
Gruppe 1

Systemutviklingsprosjekt TDAT 3022
Systemdokumentasjon

Versjon 1.0

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
17/12/19	1.0	Versjon 1.0	Kristian Kampenhøy, Simon Lilleeng

Innholdsfortegnelse

Revisjonshistorie	2
Innholdsfortegnelse	2
Introduksjon	4
Arkitektur	4
Prosjektstruktur	4
Server-tjenester	4
Sikkerhet	4
Installasjon og kjøring	5
Testing	5
Vedlegg	5

Introduksjon

Systemdokumentasjonen er skrevet i forbindelse med Systemutviklingsprosjekt TDAT 3022, høstsemesteret 2019.

Dette dokumentet viser en overordnet oversikt over hvordan systemet er laget. Dokumentet inneholder en instruksjon på hvordan man kan kjøre systemet, og hvilke avhengigheter som må installeres.

Arkitektur

Se vedlegg for arkitekturtegning

Prosjektstruktur

Prosjektstrukturen er delt inn slik at server-delene er adskilt fra resten av prosjektet som egen moduler.

rabbitMq katalogen inneholder rabbitMq serveren og tilhørende avhengigheter.

nodeServer katalogen inneholder en node server som kommuniserer med databasen gruppen har satt opp.

Denne er laget for å kunne ta i bruk Expo sitt API for push-varsler.

Se vedlegg for mappestruktur

Server-tjenester

Det er satt opp en Node-server i forbindelse med bruk av Expo sitt Api for push-varsler. Overordnet oversikt over de endepunktene som er implementert i systemet på serveren.

Endepunkt	Verb	Hva	Type
/token	GET	Henter Expo tokens	Data
/token/:tokenId	GET	Hent Expo token	Data
/token	POST	Legge til Expo token	Data
/pushSettings/:target	PUT	Oppdaterer push-innstillinger	Data
/token/:tokenId	DELETE	Sletter Expo token	Data
/pushSettings/:tokenId	GET	Henter push-innstillinger til en Expo token	Data

Sikkerhet

Løsningen har lagt inn beskyttelse mot SQL-injection med å bruke PreparedStatements på server-siden, som gjør at spørringer og data til spørringer sendes separat til databaseserveren.

Ved innlogging bruker løsningen et webview fra Signicat som tar for seg autorisering av brukere gjennom innlogging med bankID.

Installasjon og kjøring

Ved installasjon og kjøring må brukeren kjøre følgende kommandoer:

\$ git clone /repo

Klone repository fra github. Prosjektet laster ned et kopi av prosjektet lokalt på maskinen.

Deretter må bruker flytte seg til mappe der prosjektet blir lagret og kjøre følgende kommando

\$ npm install

\$ cd nodeServer/ npm install

Tilhørende avhengigheter for å kjøre applikasjonen blir lastet ned

\$ cd ..

\$ npm start

Bruker flytter seg tilbake til rot plasseringen og kjører **\$ npm start** for å starte applikasjonen. Nå vil det dukke opp en Qr-kode som brukeren kan skanne med sin mobil for å kjøre applikasjonen. For å få til å kjøre applikasjonen på sin mobil må brukeren laste ned “expo-client” fra AppleStore(IOS) eller GooglePlay(android)

\$ cd nodeServer

\$ npm start

Bruker flytter seg til nodeServer-mappen og starter opp serveren.

Testing

Enhetstesting med bruk av rammeverket Jest som hovedrammeverk på både server-siden og klient-siden. På klient-siden er det også brukt rammeverket Enzyme fra Airbnb.

Legg ved instruksjoner på hvordan en kjører testene.

For å kjøre testene må brukeren flytte seg til mappen “tests” ved å kjøre følgende kommando i terminalen:

\$ cd tests

deretter for å kjøre testene kan brukeren kjøre følgende **\$ npm test**, denne kommandoen vil da kjøre alle testene som ligger i samme module.

For å kjøre kun en test fil, kan brukeren kjøre kommandoen: **\$ npm test navn_på_fil.test.js**

Vedlegg

Systemarkitektur

Mappestruktur