Code Review Project Backend Code



In dit document kunnen jullie enkele opmerkingen vinden over de backend code van jullie team. Wegens de hoeveelheid teams die we reviewen is de commentaar misschien soms wat kort of zonder voldoende context. Als jullie ergens vragen bij hebben of gewoon nog meer willen weten, contacteer ons in dat geval. We kijken met veel plezier nog even langer mee en beantwoorden eventuele bezorgdheden.

- Via e-mail: tom.cools@infosupport.com
- Via teams van AP Hogeschool: Tom Cools

Team: Team 11

Overzicht

Volgende scores zijn een indicatie van de algemene kwaliteit. De scoring gaat van A (Zeer Goed) tot F (Zeer slecht).

- Code Conventions: C
- Database Interaction: A
- Object Oriented Programming: A
- Project Structure: A

Totale Score: A/B

Feedback voor studenten

Code Conventions

Quickfire comments:

- Namen van packages zijn standaard lowercase. Jullie 'Repositories' package volgen die richtlijn niet.
- Code in commentaar is enkel verwarrend en staat in de weg! Gooi het weg! Als je het toch nog ergens nodig hebt: use Git.

REST conventions

@PostMapping("/add")

POST impliceert al dat je iets wilt gaan toevoegen.

Als je commentaar toevoegt aan het project, doe het dan volledig. Wat betekenen volgende parameters?

/**

- * @param name
- * @param description
- * @param parent

```
*/
public Capability(String name, String description, Capability parent) throws
IllegalArgumentException {
```

```
System.out.println("something went wrong! oh no!");
ex.printStackTrace();
```

Gebruik nooit ex.printStackTrace()! 2 redenen:

- Je hebt geen controle over waar dit zal weggeschreven worden (System.out is default);
- Er staat soms te veel informatie in waar je deze eigenlijk niet nodig hebt.

Gebruik in plaats daarvan een Logger. Deze kan je aanmaken bovenaan je klasse: manueel of met Lombok

```
@Slf4j //lombok
public class ExampleClass() {
    // manueel
    private static final org.slf4j.Logger log =
    org.slf4j.LoggerFactory.getLogger(LogExample.class);

    //Code excluded, somewhere in the class
    log.debug("Error occured",ex);
}
```

Database Interactions

Kunnen we weinig op aanmerken voor nu. Er staat nog maar 1 repository. Mongo is wel interessante keuze (no judgement).

Object Oriented Programming

We zien geen afscheiding tussen jullie Database model (documents) en de REST API (DTOs). Misschien hebben jullie dat helemaal niet nodig. We merken echter op projecten dat op langere termijn je zo soms wat in de problemen komt. Iets dat je in de REST API wel nodig hebt, maar niet in de database en vice-versa. Hou dit in de gaten.

Project Structure

Duidelijk, er staat nog niet heel veel.

Extra notes

Cool dat jullie voor MongoDB gekozen hebben en ook al *flapdoodle* gevonden hadden. Wil volgende keer graag wat werkende testen zien.

Het lijkt erop alsof er nog heel wat werk aan de winkel is in de backend. We kijken uit naar de volgende code review wanneer er iets meer code in de repo aanwezig is.