Spotitube

By Dani Botland (617442)

# Inhoud

**Table of Contents**

[Inhoud 2](#__RefHeading___Toc105_810035096)

[Inleiding 3](#__RefHeading___Toc107_810035096)

[Package diagram 4](#__RefHeading___Toc109_810035096)

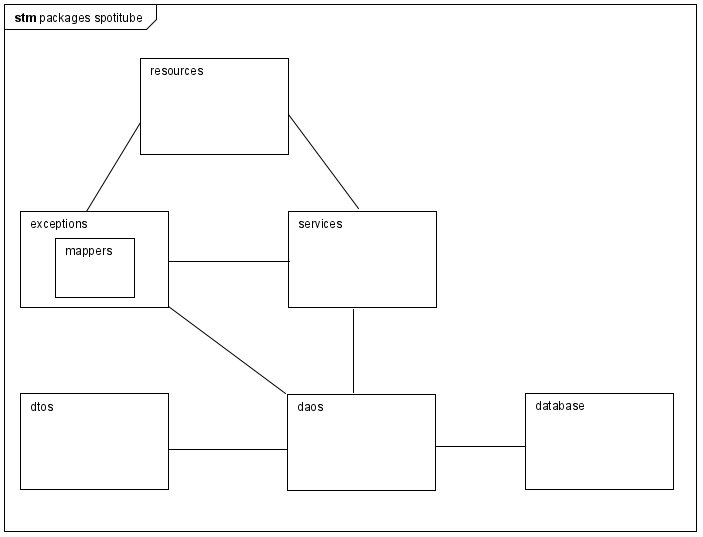
[Deployment diagram 4](#__RefHeading___Toc111_810035096)

[Ontwerpkeuzes 4](#__RefHeading___Toc113_810035096)

# Inleiding

Op de opleiding HBO-ICT hebben wij tijdens het OOSE semester de spotitube opdracht gekregen. Hierbij gaat het om een casus waarbij youtube en spotify de handen in een hebben geslagen en een gezamelijke site hebben opgericht waar je naar muziek kunt luisteren en naar video kunt kijken. In dit document beschrijf ik mijn uitwerking van de casus en geef ik eventuele toelichting waar nodig.

# Package diagram



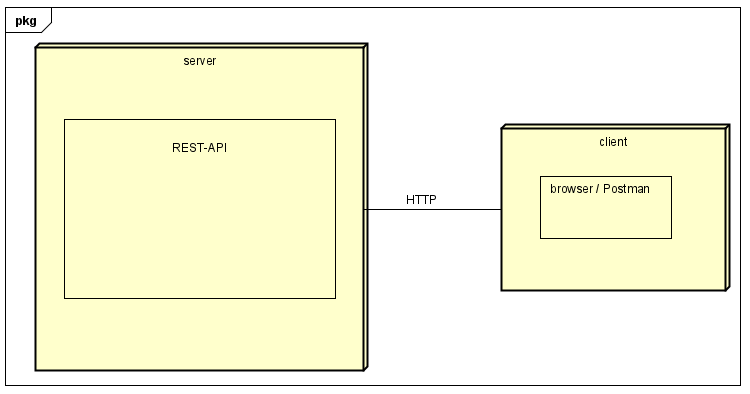
Zoals te zien is in het package diagram hierboven heb ik de applicatie opgedeeld in verschillende packages. Deze packages zijn ook mooi in 3 lagen opgedeeld, je hebt de frontend, de logica en de data laag. De data laag houd zich alleen maar bezig met het bijhouden van data, dit zijn dus de dto’s, dao’s en de database. De dao’s communiceren met de database en zetten de informatie weer om naar dto’s zodat de rest van de applicatie er makkelijker mee kan werken.

De logica laag bestaat uit de services, zij sturen de dao’s aan om de juist data op te halen, in dit geval is er weinig logica nodig en spelen ze deze data dus vooral door naar de resources. Als er iets fout gaat is deze laag er ook veranwoordelijk voor dat de juist exception wordt opgeroepen.

De laatste laag is de voorkant ook wel de resources. Dit is de daadwerkelijke API waarmee de gebruiker in aanraking komt. Alles dat het doet is de request van een gebruiker ontvangen en naar de juiste service doorverwijzen.

Als laatste zijn er nog de exceptions en de exception mappers. Deze hebben als taak om ervoor te zorgen dat niet de hele applicatie crasht wanneer er iets mis gaat. De mappers zorgen er dan voor daat de juist response wordt teruggestuurd.

# Deployment diagram

Het deployment diagram spreekt vrij voor zich naar mijn ogen, de client kan via de browser een HTTP request sturen naar de server die bij mij op apache TomEE draait.

# Ontwerpkeuzes

De grootste keuze die ik heb gemaakt is het gebruik maken van exception mappers. Hiermee voorkom ik dat exceptions die veel worden gegooid voor elke resource methode moeten worden getest. Dit scheelt enorm veeel in de hoeveelheid code in de resources zelf, maar ook voor het aantal test cases datt nodig is om alles goed te kunnen testen.