Høgskulen på Vestlandet

Fakultet for ingeniør- og naturvitskap

Institutt for datateknologi, elektroteknologi og realfag

Utvikling av kategoriseringsverktøy for PDF-notesett
Systemdokumentasjon

Versjon 1.1

Dokumentet er basert på Systemdokumentasjon utarbeidet ved NTNU. Revisjon og tilpasninger til bruk ved IDER, DATA-INF utført av Carsten Gunnar Helgesen, Svein-Ivar Lillehaug og Per Christian Engdal. Dokumentet finnes også i engelsk utgave.

REVISJONSHISTORIE

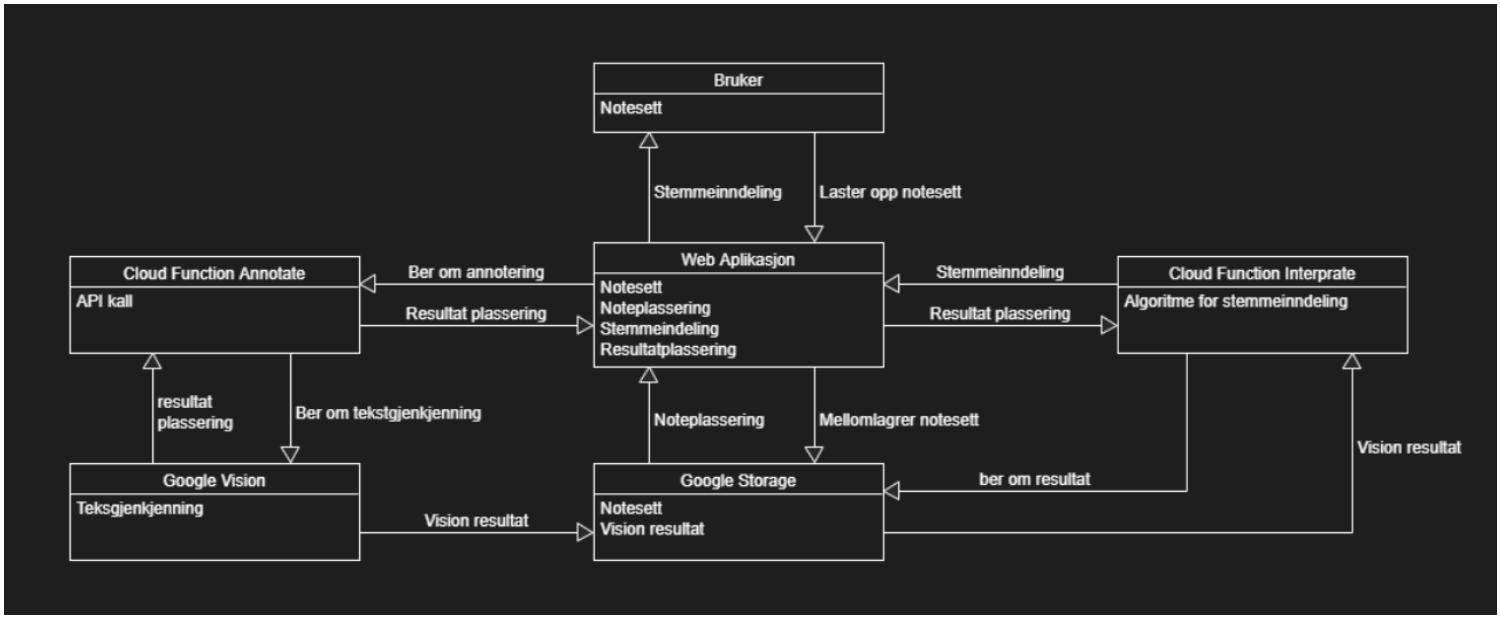
Dato	Versjon	Beskrivelse	Forfatter
21/mai/22	1.0	Første versjon av systemdokumentasjon	Johan Magnus Engevik
22/mai/22	1.1	Andre versjon av systemdokumentasjon	Johan Magnus Engevik

INNHOLDSFORTEGNELSE

1 INNLEDNING

Dette dokumentet er skrevet for å gi kort informasjon om hvordan løsningen vår er bygget opp. Det er også lenke til kildekoden, og til en fungerende web applikasjon. Det er også en installasjonsveiledning som kan brukes dersom en ønsker å lage dette i et eget firebase prosjekt.

