datum: 25-3-22

2022

Spotitube: opleverdocument

OOSE DEA CASUS SPOTITUBE

MAARTEN VAN DER LEI (STUDENT)

Inhoudsopgave

1 Inleiding

Oplever document voor de course: OOSE, vak: DEA, opdracht: spotitube-api.

In dit document vind je: een package diagram, een deployment diagram en wat code keuzes toegelicht.

Auteur: Maarten van der Lei.

StudentNummer: 658215.

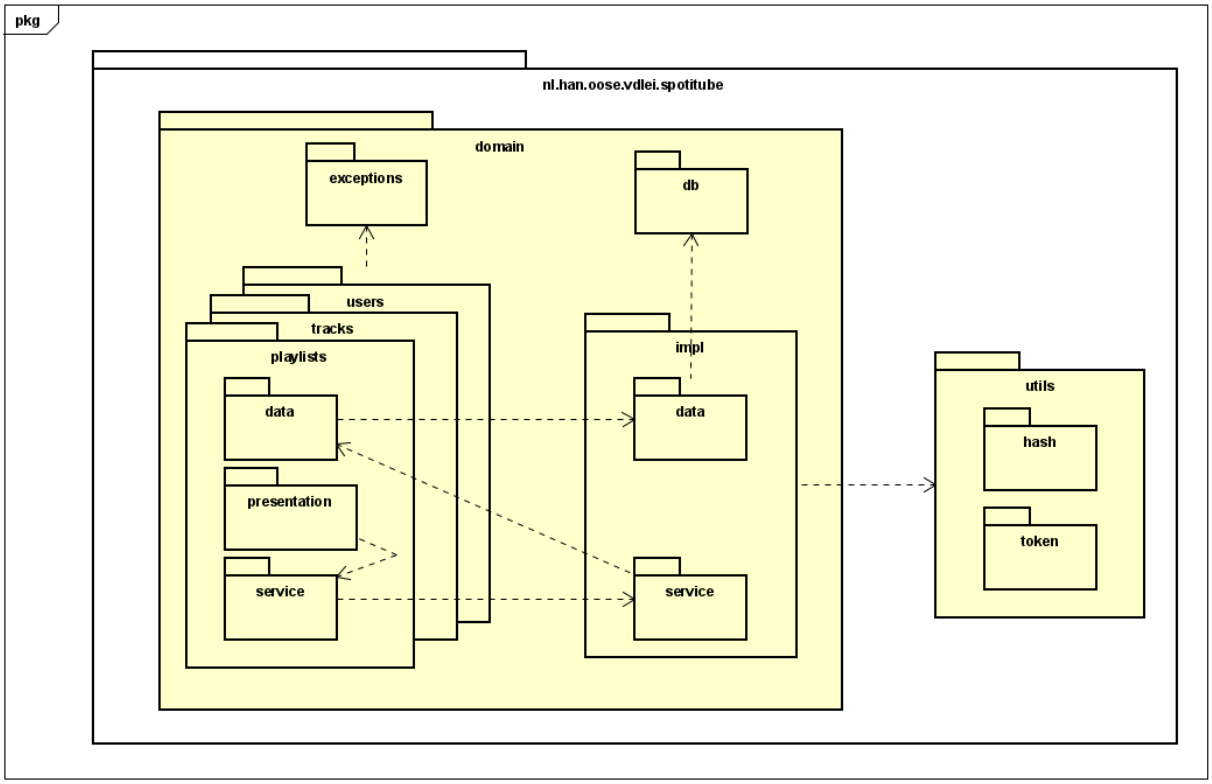
Course: OOSE.

Vak: DEA.

Casus: Spotitube.

OpdrachtGever: Wiebe Rinsma.

2 Package Diagram



2.1 Toelichting:

Het gehele project bevindt zich binnen een package van: **nl.han.oose.vdlei.spotitube**

Binnen het **domain** bevinden zich de verschillende packages voor de verschillende endpoints: **playlists**, **tracks** en **users**.

Elk van deze packages bevatten **data** DAO's als interface, **presentation** voor de endpoints en een **service** laag voor het verbinden van de vorige 2 genoemde lagen.

De package: **Presentation** bevat de inkomende objecten in de vorm van een Request, de uitgaande objecten in de vorm van een Response en de Controllers waar de http verzoeken binnen komen.

