

IDG2001 Cloud Technologies – oblig1 rapport

Gruppemedlemmer: Håvard, Trym og Ingrid

Skybasert plattform

Applikasjonen vår er både liten og relativt spesifikk, ved at nettsiden tilbyr bare opplastning og nedlastning av filer med kontaktdata. Derfor, basert på funksjonaliteten til tjenesten vår, mener vi at Platform as a service (PAAS) ville passet best til dette prosjektet. Da har vi mulighet til å komme raskt i gang med utviklingen av applikasjonen og fokusere på funksjonaliteten, siden PAAS tjenester tar seg av infrastrukturen.

I tillegg vil det være fordeler knyttet til tilgjengelighet og sikkerhet. PAAS tjenester tilbyr vanligvis høy tilgjengelighet og feiltoleranse, noe som forsikrer at applikasjonen er oppe og går, selv om det er feil i hardware eller software. Dette hjelper til med å forhindre tap av data og gjør det mulig så brukerne alltid kan ha tilgang til kontaktdataen. Tjenestene tilbyr også ofte avanserte sikkerhets egenskaper, som kryptering, tilgangskontroller og overholdelsessertifiseringer, som kan bidra til å beskytte brukerdata og forhindre uautorisert tilgang.

Hvordan funker systemet

Front end

Brukeren har to muligheter, enten laste opp en vcard fil (vcf) og/eller laste ned en vcard fil. Vcard filen må være i versjon 4.0, og blir returnert i versjon 4.0. Den opplastede filen blir lest av clienten og sendt til API'en som et json format. {"message": "vcardfilen i tekstformat"}.

Den nedlastede filen er kontaktdata fra databasen som er sendt fra API'en som et json format. {"message": "vcardfilen i tekstformat"}. Clienten lager filen utifra vcard teksten fra API'en

Back end

API'en har 5 end-points, hvor fire av de er get requests som henter ut data (enten all data eller data basert på id) fra databasen i forskjellig format (vcard eller json). Den siste er en post request som sender data til databasen. API'en gjør om vcard dataen til json objekter før den sender det til databasen. Før API'en sender ut vcard data fra databasen til clienten, blir dataen gjort om fra json til vcard.

POST /contacts

I denne routen blir filen med det ferdig strukturerte innholdet lagt til i databasen i JSON format. Dette er den routen som front end bruker når brukeren trykker på upload.

GET /contacts

Alle kontaktene blir vist i JSON format.

GET /contact/vcard

Alle kontaktene blir vist i vcard format innenfor JSON struktur. Dette er den routen som front end bruker når brukeren trykker på download vcard file

GET /contact/<id>

Vise kontakten basert på id-en.

GET /contact/<id>/vcard

Vise kontakten basert på id-en og vise det i vcard format innenfor JSON struktur.

Våre valg

Separert front end og back end

Vi har separert front end og back end i to forskjellige mapper i samme git repo. Railway hoster også disse separert. Dette gjør at API'en vår er separert fra clienten, for å løse CORS problemer relatert til dette har vi valgt å tillate bare vår client url i API'en.

Front end er skrevet i javascript. Back end er skrevet i python. På railway har vi en tilknyttet mongodb database.

Formattering av data

I API'en valgte vi å formattere dataen i vcard filen før vi lagret det som json format i databasen. Dette valgte vi å gjøre på bakgrunn av mulighet for videreutvikling av API'en. Det vil si at en fremtidig versjon av API'en kan gjennomføre endringer på kontakters informasjon uten å endre på grunnstrukturen i API'en. Siden hver enkelt kontakt lagres separat i databasen vil det også være mulig å slette enkelte kontakter fra API'en, dersom det i fremtiden blir en ønsket funksjon.