Service Design and Engineering
Final Project

Radu Loghin - 229368

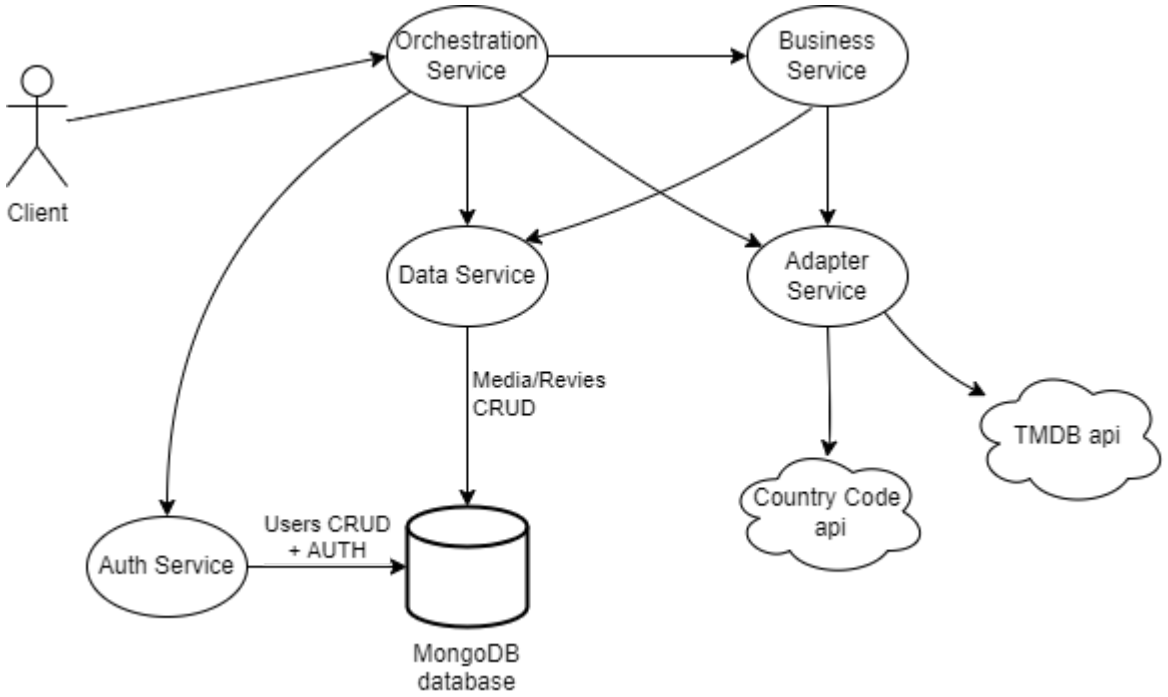
2021/2022

Introduction

The goal of this project was to develop services that provide apis for a platform/website/mobile application which offers users to track, discover and review their favorite movies and tv series. It was inspired by AniList (https://anilist.co/) which is limited only to Anime shows; other similar websites exist such as Letterboxd (https://letterboxd.com/) but in turn they are limited only to movies. This project offers a solution that can track more types of media such as movies, tv-series and anime.

Architecture

The services have been divided according to their functionalities. They communicate through api and use Token authentication. The orchestration service acts as a gateway service and is the only service reachable directly from the clients.



Auth service

- **source code:** https://github.com/radu1690/sde-project-public/tree/main/auth
- **documentation:** https://authservice24.docs.apiary.io/#
- **endpoint:** https://auth-service-sde.herokuapp.com/v1/

This service is responsible for creating and authenticating users to the system. It is connected to a mongoDB database where the users are saved.
Both the register and login operations return an access token and a refresh token. The access token is sent with every future request: the orchestration service is able to decode it and retrieve the necessary information from the user (his userId) and to check if he is authenticated (without calling again the auth service).
If the access token is no longer valid, the server will return an error code 401: in this case the user can send the refresh token to request a new access token.

Adapter service

- **source code:** https://github.com/radu1690/sde-project-public/tree/main/adapter
- **documentation:** https://adapterservice1.docs.apiary.io/#
- **endpoint:** https://adapter-service-sde.herokuapp.com/v1/

This service retrieves and filters external data. In particular it is able to retrieve information about movies and tv-series from TMDB. Both movies and tv-series have different fields on TMDB: the adapter service filters everything and merges all the information in the “media” data type.
Currently the service is able to search media, retrieve popular/recommended/similar media, get details of a specific media, get seasons and episodes, and get streaming availability.

Data service

- **source code:** https://github.com/radu1690/sde-project-public/tree/main/data
- **documentation:** https://dataservice.docs.apiary.io/#
- **endpoint:** https://data-service-sde.herokuapp.com/v1/

This service handles internal data and it is connected to the same MongoDB database of the auth service. There are two types of resources : media and reviews. Media are the elements of a user’s list which can be either a movie or tv-serie. For each media, users can post a review.
The data service does not check the data when performing CRUD operations on media and reviews: this responsibility lies with higher services (business and orchestration).

Business service

- **source code:** https://github.com/radu1690/sde-project-public/tree/main/business
- **documentation:** https://businessservice1.docs.apiary.io/#
- **endpoint:** https://business-service-sde.herokuapp.com/v1/

This service performs logic operations and it makes multiple calls on the adapter service and on the data service. It is responsible for getting all the data for the user page and media page. It also gets all the episodes of a tv series and makes all the checks in order to add correctly a media to a list.

Orchestration service

- **source code:** https://github.com/radu1690/sde-project-public/tree/main/orchestration
- **documentation:** https://orchestrationservice.docs.apiary.io/#
- **endpoint:** https://orchestration-service-sde.herokuapp.com/v1/

This service implements the "process centric service", it processes all requests coming from the user and acts as a gateway to the other services.
Thanks to JWT tokens, the service is able to check if the user is authenticated without calling the authentication service. Both the userId, retrieved from the access token, and the ip of the user, retrieved from the request, are passed to lower services for each request.

Possible extensions

The project is easily extendable: for instance a forum for discussions can be added. It is sufficient to define the data structures in the data service then implement all the actions in the business service or orchestration service.

Source code

The source code is available on: https://github.com/radu1690/sde-project-public.