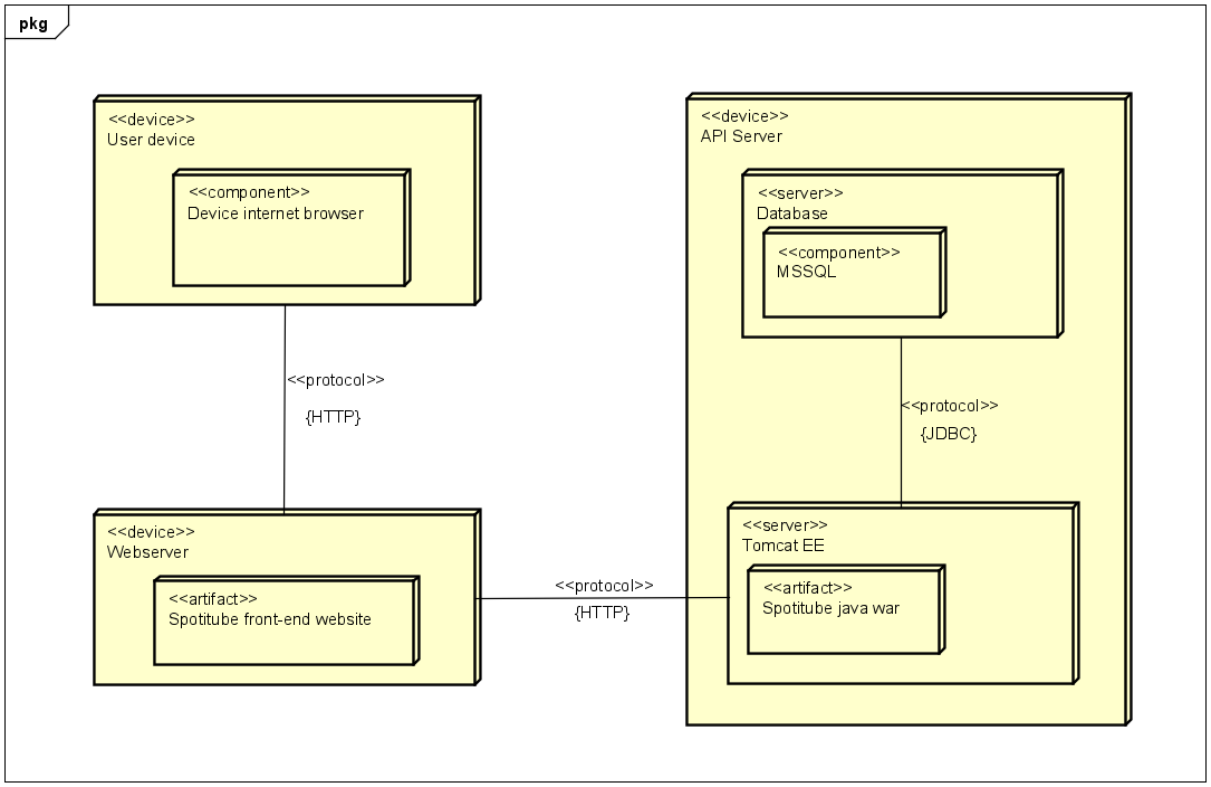
In de package: **impl**, zijn de interfaces van **data** en **service** geimplementeerd, met behulp van DI(Dependency Injection) kunnen we deze gebruiken binnen de **presentation** en **service** laag door de interface aan te roepen.

In de package: **Exceptions**, is een custom exceptie gemaakt voor het missen of sturen van onjuiste token bij protected requests.

In de package: **db** is een class met een static methode waar de dao's een nieuwe database connectie aan kunnen maken door deze static method aan te roepen.

>> dit raakt de requirements:1, 2, 3, 4, 5, 9, 10, 11 en 12.

3 Deployment Diagram



3.1 Toelichting:

Binnen het deployment diagram bevinden zich meerdere devices.

Een device voor de website van spotitube, een device voor de browser van de gebruiker en een device voor de servers van de API.

In de API server bevinden zich 2 servers: een Tomcat EE server met een java artifact, die HTTP requests verwerkt, en een Database server waarmee de Tomcat server verbonden door middel van een JDBC connectie.

Ik heb MSSQL gebruikt omdat het niet verplicht was om een bepaalde database te gebruiken, en ik daarom heb gekozen voor een database waar ik al wat ervaring had met het opzetten van gebruikers rechten en de MSSQL omgeving.

Het gebruik van Tomcat EE was wel verplicht deze opdracht.

Alternatieven zouden geweest zijn:

MySQL in plaats van MSSQL,

De webserver lokaal, op hetzelfde device als de API en DB server, runnen.

>> Dit raakt/lost de volgende requirements op: 1, 3, 6, 7, 14 en 15.

4 Code keuzes:

Binnen mijn code vind je als eerste een bepaalde structuur staan,

Deze keuze heb ik gemaakt om zo overzichtelijk de bestanden te ordenen, ik had ook alles presentation controllers in een controller package kunnen stoppen, en de object(DTO's en Entities) bestanden had ik ook in een losse package kunnen stoppen, hetzelfde geldt voor alle services en dao's(gebeurt eigenlijk ook bij de impl package).

Maar die keuze heb ik niet gemaakt, omdat ik hiermee precies kon zien welke bestanden bij welke type horen.

De keuze voor Tomcat EE en MSSQL zijn H4 uitgelegd.

Ik heb een afzonderlijke package: utils gemaakt, waar 2 packages onder zitten: hash en token.

Deze packages hebben niks te maken met de implementatie of objecten binnen het domein(ik had deze functies simpelweg los kunnen aanroepen vanuit iedere class), en staan daarom om clean-code te maken, in een afzonderlijke package.

In mijn code zijn geen mappers te vinden, ik heb de keuze gemaakt om die niet te maken, omdat ik, terwijl ik bezig was met het maken van de DAO's, het simpelweg niet nodig vond om een mapper te maken voor data, terwijl je zelf in de DAO's in de hand hebt, wat voor data-formaat je terug geeft in de return.

Zo heb ik het mappen van data dus onderverdeeld in de DAO's en services die de DAO's aanroepen om data op te halen.

De keuze om een validateToken() functie te maken is vaag, liever had ik een custom-annotatie gemaakt, maar in verband met de tijd is dat niet gelukt. Daarom word deze functie nu meerdere keren aangeroepen in verschillende controllers.

Ik gebruik een properties bestand die je zelf kan aanvullen met database naam, database user login en wachtwoord, zonder deze 3 kan er geen connectie tot stand worden gebracht.