

Podes reiseplanlegger - dokumentasjon

Podes reiseplanlegger - dokumentasjon	1
Introduksjon	2
Installasjon	2
Forutsetninger.....	2
Avhengigheter.....	2
Hent kildekoden	2
Pakket fil	2
Git	3
config-dist.php	3
Autosuggest.....	3
Tilpasninger	3
Oversikt	6
Protokoller.....	9
SRU	9
Z39.50	9
Sortering.....	9
Sortering på årstall	9
Sortering på tittel	10
Moduler.....	10
Bibliografiske moduler	10
culture	11
food	11
language	11
stories.....	11
travel	11
Andre moduler	11
debug	11
map	11
snl	12
weather	13
Referanseliste	13
Vedlegg: Katalogstruktur og eksempelkode.....	14
Katalogstruktur	14
api/mod.food.php	17
api/mod.map.php	18
api/mod.snl.no	19
api/mod.weather.php	20
scripts/bokser.js	23

Introduksjon

Podes reiseplanlegger gir brukerne et grensesnitt der de kan søke etter land og steder, og få en treffliste som automatisk er avgrenset til reisehåndbøker for det aktuelle stedet. I tillegg til denne hovedtrefflista vises en serie "bokser" som viser

- andre relevante dokumenter fra bibliotekets samling, som fortellinger fra det aktuelle stedet, eller språkkurs for de språkene som snakkes i landet
- annen relevant informasjon, hentet fra åpne APIer. Eksempler på dette er kart og værinformasjon.

Installasjon

Forutsetninger

Det skal ikke være nødvendig med programmeringskunnskaper for å installere og ta i bruk denne reiseplanleggeren, men det vil være en fordel at man har en viss glede av å "få ting til å virke". Man må også kunne få på plass det som finnes av avhengigheter, jfr avsnittet under, men dette bør en lokal IT-avdeling/serverleverandør kunne bistå med.

Avhengigheter

Du trenger følgende komponenter for å kunne ta i bruk Podes reiseplanlegger:

- En webserver som støtter PHP. Applikasjonen er utviklet på Linux/Apache (<http://httpd.apache.org/>), men bør også fungere på Windows-plattformer som støtter PHP.
- PHP (<http://no.php.net/>)
- PHP/YAZ (<http://no.php.net/yaz>, <http://no.php.net/manual/en/yaz.installation.php>)
- Tilgang til et biblioteksystem som støtter protokollene SRU og/eller Z39.50. Ta kontakt med din systemleverandør for informasjon om adresse/URL for tilkobling. (Se http://wiki.biblab.no/index.php/Norske_Z39.50_datatilbydere og http://wiki.biblab.no/index.php/Norske_SRU_datatilbydere for oversikter over norske biblioteksystem som har disse protokollene.)

Hent kildekoden

Kildekoden til Podes reiseplanlegger er tilgjengelig fra en tjeneste som heter GitHub: <http://github.com/pode/reiseplanlegger>

Det er to hovedmåter å få tak i kildekoden på:

Pakket fil

Den nyeste versjonen av kildekoden er tilgjengelig som en zip- eller tarball-fil: <http://github.com/pode/reiseplanlegger/archives/master>

Pakk ut den nedlastede fila i mappa hvor du vil ha kildekoden. På Linux kan du gjøre dette med følgende kommandoer, avhengig av hvilken fil du lastet ned:

```
unzip pode-reiseplanlegger-xxx.zip
tar xvf pode-reiseplanlegger-xxx.tar.gz
```

Git

Dersom du har installert versjonskontrollsystemet Git (<http://git-scm.com/>) på serveren din kan du hente kildekoden ved å lage en mappe der du vil ha den, og klonе prosjektet:

```
mkdir minreiseplanlegger
cd minreiseplanlegger
git clone git://github.com/pode/reiseplanlegger.git
```

En fordel med å gjøre det på denne måten er at du lett kan hente eventuelle nye versjoner av kildekoden med denne kommandoen:

```
git pull
```

Dersom du har planer om å videreutvikle kildekoden og kanskje til og med dele endringene dine med prosjektet anbefales det på det sterkeste at du benytter denne metoden for å få tak i kildekoden.

