

Oppgave 3.3 - Dokumentasjon

Gruppe 4

Vår applikasjon

Vår applikasjon skal brukes til å vise og legge til observasjoner av sjeldne arter. Data er hentet fra Artsdatabankens database. Vi vil putte dataene inn i vår egen database slik at vi kan legge til og redigere data. Observasjonene skal vises i listevisning og kartvisning. Det skal være mulig å filtrere observasjoner ut i fra navn på art, og eventuelt noen andre kriterier. I listevisningen skal man få detaljerte data om en observasjon ved å klikke på en valgt observasjon.

Dokumentasjon av applikasjonen

Sluttproduktet

Her vil dokumentasjonen av sluttproduktet vises.

Prototypen

Demoen inneholder en nedkortet liste av observasjoner hentet direkte fra Artsdatabankens API. Det er mulig å filtrere arter ut i fra deres navn, via et søkefelt. Øverst finnes en knapp for å bytte mellom listevisningen og en dummy-side som etterhvert skal inneholde kartvisningen av observasjoner. Ved å trykke på et listeelement vises nøkkeldata om observasjonen. I prototypen er det lagt til en hyperlenke til observasjonens koordinater i Google Maps. Denne hyperlenken vil fjernes i den endelige versjonen siden vi da vil bruke vår egen kartvisning.

Sjeldne arter

Liste/Kart

søk etter art...

moskusfe

moskusfe

Vitenskapelig navn

Ovibos moschatus

Dato

27/07/2011

Sted

Dovre, Oppland ([62.277541, 9.429403](#))

Registrert av

Olsen, K.M.; Olsen, C.; Olsen, S.

Beskrivelse

Veikant og rabbe

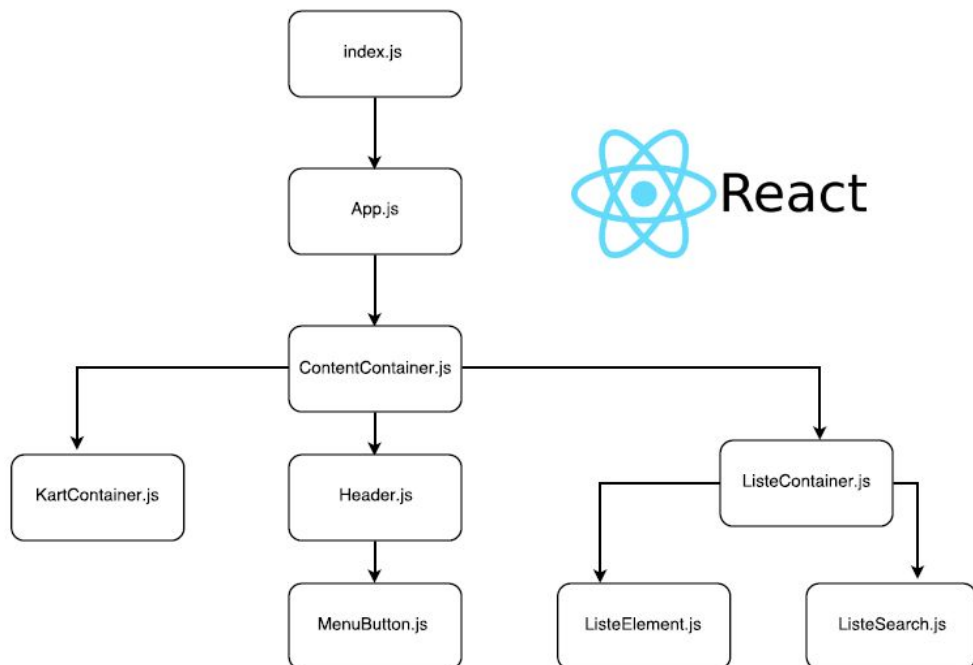
bredøre

storflaggermus

storflaggermus

storflaggermus

Struktur



Vi benytter komponenter som i størst mulig grad er uavhengige av hverandre og kobler dem sammen til en fungerende app. Figuren over viser hvordan avhengighetene mellom komponentene er, der løvnodene er fullstendig frittstående.

Øverst i strukturen finnes så klart **App**-komponenten. Her vil eventuelle “routes” bli bestemt for hvilket innhold som skal føres videre. App inneholder en komponent kalt **ContentContainer** som alt annet innholdet puttes inn i. ContentContainer inneholder komponentene **Header** og **toggleContainer**. Førstnevnte er visualisert med det horisontale blå feltet øverst i appen, med tittel “Sjeldne arter” og med en **MenuButton**-komponent som brukes til å bytte mellom kart- og listevissning. toggleContainer er referansen som brukes til enten kartvisningen (**KartContainer**) eller listevissningen (**ListeContainer**). KartContainer viser foreløbig kun en tekst: “Her kommer det kart etterhvert”.

Funksjonalitet

Header lytter på MenuButton sitt **onClick** event og håndterer det i en ny handler delegate kalt **toggleHandler**. I ContentContainer registreres en handler-funksjon for toggleHandler, kalt **toggleEvent**. toggleEvent oppdaterer toggleContainer og GUI skifter visning vha. React sin **setState**-funksjon.

Når ListeContainer initieres kjøres konstruktøren som henter inn data til listevissningen i vha. metoden **fetchHandler**. fetchHandler kjører en HTTP GET request mot Artsdatabankens API og prøver å hente ut inntil 50 observasjoner av artene som vi har spesifisert, i et **JSON**-objekt. Dersom GET requesten feiler skriver vi feilmeldingen til konsollen. Om

dataene finnes, skriver vi til **data**-variabelen. I konstruktøren initieres også søketeksten **searchFilter** som skal brukes for å filtrere ut observasjoner basert på artsnavn.

I **render**-funksjonen til **ListContainer** benyttes søketeksten til å hente ut observasjoner av arter med navn som inneholder søketeksten. Ved å søke på “mos” hentes for eksempel alle observasjoner av moskusfe ut. For hver observasjon i den filtrerte lista oppretter vi en ny listelement-komponent, **ListElement**. Listelementet gis en unik id og puttes inn i **ListContainer**. Id-en brukes til å vise detaljvisning av elementet når man klikker på det. Selve detaljvisningen er ordnet ved at listelementet oppfører seg som en checkbox som huker av/på detaljvisningen vha CSS og label-tag.

Søkefilteret er knyttet til et input-felt i komponenten **ListSearch**. **ListSearch** oppretter en ny handler delegate kalt **changeHandler** for input-feltets **onChange** event. I **ListContainer** registreres en handler-funksjon for changehandler, kalt **changeEvent**, som påfører søkefilteret på lista over observasjoner og oppdaterer GUI vha. **setState**.

Prosjektstruktur (Bitbucket)

- artsdata-angular2
 - app - her ligger alle komponentene
 - typings - Typescript-definisjonsfiler for bruk av eksterne biblioteker
- artsdata-react
 - public - her ligger index.html og favicon
 - resources
 - data - her ligger data (i json-format) som skal brukes i MongoDB
 - observations - observasjonsdata per art
 - src - her ligger alle komponentene