

Solvas Fleet - Component Overview

Steven Bastiaens
Benjamin Cousaert
Nils Mak
Niko Strijbol
Karel Vandenbussche
David Vandorpe
Domien Van Steendam

Contents

Presentatielaag	1
Web laag	1
Domeinlaag/servicelaag	2
Persistentie/Repository laag	2
Component Overview	2

Het is belangrijk om te weten dat de specifieke implementatie en technologieën van deze lagenstructuur zeer flexibel zijn. Men kan dus beslissen om bv. NGINX te vervangen door een andere reverse proxy service. De communicatie gebeurt in principe enkel tussen de componenten die boven of onder elkaar staan hieronder.

Presentatielaag

Web UI Deze ontvangt de acties van de gebruiker en vertaalt deze naar HTTP requests naar de backend. Het antwoord van de backend wordt vervolgens vertaald en weergegeven op het scherm.

Web laag

NGINX & Rest Controllers Deze componenten ontvangen de HTTP verzoeken die in het normale geval van de Web UI afkomstig zijn. NGINX is degene die werkelijk de HTTP verzoeken zelf ontvangt en vervolgens doorgeeft in een formaat dat de restcontrollers aanvaarden. De restcontrollers ontvangen de verzoeken in het formaat dat ze geëist hebben. Deze verzoeken worden vervolgens vertaald

naar verzoeken (lees: methodeoproepen) voor de domeinlaag/servicelaag. De antwoordberichten volgen de omgekeerde richting.

Domeinlaag/servicelaag

- Solvasfleet * Deze nog onbestaande component bevat de business logica van de applicatie. Dit houdt onder andere de berekening van de facturen in. Momenteel is er nog niet veel logica nodig voor milestone 1 en is de functionaliteit van deze component geïntegreerd in de restcontrollers. De servicelaag is met andere woorden ingebed in de weblaag.

Persistentie/Repository laag

Data access service Het aanspreekpunt voor deze service wordt gerealiseerd met DAO's. Deze bevatten de methoden om data op te halen uit de persistente opslag. Elke DAO-klasse is verantwoordelijk voor 1 model. We bundelen deze DAO's samen in een DAOContext om de schaalbaarheid te bevorderen. Voor iedere mogelijke opslagbron is er 1 Daocontext vereist.

Momenteel hebben we 1 opslagbron en dus 1 opslagcontext, namelijk de Postgres databank. De DAO's (van deze context) zelf maken momenteel gebruik van Hibernate die dienst doet als ORM. Merk op dat men hier niet gebonden is aan Hibernate.

- Postgres databank driver * We gebruiken als opslag van de data een postgres databank zoals gesuggereerd in de opgave.

Component Overview