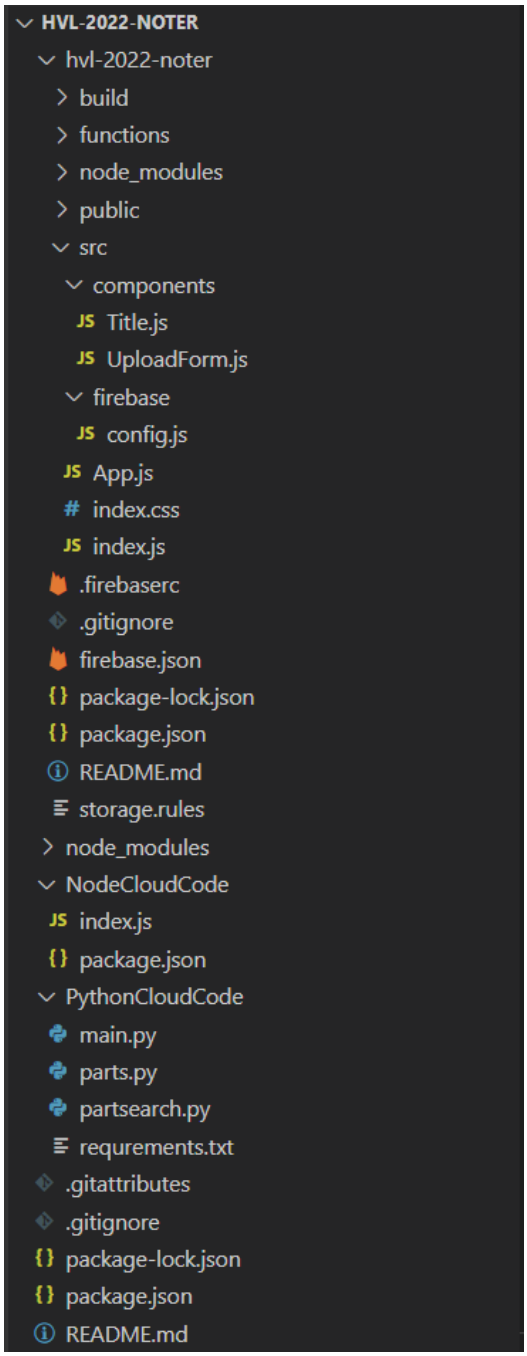
2 ARKITEKTUR



Figur 1 - Prosjekt arkitektur

I Figur 1 vises en skisse av hvordan arkitekturen til prosjektet er. En bruker laster et notesett opp Google Storage ved hjelp av en web klient. Deretter kalles en skyfunksjon som ber Google Vision API-et om tolking av notesettet som er lastet opp. Vision lagrer så dette resultatet i Google Storage og sender plasseringen tilbake til skyfunksjonen som igjen sender den tilbake til klienten. Klienten kaller deretter en annen skyfunksjon for å tolke Vision resultatet og lage en liste av stemmer. I http forespørselen sendes plasseringen til resultatet klienten fikk av forrige skyfunksjon. Skyfunksjonen henter deretter resultatene fra denne plasseringen og lager en liste av stemmer som den sender tilbake igjen til klienten som viser denne inndelingen til brukeren som lastet opp notesettet.

3 PROSJEKTSTRUKTUR



Figur 2- fil katalog for prosjektet

Figur 2 viser hvordan prosjektet vårt er satt opp. Under hvl-2022-noter er klient koden, under NodeCloudCode er koden til skyfunksjonen som kaller Vision API-et, og under PythonCloudCode er koden til skyfunksjonen som tolker dette resultatet om til en stemme

4 SERVER-TJENESTER

Løsningen baserer seg på Firebase sin «Backend as a Service». Det er ikke direkte koding av REST-servere, men det er blitt laget to «Google Cloud Functions» som utfører tjenerside logikk.

