Vejledning

## Eksempel 1 - Tematisk danmarkskort

I dette eksemplev vises hvordan man kan tematiser eksempelvis kommuner efter nogle værdier. Det kunne være andre område inddelinger. I DAGI datasættet fra GST findes Kommuner, Regioner, Sogne, Retskredse, Politikredse, Postnumre og Opstillingskredse, (læse mere på <http://www.gst.dk/Emner/Frie_data/Hvilke+data+er+omfattet/Hvilke+data+er+frie/Landinddelinger/DAGI/>). I dette eksempel anvender vi Kommunegrænserne. DAGI datasættet findes i flere opløsninger, til tematiseringformål er versionen i måle­stoks­for­hold 1:2 mill et godt valg

### Hent data fra GST

DAGI datasætte downloades fra <ftp://ftp.kms.dk/DAGI/SHAPE/DAGI2M.zip>.

Data ligger i GIS formatet Shape (kan også downloadses i andre formater). Da vi skal vise data på internettet i OpenLayers ønsker vi at konvertere data til GeoJSON. Dette kan gøres med forskellige værktøjer, fx. komandolinjeværktøjet OGR (<http://trac.osgeo.org/gdal/wiki/DownloadingGdalBinaries>). Vi anvender en online service til at konvertere data fra Shape til GeoJSON <http://ogre.adc4gis.com/>.

Første skridt er at udpakke DAGI datasættet fra GST. Derpå skal der laves en zip-fil der kun indeholder der tre filer: KOMMUNE.dbf, KOMMUNE.shp og KOMMUNE.shx. Denne fil uploadses til <http://ogre.adc4gis.com/>. Source SRS og Target SRS sættes begge til EPSG:25832. Derefter konverteres data ved at trykke på “Convert to GeoJSON” resultatet gemmes i en fil - kald den for kommune.geojson.

### Hente data fra Danmarks Statistik (DST).

DST har en masse gratis datasæt der kan hentes fra Statistikbanken. I eksemplet er data hentet fra <http://www.statistikbanken.dk/FRKM213> hvor alle kommuner er valgt, Befolkningen primo og 2013. Data hentes ud i Excel format, husk at sætte kryds i “Inkl. koder i sep. kolonner”, da koden (kommunekoden) er den kode der skal bruges til at koble datasættet sammen med kommunetemaet fra GST. Excel filen åbnes i excel og de overflødige kolonner og rækker fjernes (kolonne A og B samr række 1,2,3 og 4). Herefter gemmes data som CSV fra Excel som indbyggertal13.csv

NB: Det er vigtigt ikke at hente data som CSV direkte fra DST, da filen så ikke indeholder kommunekode, selvom “Inkl. koder i sep. kolonner” er markeret.

### Kode

For at få vist det tematiske kort i en webbrowser kan JavaScript API’et OpenLayers benyttes. Til at starte med tilføjes OpenLayers til siden:

<script src="http://openlayers.org/api/OpenLayers.js"></script>

Fælles for alle eksemplerne, der er beskrevet i dette dokument, tilføjes der først et baggrundskort fra GST. Hvordan det gøres kan ses i filen map.js, hvor det baggrundskort, der hedder “Skærmkort” benyttes.

