***curriculumService***

**Project Outline**

**Main.Java:**

- com.revature.curriculumservice

\* CurriculumServiceApplication.java

Application entry point, configures project as spring boot project and discovery client.

- com.revature.curriculumservice.config

\* SwaggerConfig.java

Configure and enable Swagger

-com.revature.curriculumservice.domain – POJOS

\*Activatable.java

Interface that marks class as activatable and provide setter

\* Curriculum.java

Object class that holds Curriculum which contains the name and set of skills that a batch will learn.

-com.revature.curriculumservice.domain.dao

\* ActivatableObjectRepository.java

Fancy file by August - no actual content, but anything activatable should extend this interface, such as a room or a trainer.

      \* BaseRepository.java

Anything not activatable should extend this interface.

       \* CurriculumRepository.java

extends ActivatableObjectRepository

- com.revature.curriculumservice.domain.dto

\*\*DTO’s reflect the POJO’s.  They are the in-between.  In our REST controllers, we accept respective DTO formats  coming from our javascript files.  The major differences between the POJO’s and the DTO’s is that DTO’s do not contain defined constructors, and they are not mapped to database tables.

\*CurriculumDTO.java

\*ResponseErrorDTO.java

This DTO has no mapped POJO, it has only one field defined as errorMessage

-com.revature.curriculumservice.service

\*ActivatableObjectDaoService.java

Service layer for ActivatableObjectRepository to iactivate activatable object instead of deleting them. Extends DaoService.

\*CurriculumService.java

Service layer for CurriculumRepository that extends ActivatableObjectService, services specific to curriculum will be defined here.

\*DaoService.java

Basic service layer for BaseRepository, provides generic services to all other services.

- com.revature.curriculumservice.web

            \* CurriculumCtrl.java - \*

REST Ctrl’s are for translating javascript objects via DTO to their corresponding POJO’s (or vice versa), and make calls to the db to update, delete, create new, or retrieve records.\

**Main.Resource:**

\*application.yml

Configuration file that defines server port, database configuration, pivotal cloud configuration.

\*log4j2.xml

Configuration file for log4j

**Test.Java:**

-com.revature.curriculumservice

\*AssignforceCurriculumMsApplicationTests.java

Default Spring Boot Test class to hold test method

-com.revaure.curriculumservice.utils

\*JsonMaker.java

Class that provides that holds Gson object and other utilities

-com.revature.curriculumservice.web

\*CurriculumCtrlTest

Class that holds tests for Curriculum Controller.

**Data Model and Controller**

**Curriculum.java**

**Implements Activatable**

**Properties**

* int **currId**
* String **name**
* List<Integer> **skills**
* Boolean **active**
* Boolean **core** – true if it is a core curriculum, false if it is a focus

**Behavior**

* **public** Curriculum()

default empty constructor

* **public** Curriculum(**int** currId, String name, List<Integer> skills, **boolean** core)

constructor sets all but active

* **public** Getter and Setter

getters and setters of all properties

* **public** String toString()

return all info as String

**CurriculumCtrl.java**

**Behavior**

* **public** Object createCurriculum( @RequestBody CurriculumDTO in )

REST POST method for creat curriculum in DB

* **public** Object retrieveCurriculum( @PathVariable(**"id"**) **int** ID )

REST GET method for retrieving particular curriculum using currId

* **public** Object updateCurriculum( @RequestBody CurriculumDTO in )

REST PUT method for updating existing curriculum

* **public** Object deleteCurriculum( @PathVariable(**"id"**) **int** ID )

REST DELETE method

* **public** Object retrieveAllCurricula()

REST GET method for getting all curricula

* **public** Object retrieveAllActiveCurricula()

REST GET method for getting all active curricula

* **public** Object retrieveAllCore()

REST GET method for getting all core

* **public** Object retrieveAllActiveCore()

REST GET method for getting all active core

* **public** Object retrieveAllFocus()

REST GET method for getting all focus

* **public** Object retrieveAllActiveFocus()

REST GET method for getting all active focus