En innføring i bruken av Git er utenfor rammene av dette dokumentet, men se feks dokumentasjonssiden på Gits hjemmeside for noen tips: <http://git-scm.com/documentation>

config-dist.php

Når du har pakket ut kildekoden må du flytte fila config-dist.php til config.php. På Linux kan du gjøre dette med følgende kommando:

```
mv config-dist.php config.php
```

Autosuggest

Reiseplanleggeren kan foreslå stedsnavn etter hvert som brukeren skriver. Denne funksjonen er basert på et uttrekk fra katalogen til det biblioteket du setter opp reiseplanleggeren for å søke i. Uttrekket lagres i fila dewey/dewey_list.txt. Lista består av en linje pr sted, og hver linje består av stedsnavn og Dewey-nummer, adskilt med kolon, feks slik:

```
Afghanistan:915.8104
Afrika:916.04
Agra:915.4604
Akershus:914.82204
Alaska:917.9804
```

Se kildekoden (http://github.com/pode/reiseplanlegger/blob/master/dewey/dewey_list.txt) for et fullstendig eksempel.

Tilpasninger

Normalt skal du bare trenge å endre på innholdet i fila config.php for å ta i bruk reiseplanleggeren, og sette den opp for å søke mot ditt eget bibliotek. Denne fila er utstyrt med rikholdige kommentarer, så vi gjengir den her:

```
<?php
```

```
// Setter navnet på applikasjonen
$config['app_title'] = 'Podes reiseplanlegger';

// Hvor mange treff skal maksimalt hentes ved søk?
$config['main_limit'] = 100;
// Hvor mange treff skal vises pr "side" i hovedvisningen
$config['mainPerPage'] = 10;

/*
BIBLIOTEK

Her konfigureres de bibliotekene det skal være mulig å søke i, og
de opplysningene som trengs for å utføre søket. Rekkefølgen her
bestemmer rekkefølgen når bibliotekene skal velges ved søk.

Opplysninger som trengs:
title: navn på biblioteket
z3950 ELLER sru og item_url
z3950: tilkoblings-streng for Z39.50
sru: tilkoblingsstreng for SRU
item_url: grunn-URL for postvisning i katalogen
*/
$config['libraries']['deich'] = array(
    'title' => 'Deichmanske',
    'z3950' => 'z3950.deich.folkebibl.no:210/data'
);
$config['libraries']['pode'] = array(
    'title'      => 'Pode',
    'sru'        => 'http://torfeus.deich.folkebibl.no:9999/biblios',
    'item_url'   => 'http://dev.bibpode.no/cgi-bin/koha/opac-
detail.pl?biblionumber='
);
/*
$config['libraries']['bibsys'] = array(
    'title' => 'BIBSYS',
    'z3950' => 'z3950.bibsys.no:2100/BIBSYS',
);
$config['libraries']['trondheim'] = array(
    'title' => 'Trondheim folkebibliotek',
    'z3950' => 'z3950.trondheim.folkebibl.no:210/data',
);
$config['libraries']['bergen'] = array(
    'title' => 'Bergen offentlige',
    'z3950' => 'z3950.bergen.folkebibl.no:210/data',
);
*/

/*
DIVERSE MELDINGER
*/
```

```

$config['msg'] = array(
    'zero_hits' => '<p>Ingen treff...</p>',
);

/*
AUTOSUGGEST

Konfigurerer hvordan autosuggest i søkeboksen oppfører seg.
show_dewey [true|false] - slår av og på visningen av Dewey-nummer
maxresults 10 - bestemmer antall forslag som skal vises
*/

$config['autosuggest'] = array(
    'show_dewey' => false,
    'maxresults' => 10,
);

/*
MODULER

Moduler konfigureres med et array på formen
$config['modules']['MODUL'] = array();
der MODUL tilsvarer den midterste delen av filnavnet modulen
er implementert i: mod.MODUL.php.

Rekkefølgen på modulene nedenfor bestemmer rekkfølgen modulene
vises i på siden. (Men dette kan overstyres av brukerne, som selv kan
flytte rundt på modulene.)

Alle moduler har minst to parametere:
'enabled': true eller false, dvs om modulen er slått av eller på.
'title': tittelen som vises i modul/widget-boksen

Dersom modulen inneholder en liste med elementer hvor antallet
elementer skal kunne begrenses ved hjelp av en parameter gjøres
dette med en parameter som heter 'limit'.
*/

$config['modules']['language'] = array(
    'enabled' => true,
    'title' => "Lærebøker og språkkurs",
    'limit' => 5,
);

$config['modules']['travel'] = array(
    'enabled' => true,
    'title' => "Reiseskildringer",
    'limit' => 5,
);

$config['modules']['stories'] = array(
    'enabled' => true,