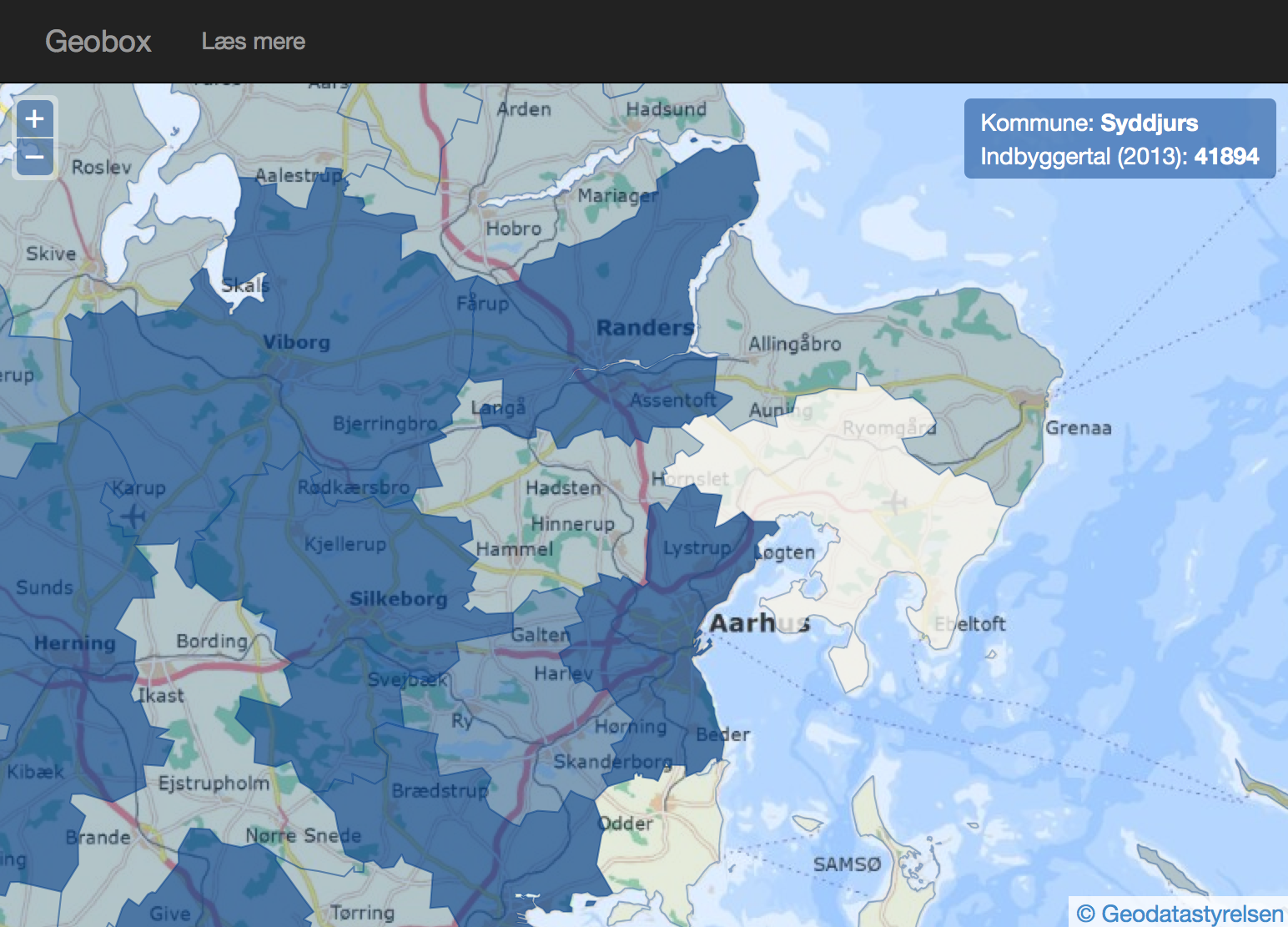
Derefter sætte et passende centerpunkt og zoomniveau (se ligeledes map.js).

Det tematiske kort tilføjes ved hjælp af “showData” funktionen i example1.js. Her tilføjes først laget, der skal vise data. På et lag er det muligt at tilføje en “style”, så data vises på en bestemt måde. Det er også muligt at denne style er afhængig af en værdi, der er angivet på data. Dette benyttes i eksemplet til at tematisere på baggrund af antallet af indbyggere i en kommune. Se detaljer i example1.js.

Dernæst tilføjes funktionalitet så det er muligt at klikke på en kommune og få informationer om den valgte.

Til slut benyttes jQuery.ajax til først at hente kommune.geojson filen ind i browseren. Dernæst hentes indbyggertal13.csv og sammenstilles med data fra kommune.geojson.

Eksemplet kan bruges til at vise et hvilken som helst datasæt med oplysninger på kommuner med meget få tilpasninger.



NB: For at danske tegne skal vises korrekt er det vigtigt at være opmærksom på det valgte tegnesæt på både server og filer. Det anbefales at anvende UTF-8 som tegnesæt hvilket også fremgår af html-filen.

## Eksempel 2 - Visning af CVR data på et kort

I dette eksempel vises hvordan det er muligt at vise flere punkter på et kort ud fra en liste. Vi tager udgangspunkt i en række adresser som er hentet fra CVR.dk. Data fra CVR.dk omhandler virksomheder og er, lige som data fra GST, en del af de frie gruinddata. Datasættet vi har valgt at vise, indeholder virksomheder med branchekode “107120 Fremstilling af friske bageriprodukter”. Det er udelukkende et eksempel, og det kunne lige så godt være butikker i en butikskæde eller golfklubber der er medlem hos Dansk Golf Forbund.

Oftest kender man adressen men ikke koordinaten på de poster man ønsker at vise på kortet. Dette er også tilfældet med de data vi har hentet fra CVR. Her er der en adresse men ikke en koordinat. Derfor bruger vi en service fra GST til at geokode vores poster. Altså en service der oversætter adressen til en koordinat som vi efterfølgende kan vise på kortet.

