

Programstruktur

TMDB-api

<https://developers.themoviedb.org/3/getting-started/introduction>

Genom att anropa detta API hämtar vi information på filmer.

Returnerar alltid JSON som vi får parse:a själva.

Möjligheter: Vi kan använda specifika calls här för att läsa in filmer eller använda daily file exports till vår egna databas och göra calls där.

Villkor: Vi måste tydliggöra att vi använder TMDB i appen.

Information vi tagit från TMDB får lagras så länge "det är rimligt".

Firebase

Tillämpning av Google cloud storage där allt också kommer ligga. Ger oss databas där vi kan hantera användare och spara listor/filmer och synka dem mellan användare när de uppdateras.

Villkor: Finns gräns för data/reads/writes men kommer inte kosta något om man inte validerar betalmedel.

Firebase Authentication

Här finns konton och val för sign-in. API för detta rör kontoverifiering och kommer vara centralt för mycket.

Exempel:

https://github.com/FirebaseExtended/flutterfire/tree/master/packages/firebase_auth/firebase_auth/test

Cloud Firestore

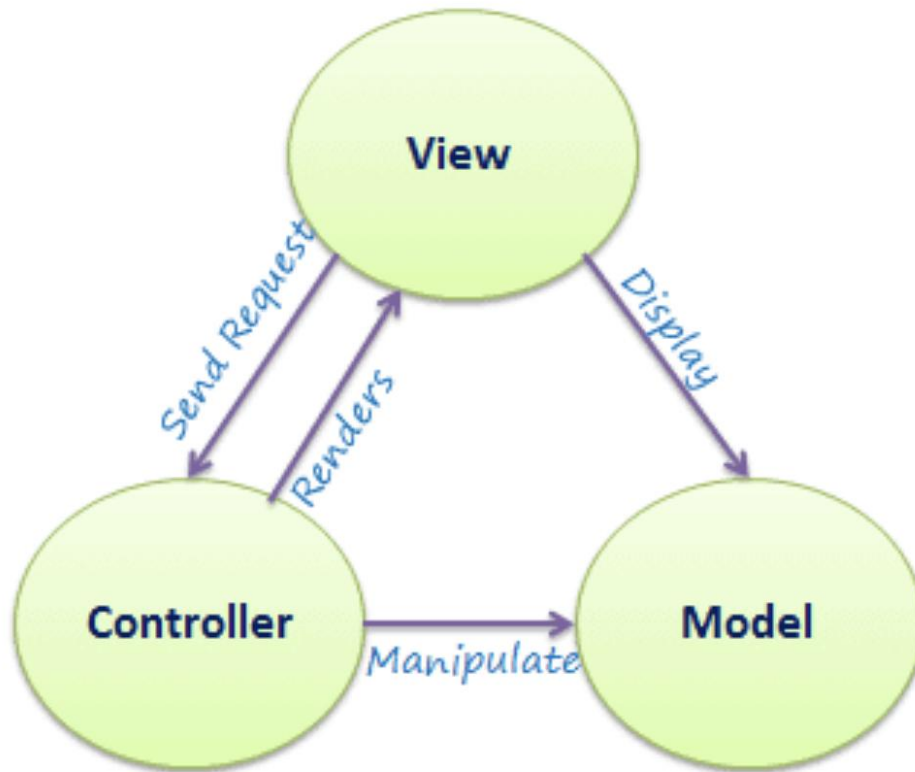
Vår databas där allt sparat ligger. Man kan lyssna på uppdateringar så att vi kan uppdatera en lista eller kö om motparten har agerat med listan m.m. Databasens collections och innehåll uppdateras dynamiskt om man lägger till något från en klient. D.v.s. vi behöver inte definiera strukturen i databasen först om vi inte vill. Man kan skriva regler för vilken klient/användare som får göra read/write-operationer eller constraints för insättningar men för tillfället är allt tillgängligt för alla.

Exempel(lättare att bara googla/kolla youtube:

https://github.com/FirebaseExtended/flutterfire/blob/master/packages/cloud_firestore/cloud_firestore/example/lib/main.dart

BeOs

1. MVC-arkitektur



MVC Architecture

Model

- Här finns applikationslogik.
- Här samlar vi all kod för en enbart logik i applikationen och designen för hur datan bör representeras.
- Data i modellen är inkapslad och bör inte ha tydlig koppling med användargränssnittet (Vyn).

View

- Vyn är den delen av applikationen som visas för användare
- Här lägger vi allt som är grafiskt
- Här finns de flesta klasser som har till syfte att måla/visa något på skärmen.

Controller

- Den delen som tar emot instruktioner från användaren och gör om instruktionerna till något som applikationen kan hantera.
- Controllern känner användarens musklick samt kommandon från tangentbordet och informerar modellen eller vyn vad som ska utföras eller visas
- Här lägger vi alla klasser som har till syfte att utföra någon slags interaktion mellan användare och applikationen eller klasserna som fungerar som mellanhand mellan applikationens vyobjekt och modellobjekt

Resumé: Genom att använda MVC som uppdelning av kod skickas användarens kommandon till kontrollern som har i uppgift att arbeta med modellen för att utföra användarens förfrågningar. Controllern väljer vilken vy som ska visas för användaren och ger den all data från modellen som behövs.