```

```

    'title' => "Fortellinger",
    'limit' => 5,
);

$config['modules']['food'] = array(
    'enabled' => true,
    'title' => "Mat og matlaging",
    'limit' => 5,
);

$config['modules']['culture'] = array(
    'enabled' => true,
    'title' => "Kulturhistorie",
    'limit' => 5,
);

$config['modules']['snl'] = array(
    'enabled' => true,
    'title' => "Store norske leksikon",
    'limit' => 1,
);

$config['modules']['weather'] = array(
    'enabled' => true,
    'title' => "Været",
);

/*
For at kart skal fungere må man ha en "Google Maps API key", dette får man
her:
http://code.google.com/intl/en/apis/maps/
*/
$config['modules']['map'] = array(
    'enabled' => true,
    'title' => "Kart",
    'api_key' => 'legg inn din nøkkel her'
);

$config['modules']['debug'] = array(
    'enabled' => false,
    'title' => "Debug",
);

?>

```

Oversikt

Når en bruker utfører et søk skjer følgende:

index.php tar i mot søkebegrepet og utfører et søk mot den valgte bibliotek-katalogen. Søkebegrepet kombineres med andre avgrensninger, i det søkespråket som er aktuelt for den enkelte protokollen, for å gi treff på reisehåndbøker. "\$query" representerer

brukerens søkebegrep:

- Z39.50 (CCL): eo=\$query Reisehåndbøker
- SRU (CQL): dc.subject = \$query and dc.subject = Reisehåndbøker

Se mer om de ulike protokollene nedenfor.

