1 Interface specification

This document outlines the specifications of the back-end application. The back-end application has a RESTful interface with which the front-end communicates with.

The data format used will be JSON.

Complete overview of back-end endpoints. P stands for parameter	
Uri path without source	Description
sp/p/curricula	a list of curricula
sp/p/qualification	a list of qualifications
semester/p/modules	a list of modules
semester/p/qualifications	a list of qualifications
curriculum/:curriculum_id/semesters	a list of semesters
module/p/lg	a list of learning goals

2 Sprint implementation

For the first sprint endpoint /curriculum/:curriculum_id/semesters will be implemented.// Example of this would be: curriculum/1/semesters

semesters of curriculum	
URL	/curriculum/:curriculum_id/semesters
Method	GET
Returns	json array of semester objects which in turn are filled with corresponding modules.
Returns	200 OK
	404 Not found

3 Endpoint definitions

This section describes the urls more specifically with functionality, the JSON format and HTTP codes. The dataobjects are described in json_objects.pdf in this repository

3.1 Curricula of student program

curricula of student program	
URL	{source}/sp/:id/curricula
Method	GET
Returns	json array of semester objects which in turn are filled with corresponding modules.
Returns	200 OK
	404 Niet gevonden

qualifications of student program	
URL	{source}sp/p/qualification
Method	GET
Returns	200 OK
	404 Not found

modules of a semester	
URL	{source}semester/p/modules
Method	GET
Returns	200 OK
	404 Niet found

qualifications after semester	
URL	{source}semester/p/qualifications
Method	GET
Returns	200 OK
	404 Niet found

learning goals of a module	
URL	{source}module/p/lg
Method	GET
Returns	200 OK
	404 Niet found