***skillService***

**I. Project Outline**

**Main.Java:**

- com.revature.skillservice

\* AssignforceSkillMsApplication.java

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

-com.revature.skillservice.domain – POJOS

\*Activatable.java

Interface that marks class as activatable and provide setter

\* Skill.java

Object class that holds Skill which contains the name and id of skill.

-com.revature.skillservice.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.

       \* SkillDao.java

extends ActivatableObjectRepository

- com.revature.skillservice.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.

\*ResponseErrorDTO.java

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

-com.revature.skillservice.service

\*ActivatableObjectDaoService.java

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

\*SkillService.java

Service layer for SkillDao that extends ActivatableObjectService, services specific to skill will be defined here.

\*DaoService.java

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

- com.revature.skillservice.web

            \* SkillCtrl.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.

**Test.Java:**

-com.example.demo

\*AssignforceSkillMsApplicationTests.java

Default Spring Boot Test class to hold test method

-com.example.demo.util

\*JsonMaker.java

Class that provides that holds Gson object and other utilities

-com.example.demo.web

\*SkillCtrlTest

Class that holds tests for Skill Controller

**II. Data Model and Controller**

**Skill.java**

**Implements Activatable**

**Properties**

* int skillId
* String name
* Boolean active

**Behavior**

* **public** Skill()

default empty constructor

* **public** Skill(**int** skillId, String name)

constructor sets all but active

* **public** Getter and Setter

getters and setters of all properties

* **public** String toString()

return all info as String

**SkillCtrl.java**

**Behavior**

* **public** Object createSkill( @RequestBody Skill in )

REST POST method for creat skill in DB

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

REST GET method for retrieving particular skill using skillId

* **public** Object updateSkill( @RequestBody Skill in )

REST PUT method for updating existing skill

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

REST DELETE method

* **public** Object retrieveAllSkills()

REST GET method for getting all skills

* **public** Object retrieveSkillsByIds(@RequestBody List<Integer> in)

REST GET method for getting a list of skills using a list of skillId