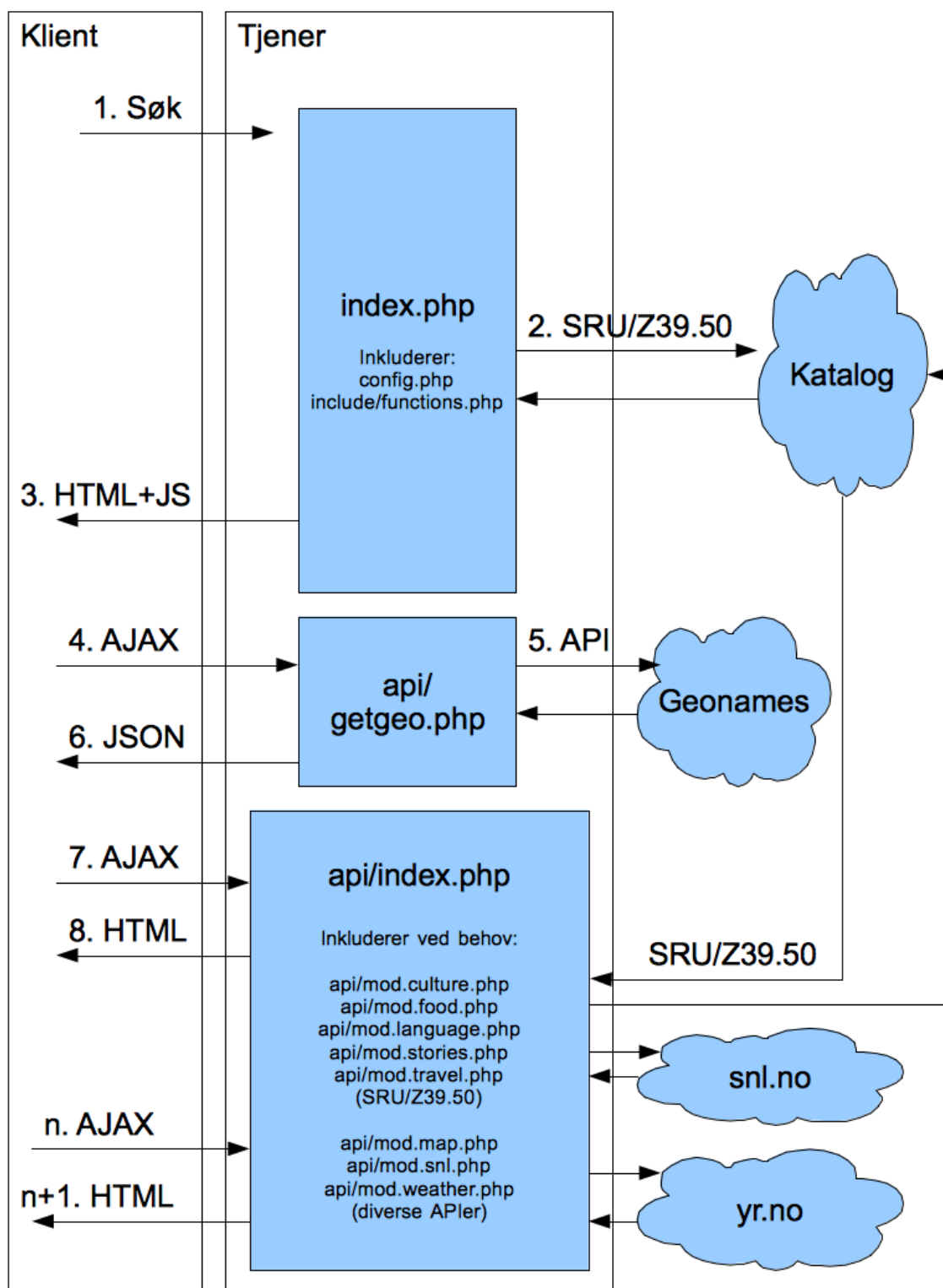
Begge protokollene returnerer treff i form av MARC-poster i XML-format (MARCXML). Disse transformeres til en HTML-presentasjon ved hjelp av XSLT, og resultatet blir at det sendes en HTML-side med trefflista til brukerens nettleser.

Sammen med HTML-siden sendes det en JavaScript-fil, script/bokser.js, som har til oppgave å lage boksene med ekstradata på høyre side av skjermen. Dette skriptet plukker ut hva brukeren har søkt etter, og sender dette til en web service fra Geonames (<http://www.geonames.org/>), via skriptet api/getgeo.php. Dersom søkebegrepet returnerer flere enn ett treff i Geonames blir brukeren bedt om å velge hvilket av disse stedene hun er interessert i. Når et sted er valgt hentes geografiske koordinater (lengde- og breddegrad) og andre relevante opplysninger fra Geonames, og bokser.js kan fylle boksene med innhold. Data hentes fra serveren med AJAX-teknikker, og boksene fylles med innhold mens brukeren venter. Innholdet i boksene er av to hovedtyper:

- Katalogposter - her gjøres det et søk i katalogen på omtrent samme måte som for hovedsøket etter reisehåndbøker, brukerens søkebegrep kombineres med andre søkeuttrykk for å gi andre, relevante treff, feks på fortellinger eller kokebøker.
- Andre data som kart og vær. Innholdet her er ofte basert på de geografiske koordinatene fra Geonames.

Se [kapittelet om moduler](#) for flere detaljer om disse boksene.