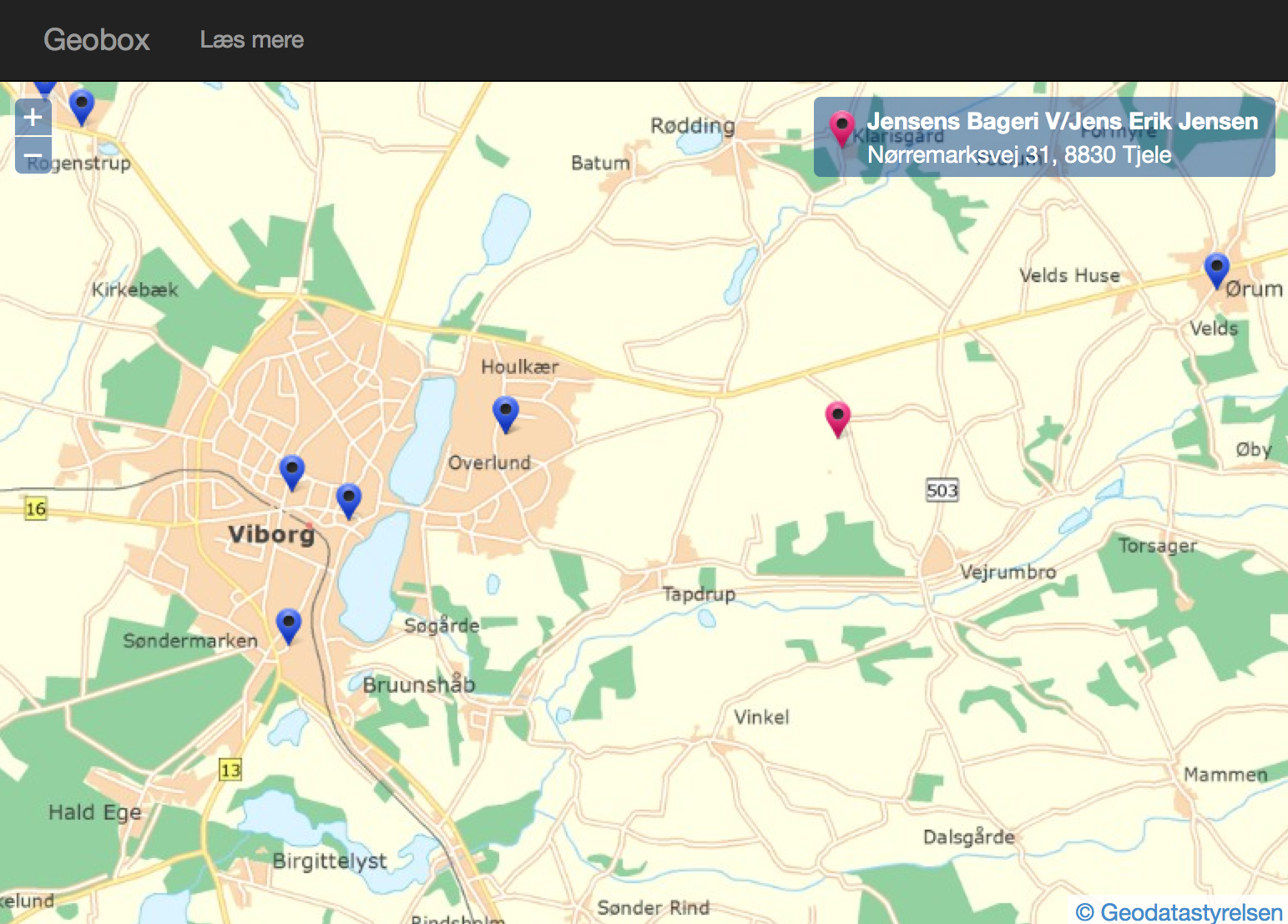
Har du eksempelvis en liste over koncertsteder til den årlige byfest kan du bare kopiere koden og udskifte data fra CVR med dine egne data og lave en kort der viser det.

Data fra CVR.dk gemmes som csv under navnet virksomheder.csv.

### Kode

Som i eksempel 1, så tilføjes baggrundskort fra GST og et passende centerpunkt og zoomniveau vælges. I dette tilfælde omkring Viborg.

Virksomheder.csv indlæses vha. jQuery.ajax. For hver række i filen, kaldes en REST service fra GST. Læs mere om anvendelsen REST servicen på <http://kortforsyningen.dk/dokumentation/geonoeglerrest>. Se detaljer om hvordan det gøres i example2.js.



NB: For at danske tegne skal vises korrekt er det vigtigt at være opmærksom på det valgte tegnesæt på både server og filer. Det anbefales at anvende UTF-8 som tegnesæt hvilket også fremgår af html-filen.

