

Dokumentasjon av koden, 23. oktober 2016

Filene “package.json”, “tsconfig.json”, “typings.json” og “systemjs.config.js” bestemmer hvordan applikasjonen konfigureres, blant annet hvor den skal se etter filer og hvordan filene skal kompileres. Når man åpner applikasjonen i en browser vil browseren først se etter “index.html”, som så igjen vil kjøre “app.module.ts”. Dette er hovedfilen til applikasjonen og det er denne som starter opp resten av applikasjonen, ved å blant annet gi tilgangene videre inn til de andre klassene.

I Angular 2.0 finnes det navnekonvensjoner som gjør det enkelt å holde styr på hva en gitt fil skal brukes til. Filene som har filnavn som inneholder “.component.” er komponenter som vi til slutt skal kunne se, eller bruke i den ferdige applikasjonen. Filene som har filnavn som inneholder “.module.” er filer som har ansvar for å definere hvilke moduler vi har med fra Angular-rammeverket og hvordan vi bruker dem. Filen “app-routing.module.ts” inneholder beskrivelse av hvordan vi ønsker å bruke Routing fra Angular-rammeverket, mens filen “app.module.ts” inneholder en oversikt over hvilke komponenter og andre moduler som er en del av hele applikasjonen og som den da trenger tilgang til.

Vi har organisert mappestrukturen etter hvilke deler av applikasjonen vår vi viser til brukeren. I “app\home”-mappen ligger hovedsiden vår, og i “app\movie”-mappen ligger de komponentene som har med filmer å gjøre. I “app\navbar”-mappen ligger komponentene som skal være inne i navigasjonsbaren. I “app\profile” ligger komponentene som hører til profil-siden.

For å få et eksempel på en komponent kan vi se på “front-page.component”-filene, som ligger i mappen “app\home”. HTML-filen blir brukt inne i ts-filen, men blir liggende utenfor i en egen fil for å gjøre koden mer ryddig og oversiktlig. Inne i ts-filen blir det definert variabler, selector og hvilken fil HTML-delen av komponenten ligger i. For mer detaljer om hver av “.component.”-filene, se kommentarene som ligger i den filen du ønsker mer informasjon om.

CSS kan også legges inn til hver komponent, og dette er gjort til navbar.component, i mappen “app\navbar”. Dette er lagt til på samme måte som vi la til HTML-filen i “front-page.component.ts”. I tillegg til dette har vi med den overordnede “style.css”-filen, som blir hentet inn allerede i “index.html” på samme måte som når man skriver vanlig html-kode.

Typings-mappen inneholder TypeScript-definisjoner som brukes for at kompilatoren skal gjenkjenne funksjoner.

FacebookComponent er en komponent som håndterer kall mot Facebook sitt API. Den har en template med login -og logout-knapper som settes inn i navbar-komponenten. Knappen har en (click)-funksjonalitet som kaller på login og logout, og henter brukerdata fra API-et. Det som hentes foreløpig er navn, epost og bilde-url. Det som mangler her er å få lagret dataene og satt de i komponentene slik at de vises på nettsiden. Det er i tillegg lagt til en ekstra typings-fil(typings/globals/fbsdk.d.ts) for å kunne bruke FB-definisjoner i TypeScript.

DATABASEFILER

Vi har for øyeblikket to databasefiler: "db-init.js" og "db-handler.js". Init-filen står for å opprette databasen med de korrekte tabellene, attributtene, og å sette in "dummy"-data. Denne prosessen kjøres kun hvis databasefilen ikke eksisterer fra før. Handler-filen er ansvarlig for å håndtere videre kommunikasjon med databasen - dette gjelder både oppdatering, innsetting, sletting og uthenting av data fra databasen. Det er denne filen vi kommer til å jobbe mest mot, da den skal være ansvarlig for å hente ut brukerdata eller filmdata, slik at vi kan bruke disse i webapplikasjonen vår.