Skjematisk fremstilling av funksjonaliteten:



Protokoller

Podes reiseplanlegger søker som standard i to kataloger:

- Deichmanskes Bibliofil-katalog, ved hjelp av Z39.50
- En eksport av alle Deichmanskes bibliografiske poster, importert og indeksert i Koha, ved hjelp av SRU-protokollen

SRU

Se http://wiki.biblioteklaboratoriet.no/index.php/Koha_som_SRU-tjener for en oversikt over hvilke indekser som er tilgjengelige ved SRU-søk i en Koha-database.

Z39.50

Z39.50 benytter i utgangspunktet den lettere kryptiske RPN-syntaksen, men ved hjelp av PHP/YAZ er det mulig å gjøre en "oversetting" (se http://no.php.net/yaz_ccl_conf), slik at man kan bruke den noe mere brukervennlig CCL-syntaksen. Følgende "mapping" er gjort mellom Bib1-attributter og CCL-kvalifikatorer:

Forklaring	CCL-kvalifikator	Bib1-attributt
Tittel	ti	1=4
Klassifikasjon (dewey)	kl	1=13
Forfatter	fo	1=1003
År	år	1=31
Språk	sp	1=54
Emneord	eo	1=21
ISBN	is	1=7
Tittelnummer	tnr	1=12

Dette innebærer at når vi søker etter "eo=\$query Reisehåndbøker", slik det er beskrevet ovenfor, avgrenser vi egentlig med Bib1-attributtet "1=21".

Sortering

Standard sortering er synkende etter utgivelsesår, men brukeren kan også velge å sortere på tittel. For begge valgene er det også mulig å velge stigende eller synkende sortering.

Sorteringen utføres som en del av XSLT-transformasjonen, siden sortering ikke er tilgjengelig i noen av de implementasjonene av SRU og Z39.50 vi har søkt mot.

Sortering på årstall

En del årstall er ikke angitt som rene tall, men med tilføyelser (i tråd med katalogiseringsreglene), feks "cop. 2005" eller "[2006]". Disse ekstra tegnene fjernes fra verdien i MARC-felt 260\$c, før sorteringen gjennomføres:

```
<xsl:sort select="translate(datafield[@tag=260]/subfield[@code='c'],  
'cop.[]', '')" data-type="number" order="{ $order }"/>
```

Verdien av \$order kan være "ascending" eller "descending".

Sortering på tittel

Her sorteres det først på hovedtittel (MARC-felt 245\$a), deretter på undertittel (245\$b) og til slutt på utgivelsesår (260\$c) der ekstra tegn er fjernet, slik det er beskrevet ovenfor:

```
<xsl:sort select="datafield[@tag=245]/subfield[@code='a']" data-type="text"
order="{ $order }"/>
<xsl:sort select="datafield[@tag=245]/subfield[@code='b']" data-type="text"
order="{ $order }"/>
<xsl:sort select="translate(datafield[@tag=260]/subfield[@code='c'],
'cop.[]', '')" data-type="number" order="descending"/>
```

Moduler

Moduler - også kjent som boksene til høyre for hovedtrefflista - er implementert i filer som ligger i mappa "api", og som har navn på formen "mod.MODUL.php", der MODUL fungerer som en slags identifikator for modulen. Modulene kan slås av og på og ellers konfigureres fra fila config.php, jfr [kapitlet om tilpasninger](#).

Det er variabelen "enabled" som bestemmer om en modul/boks vises eller ikke. Denne variabelen kan ha to verdier: "true" eller "false". Dlik ser det ut når en modul er slått på:

```
$config['modules']['travel'] = array(
    'enabled' => true,
    'title' => "Reiseskildringer",
    'limit' => 5,
);
```

Endre "true" til "false" for å slå den av:

```
$config['modules']['travel'] = array(
    'enabled' => false,
    'title' => "Reiseskildringer",
    'limit' => 5,
);
```

Når brukeren søker i reisepanleggeren returneres trefflista i et HTML-dokument, slik det er beskrevet over. Samtidig sjekker index.php opplysningene i config.php for å se hvilke moduler som er aktive. De aktive modulene blir skrevet ut som tomme "bokser" i HTML-dokumentet, men disse er gjort "usynlige" ved hjelp av CSS. Når det er klart hvilket sted brukeren er interessert i (enten fordi søket i Geonames, beskrevet i kapittelet ["Oversikt"](#), bare returnerte ett sted, eller fordi brukeren ble presentert for en liste med steder og valgte ett av dem) fyller skriptet bokser.js boksene med innhold og gjør dem synlige.