## 

## 

## Eksempel 3 - Visning af popup for en given adresse

Dette er et enkelt eksempel på hvordan man kan vise en adresse på et kort. Eksemplet kan bruges hvis du ønsker at vise hvor en virksomhed eller forening bor på et kort hvor brugeren selv kan zome og panorerer. Koden kan bruges som den er ved blot udskifte adressen i kaldet til funktionen showData().

showData('[*vejnavn*]',[*husnummer*],[*postnr*],'[*tekst der skal vises*]’);

*vejnavn* er navnet på vejen

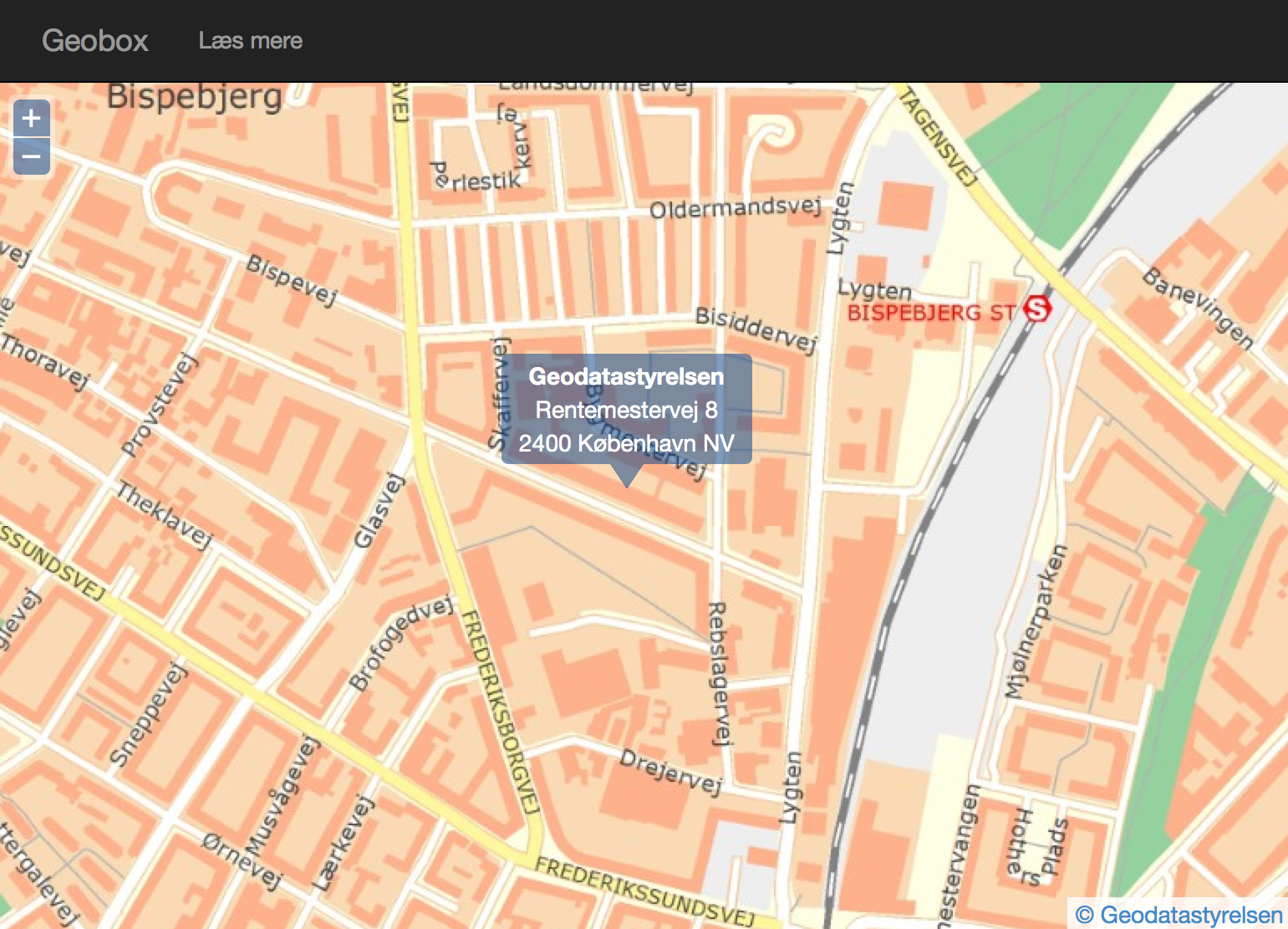
*husnummer* er husnummeret

*postnr* er postnummeret

*tekst der skal vises* er den tekst der vises på kortet. Kan formateres som HTML

*Eksempel:*

showData('Rentemestervej',8,2400,'<strong>Geodatastyrelsen</strong><br/>Rentemestervej 8<br/>2400 København NV');



NB: For at danske tegne skal vises korrekt er det vigtigt at være opmærksom på det valgte tegnesæt på både server og filer. Det anbefales at anvende UTF-8 som tegnesæt hvilket også fremgår af html-filen.