

IS	105
Gruppe 3 NoName	ICA05

Gruppemedlemmer:

Ali Al Musawi
 Tor Borgen
 Ann Margrethe Ly Pedersen
 Brage Fosso
 Adrian Lorentzen
 Arne Bastian Wiik
 Morten Schibbye

Alle kode henvisninger ligger i README.md

<https://github.com/GB-Noname/is105-ica05>

Hva er ideen bak siden?

I ICA05 så har vi bestemt å utvikle en samlingsside for noe vi mener er nyttig informasjon. Vårt formål var å teste forskjellige API-er, for å få en bedre forståelse på hvordan vi kommuniserer mellom ulike software komponenter gjennom bruk av Application Programming Interfac (API). Vi merket også fort at det var vanskelig å finne dokumenterte api-er under arbeidet for denne ICA-en.

Hvordan skal det formidles? (brukergrensesnitt) Dvs. hvilke data som skal vises og hvordan? (systemarkitektur, inkludert programmeringsmiljøer, datalagring osv.)

Dataen vises i en webside i html-format ved bruk av søkefelt eller taster.

Golang miljø

Søker på ip-adressen. Og får informasjon om server. Hvor den er registrert og ISP.

Så timezone informasjon, hvor serveren er.

Så får du informasjon om været i Kristiansand live.

Du får også en random pokemon.

Så lokasjonsdata i koordinater.

Så får du kart over hvor IP er registret ut i fra lokasjonsdata

Istedenfor å laste hele standard templates med ikke brukte API'er i så har vi forsøkt på en dynamisk parsing av templates for hver API. Denne fungerer, men kan ende opp med å bygge på seg selv og ende opp med flere visninger av samme API. Ikke funnet måte å flushe for hver gang.

Så se bort ifra eventuelle bugs med søk av enkelte API. Er hovedsiden etter trykk på "Show formatted JSON" med ingen tekst i "INPUT" som er hovedappen.

Hvilke api-er brukes? (systembeskrivelse, hvilke funksjoner som gjør hva)

IP - ren IP henter

IP Search - søker info om IP-adresse

OWL - værdata på lokasjon

Pokemon - random pokemon

Google Maps embeded html - viser din lokasjon ut ifra hvor IP til serveren er registrert

TimeZone - hvilken tidssone du er i

OWL og TimeZone er statiske forespørsler, mens de andre er dynamiske.

Hvis vi hadde implementert caching til de statiske API'ene så hadde man kunne spart mye ressurser.

For eksempel API'er som ikke oppdateres ofte eller som har begrensede forespørsler, vil caching av data hjelpe med å redusere unødvendig trafikk eller spare penger på betaling av forespørsler.

Test-scenarioer

- Mange forespørsler
- Ulike variasjoner av forespørsler

Bruker scenario:

Bago befinner seg hjemme i sin nye leilighet i Kristiansand. Han har nettopp flyttet hit fra hjembyen Bergen. Han er veldig glad i å gå på tur, så han ønsker å utforske byen. Derfor åpner han nettsiden vår for å se om været er fint og hvordan temperaturen er slik at han kan kle seg riktig. Bago er leder for en skattejaktklubb, som skal få barn til å tilbringe mer tid utendørs. Han tar med seg flere gjenstander som han skal gjemme på ulike lokasjoner. Han bruker nettsiden til å finne lengde og -breddegrader på stedene han skal gjemme "skattene". Disse lengde og -breddegradene publiserer han så på forumet til skattejaktklubben, og så er det førstemann til mølla for å finne skattene. Den personen som samler inn flest skatter vinner en premie. Klubben har en felles interesse for Pokemon. Bago bruker da nettsiden for å genere en tilfeldig Pokemon. Han går til innkjøp av en kosebamse av den gitte Pokemonen, som vinneren av skattejakten får.

16 år gamle Bob Kåre har nettopp fått ny pc til bursdagen sin. Det første han gjør er å installere et dataspill som alle vennene hans spiller. Han oppdager fort at det er problemer med internett siden forbindelsen er dårlig når han snakker over Skype med sine kompiser. Han benytter seg derfor av vår nettside for å finne informasjon om IP adressen han har. Siden han bare er 16 år, er det foreldrene som betaler for internettet. Han vet derfor ikke hvilken ISP familien benytter seg av. Det får han heldigvis informasjon om på nettsiden vår, slik at han kan kontakte riktig internettleverandør, for å løse problemet med internettforbindelsen.