Modulene er av to hovedtyper:

Bibliografiske moduler

Dette er moduler som søker i bibliotekskatalogen, og viser treff som er relevante for det stedet brukeren søkte etter. De fleste av disse modulene er i prinstippet like, det eneste

som varierer er avgrensningene i søk. Søkeregrepene for de ulike modulene er gjengitt nedenfor. "\$query" representerer brukerens søkebegrep:

culture

- Z39.50: eo=\$query and eo=kulturhistorie
- SRU: dc.subject = \$query and dc.subject = kulturhistorie

food

- Z39.50: eo=\$query and eo=matlaging
- SRU: dc.subject = \$query and dc.subject = matlaging

language

Denne modulen skiller seg fra de andre på to måter: dels handler den ikke om land, men om språk, og dels viser den to ulike dokumenttyper. Språkene som snakkes i et land hentes ut fra Geonames, og brukeren presenteres i utgangspunktet for en liste over disse språkene. Når et språk er valgt viser modulen språkkurs og lærebøker innenfor dette språket. Følgende søkebegrep benyttes, "\$lang" representerer det språket brukeren har valgt:

- Z39.50: eo=\$lang and eo=språkkurs
- SRU: dc.subject = \$lang and dc.subject = språkkurs
- Z39.50: eo=\$lang and eo=lærebøker
- SRU: dc.subject = \$lang and dc.subject = lærebøker

stories

- Z39.50: eo=\$query fortellinger
- SRU: dc.subject = \$query and dc.subject = fortellinger

travel

- Z39.50: eo=\$query and eo=Geografi og reiser
- SRU: dc.subject = \$query and dc.subject = Geografi og reiser

Andre moduler

debug

Dette er en liten modul som har en eneste oppgave: å vise hvilke data som er tilgjengelige for modulene. Slå den på og gjør noen søk for å se dataene. Skal ikke være slått på i et system i produksjon!

map

Viser kart fra Google. For at dette skal fungere trenger man å legge inn en "Google Maps API key" i config.php (jfr [kapitlet om tilpasninger](#)). En slik nøkkel får man herfra: <http://code.google.com/intl/en/apis/maps/>

Når man oppretter nøkkelen må man oppgi hvilket domene den skal gjelde for (feks mittbibliotek.no).

Googles kart-API fungerer på følgende måte:

1. man inkluderer et JavaScript på siden der man ønsker å vise et kart:

```
<script src="http://maps.google.com/maps?file=api&v=2&key=[din API-nøkkel]&sensor=false" type="text/javascript"></script>
```

(For Reiseplanleggeren sin del oppgis API-nøkkelen i fila config.php, slik at man ikke trenger å endre på HTML-sidene for å få det hele til å fungere.)

2. På siden der kartet skal vises setter man in et div-element med id = "map" og kaller opp funksjonen map() som "importeres" fra JavaScriptet man inkluderte ovenfor.

map() tar følgende argumenter:

- lengdegrad
- breddegrad
 - Disse to punktene definerer midtpunktet i det kartutsnittet som skal vises
- bredde
- høyde
 - Disse to argumentene definerer størrelsen på kartutsnittet
- zoom
 - Definerer hvor mye kartutsnittet skal være "zoomet inn", dvs hvor mye av omgivelsene til midtpunktet som skal vises.

snl

Gjør et søk i Store norske leksikon (<http://www.snl.no/>) og viser tittel, lenke og ingress for det første treffet.

