course is given (ECTS-format) (mutlivalue)	• An array (=set) of strings, we should be able to give weighted options.		
modeOfDelivery	information on the mode of delivery of the component, e.g. via e-learning, face-to-face, etc. (ECTS-Mode of delivery)	• WIll be some form of weighted enum	• Possibly dependent on EducationalProgramme /modeOfStudy	
mainLanguage	The main language in which this course is given, RFC3066 (ECTS-language of instriction)	• Enum, either nl-NL or en-GB		
enrollment	The extra information that is provided for enrollment	• For now Lorem Ipsum		
resources	An overview of the literature and other resources that is used in this course (ECTS-recommended reading and other sources)	• For now Lorem Ipsum		

exams	A description of the way exams for this course are taken (ECTS-assessment method and criteria)	• For now Lorem Ipsum		
schedule	A description of the timeframes, when this course takes place (ECTS-Semester /trimester when the component is delivered)	• Some sort of generated string / enum		
link	The url containing the address of the website with more information about the course	• Some example url		
_links to lecturers	Links to the lecturers (persons) of this course (ECTS-lecturers)			
_links to course offerings	Link to the list of courseOfferings for this course			

CourseOffering

Attribute (OOAPI v3)	Description (OOAPI v3)	Notes	Depends on	Notes for the OOAPI workgroup
courseOfferingId	Unique id of this course offering	• Random id		
course	The course object of which this courseOffering is an instance		• Course (entity)	
maxNumberStudents	The maximum number of students allowd to enroll for this course offering	• Some integer		
currentNumberStudents	The number of students that have already enrolled for this course offering	• Some integer, should be lower than maxNumberStudents	• CourseOffering /maxNumberStudents	
academicYear	The year in which this courseOffering takes place	• Should be a string of the form #"\d{4}-\d{4}". • Second year should be first year +1. • Might be useful to model this with a hidden attribute in some way.		• Add regex?
period	The period in which this courseOffering is offered	• Some form of generated string.	• Possibly on CourseOffering /academicYear • Possibly on Course /schedule	
_links to lecturers	Links to the lecturers (persons) of this course (ECTS-lecturers)			

Person

Attribute (OOAPI v3)	Description (OOAPI v3)	Notes	Depends on
personId	Unique id of this person	• Random id	
givenName	The first name of this person	• We will select a random name from a list, based on gender.	• Person/gender
surname	The family name of this person	• Random name from list.	
surnamePrefix	The prefix of the family name of this person	• Random infix for a list.	
displayName	The name of this person which will be displayed	• String combination	• Person/givenName • Person/surnamePrefix • Person/surname • Person/title
dateOfBirth	The date of birth of this person, RFC3339 (full-date)	• Randomly generated valid date. Would be good to make this dependent on affiliations for example: younger persons are more likely to be students and vice versa. • Since we limit ourselves to employees in this example, would be good to have some allowable range for the dates.	• Possibly on affiliations.
affiliations	The affiliations of how this person is associated with the organization	• Should be an enum, but for now we will only generate "employees"	
mail	The e-mailaddress of this person	• String combination • Should be **unique** for every person	• Person/givenName • Person/surnamePrefix • Person/surname • Institution/name (or a hidden attribute Institution/domain)
telephoneNumber	The telephone number at the office of this person	• Some string combination ("030-235 3545" for example) • Could be unique?	
mobileNumber	The mobile number of this person	• Same as above, but should be **unique** for every person.	
photoSocial	The url of the informal picture of this person	• For now a link to a default user avatar. • Maybe in the future we could use something like this? https://thispersondoesnotexist.com/ There are archives available with lots of pictures of non-existent persons.	• In the future might depend on Person /dateOfBirth and Person /gender
photoOfficial	The url of the official picture of this person	See photoSocial	

gender	The first letter of this person's gender	• Weighted enum:
		  • M
		  • F
		  • X
		  • U	

title	A title to be used for this person	• Some weighted enum. • Would like the option to not have this attribute on some entities based on some chance.	
office	The name of the office where this person is located	• Some random string generated from a regex. • Could be something like: #"[A-Z]{3} \d.\d{2}"	
s to courses	Link to the list of courses for this person		