

VAKOVERSCHIJDEND PROJECT

UITBREIDINGEN

Thomas De Pontieu, Billie Devolder, Jarre Knockaert, Sam Persoon, Jorg Van Renterghem, Tomas Van Roose, Freek Verschelden
16 maart 2017

1 Vloten samenvoegen

De eerste uitbreiding die we zullen bespreken is het samenvloegen van vloten. Indien 2 vestigingen van een bedrijf fuseren en de vloten moeten worden samengevoegd, zou het handig zijn dit met een simpele druk op een knop te kunnen doen. In principe moet er voor deze uitbreiding niets worden veranderd in de backend. De frontend kan alle voertuigen van een bepaalde vloot opvragen en dan via een PUT request de vloot veranderen van alle voertuigen in de vloot.

De aanpassingen staan hier zeker in verhouding tot de uitbreiding. In de backend en database moet er namelijk niets aangepast worden.

2 Ondersteunen van boten

Misschien wil Solvas, in een verre toekomst, zich ook gaan bezighouden met het verzekeren van boten. Hier nemen we aan dat het verzekeren van boten gebeurt op een gelijkaardige manier.

In het domeinmodel zou er een nieuwe klasse, **WaterCraft**, moeten toegevoegd worden. Deze klasse zou een boot voorstellen en sterk lijken op de huidige klasse **Vehicle**. Deze 2 klassen zouden grote gelijkenissen vertonen dus zou hier best een gemeenschappelijke superklasse voor gemaakt worden. In dit document zal deze superklasse **AbstractVehicle** worden genoemd. De klasse **Fleet** zou dan ook een collectie **AbstractVehicle** objecten moeten hebben in plaats van **Vehicle** objecten.

In de andere lagen zou er een **DataAccessObject**, een **Controller** en een **RestController** moeten bijkomen specifiek voor boten. **WaterCraftDAO** en **WaterCraftController** zouden dan overerven van respectievelijk **DAO** en **AbstractController** zodat enkel de update en create methodes nog moeten worden geïmplementeerd. Aan de klasse **DAOProvider** zou er een methode moeten worden toegevoegd zodat de nieuwe DAO kan worden opgevraagd.

We denken dat de aanpassingen in verhouding staan tot de uitbreiding. Het meeste werk zou kruipen in het doorgeven van de waarden van de velden en deze te zetten in de objecten. Dit is repetitief werk maar niet iets dat gemakkelijk te veralgemenen is. De meeste veranderingen in het domeinmodel zijn eenmalig. Indien er nog een ander soort voertuig zou bijkomen met andere velden, zou er enkel een klasse moeten worden toegevoegd die overerft van **AbstractVehicle**.