SNL har i kommunikasjon med PODE-prosjektet beskrevet sitt API slik:

Man søker i leksikonet med følgende url:

```
http://www.snl.no/.search?query=test&format=xml&size=0&y=0
```

query er spørreordet, x og y er antall ønskede svar og startpunkt, format er hvilket format dere ønsker resultatet i. Vi støtter formatene html, xml og json.

Strengen kan også ta parameteren authorized som kan være 0 eller 1, avhengig av om man vil søke etter autorisert innhold, uautorisert, eller begge deler (i siste tilfellet utelater man parameteren).

Argumentene x og y har ikke sett ut til å fungere, og vi har i ettertid fått vite at argumentet "size" også kan benyttes, for å bestemme antall treff som returneres.

I snl-modulen har vi valgt å sende brukerens søkebegrep videre til SNL, slik at dersom brukeren søker etter "london" henter vi følgende URL fra SNL:

```
http://www.snl.no/.search?query=london&format=json&size=1&y=0
```

Verdien for "size" kan styres fra config.php, i dette tilfellet er den satt til 1.

Resultatene returneres i JSON-format.

weather

Henter værdata fra yr.no (<http://www.yr.no/>) ved hjelp av "locationforecast"-APIet (<http://api.yr.no/weatherapi/locationforecast/1.6/documentation>), basert på lengde- og breddegrad.

APIet henter data fra denne basis-URLen

```
http://api.yr.no/weatherapi/locationforecast/1.6/?
```

Denne URLen tar to parametere:

- Lengdegrad
- Breddegrad

Feks slik:

```
http://api.yr.no/weatherapi/locationforecast/1.6/?lat=10;lon=10
```

Data om temperatur, vind og værsymbol som viser skyer, bedbør osv returneres i XML-format.

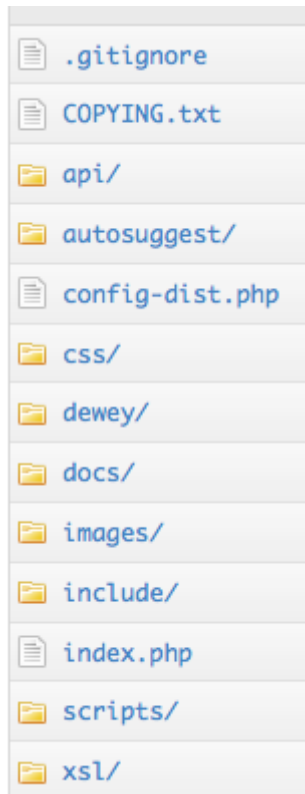
Referanseliste

- Apache <http://httpd.apache.org/>
- Git <http://git-scm.com/>
- Geonames <http://www.geonames.org/>
- Google Maps API <http://code.google.com/intl/en/apis/maps/>
- Norske biblioteksystem som støtter protokollene SRU og/eller Z39.50 http://wiki.biblab.no/index.php/Norske_Z39.50_datatilbydere og http://wiki.biblab.no/index.php/Norske_SRU_datatilbydere
- PHP <http://no.php.net/>
- PHP/YAZ <http://no.php.net/yaz>, <http://no.php.net/manual/en/yaz.installation.php>
 - Konfigurasjon http://no.php.net/yaz_ccl_conf
- Store norske leksikon <http://www.snl.no/>
- yr.no <http://www.yr.no/>