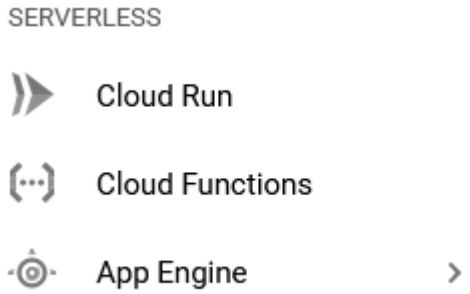
Den første REST ressursen er en skyfunksjon som kaller Googles Vision API til å utføre Tekstgjenkjenning på et tekstdokument på en bestemt plassering. Plasseringen til dokumentet i pdf-format sendes som dataen i en http POST forespørsel. Funksjonen henter denne informasjonen fra forespørselen og kaller Vision API-et med denne plasseringen. Resultatet lagres på en ny plassering, og denne plasseringen sendes i retur med http responsen.

Den Andre REST ressursen er en skyfunksjon som tolker resultatet fra Vision. Plasseringen til Vision resultatet sendes som dataen i en http POST forespørsel. Funksjonen henter deretter resultatet på denne plasseringen og lager en liste over stemmer som sendes i retur med http responsen.

5 INSTALLASJON OG KJØRING

For at dette prosjekt skal kunne kjøres må det først lages et prosjekt i Firebase som prosjektet kan kobles opp mot [1]. Prosjektet er også avhengig av at Goole Vision API-et er aktivert [2]. Installasjon av avhengigheter gjøres med NPM [3].

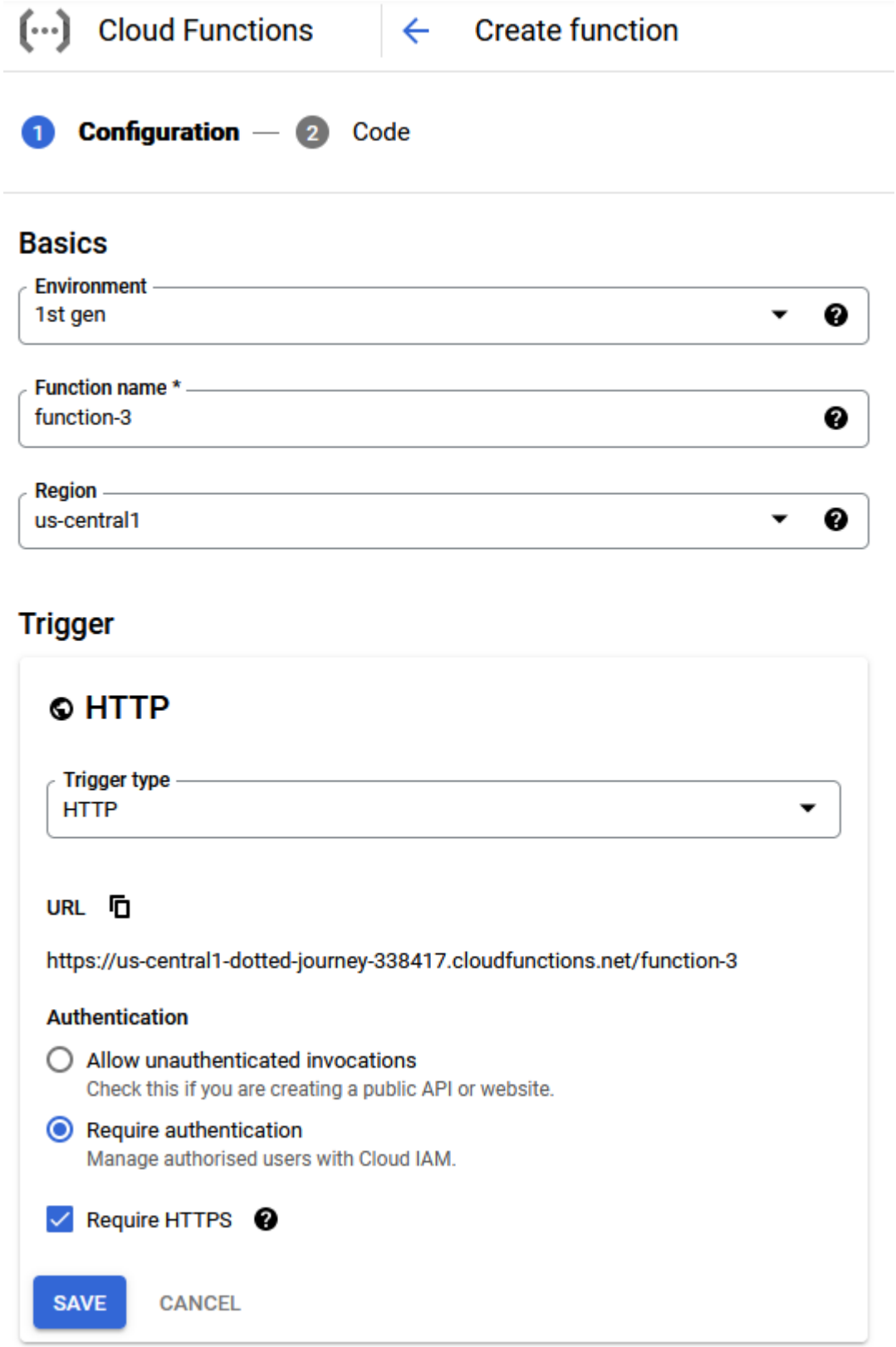
Vi starter med å konfigurere Google Cloud Functions. I Google Cloud konsollen [4] velger man dashbord og deretter «Cloud Functions» fra listen på venstre side slik som vist i Figur 3.



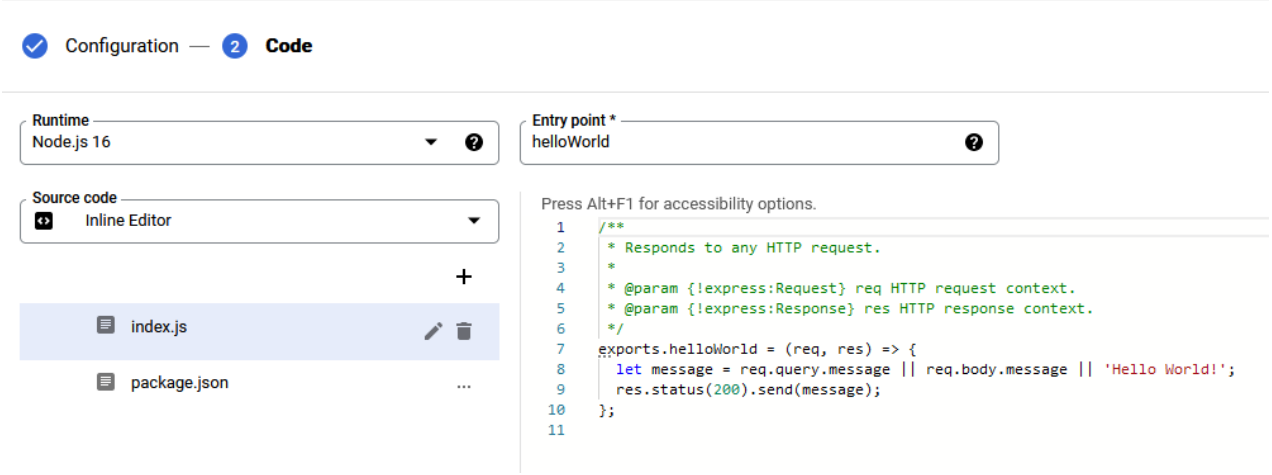
Figur 3 - cloud functions i menyen

Videre velger man «create function» og får menyen vist i Figur 4. Her velger man «1st gen» som miljø. Region velges etter behov og navnet settes etter ønske, men triggeren til funksjonen må oppdateres i klientkoden. Vi velger trigger type som HTTP og velger at vi tillater ikke-autentiserte påkallelser. Trykk så på «next» for å gå videre til kodekonsollen slik som vist i Figur 5. Her kopierer vi koden fra NodeCloudCode i kildekoden til index.js og package.json. Vi lar «runtime» være Node.js 16 og oppdaterer «entrypoint» til «annotateImage». Til slutt trykkes det på «deploy» og funsjonen utplasseres. Dette er funksjonen som kaller Vision API-et.

Videre må det lages en funksjon til for å tolke resultatet fra Vision. Denne funksjonen lages helt likt som forrige på konfigurasjons siden slik som vist i Figur 4, men i kodekonsollen velges i stedet «Python 3.9» som «runtime». Deretter kopiers innholdet fra PythonCloudCode i kildekoden inn i konsollet, og her må man også opprette to nye filer, «parts.py» og «partsearch.py». Til slutte utplasseres også denne funksjonen ved å trykke på «deploy»



Figur 4- Cloud Function konfigurasjon



Figur 5 - Cloud Functions kode

I Firebase konsollen velges prosjektet som ble laget tidligere. Her velges «add app» og deretter «Web». Gi appen et navn. Deretter kopieres koden i firebaseConfig til samme plass i filen config.js under firebase i kildekoden, dette for å koble klienten opp mot rett firebase prosjekt. Trykk derett «Continue to console».

Hent URL-ene som utløser skyfunksjonene som vi utplasserte tidligere og legg dem inn i UploadFrom.js. URL-ene finnes i Google Cloud konsollen. Velg «Cloud Functions» og klikk på Node.js funksjonen. Velg deretter «Trigger» fra menyen øverst. Kopier URL-en til feltet «url:» under funksjonen «httpHandler». Gjør det samme med Python funksjonen og kopier URL-en til feltet «url:» under getInterpretation.

Åpne en terminal i mappen hvl-2022-noter og installer avhengigheter med kommandoen «npm install». Logg på firebase med kommandoen «firebase login». Initialiser prosjektet med komandoen «firebase init». Velg «Hosting» og «Storage» under oppsettet. Når det blir spurt om mappe for «public root directory», skriv mappen «build». Når det spørres om «Storage rules», trykker man enter for standard regler.

Etter oppsettet er ferdig kjøres kommandoen «npm run build» for å bygge filstrukturen som skal utplasseres. Deretter kjøres kommandoen «firebase deploy –only hosting» for å utplassere koden på firebase.

Til slutt må en gå inn i firebase konsollen og velge «Storage» i menyen til venstre. Trykk deretter på rules i menyen over. Kopier koden fra storage.rules i kildekoden.

Nå bør klienten fungere på prosjektets webadresse.

6 DOKUMENTASJON AV KILDEKODE

Kildekode er lagt på GitHub https://github.com/h584991/HVL-2022-noter/

Koden er privat, men vi har laget en GitHub konto som har tilgang.
Brukernavn: d17-test-user@hotmail.com
Passord: fACeE9g7JeAFXD

Applikasjonen kan testes på https://hvl-2022-noter.web.app

7 REFERANSER

[1] Google, «Getting started with Firebase,» 20 Mai 2022. [Internett]. Available: https://cloud.google.com/firestore/docs/client/get-firebase. [Funnet 22 Mai 2022].

[2] Google, «Before you begin | Cloud Vision API | Google Cloud,» 20 Mai 2022. [Internett]. Available: https://cloud.google.com/vision/docs/before-you-begin. [Funnet 22 Mai 2022].

[3] npm, Inc., «npm - npm,» 12 Mai 2022. [Internett]. Available: https://www.npmjs.com/package/npm. [Funnet 22 Mai 2022].

[4] Google, «Google Cloud Console,» [Internett]. Available: https://console.cloud.google.com/. [Funnet 1 Mai 2022].