Vedlegg: Katalogstruktur og eksempelkode

Katalogstruktur

Øverste nivå



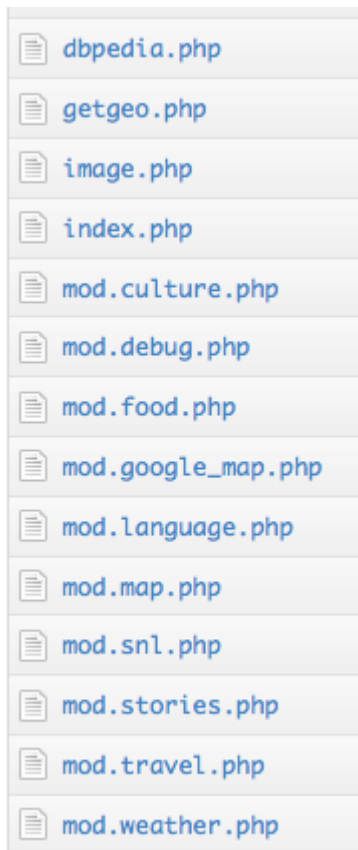
.gitignore er en fil som benyttes av Git. Den er satt opp slik at Git ignorerer fila config.php, noe som medfører at endringer her ikke vil bli plukket opp av versjonskontrollsystemet.

COPYING.txt inneholder GPL-lisensen i fulltekst.

config-dist.php må flyttes til config.php under installasjonen, jfr avsnittet om [config-dist.php](#).

index.php er hovedfila for hele prosjektet.

api/



Her ligger filene som kalles opp fra JavaScriptet bokser.js, for å fylle inn innholdet i boksene på høyre side av reiseplanleggeren.

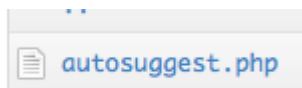
dbpedia.php stammer fra et ikke fullført forsøk på å hente inn data fra dbpedia. Fila beholdes for å kunne tjene som grunnlag for eventuelle senere forsøk på å få dette på plass.

getgeo.php henter geografiske data fra Geonames og returnerer dem som JSON til bokser.js. Se nærmere beskrivelse under [Oversikt](#).

image.php sørger for visning av bokbilder. Dette skriptet tar et ISBN-nummer som argument og forsøker først å finne bilder hos Bokkilden, deretter hos Open Library.

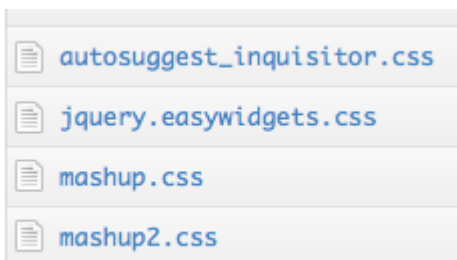
index.php er et svært enkelt skript som inkluderer de underliggende modulene (mod.MODUL.php) ved behov, og returnerer data i HTML-format til bokser.js. Se avsnittet om [Moduler](#).

autosuggest/



Her ligger det ett skript som sørger for data til autosuggest-funksjonen.

css/



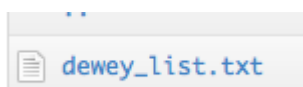
autosuggest_inquisitor.css inneholder formatering knyttet til autosuggest-funksjonen.

jquery.easywidgets.css besørger utseende for boksene på høyre side.

mashup2.css er hovedstilarket for reiseplanleggeren.

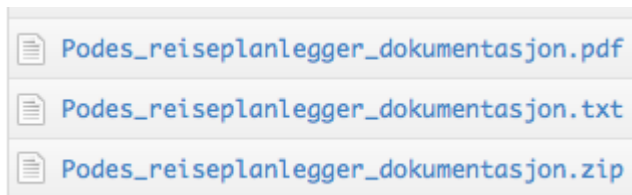
mashup.css er ikke i bruk, men er beholdt av historiske årsaker.

dewey/



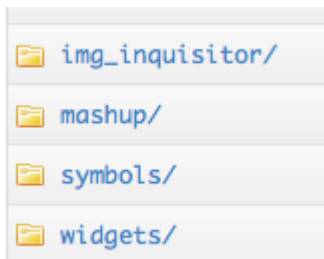
Her ligger et uttrekk fra katalogen som inneholder stedsnavn og tilknyttede Dewey-nummer. Dette danner grunnlaget for autosuggest-funksjonen, og ble i tidligere versjoner også brukt i forbindelse med søking. Se avsnittet om [Autosuggest](#).

docs/



Her ligger dokumentasjonen til reiseplanleggeren i formatene PDF, ren tekst og HTML (pakket som en ZIP-fil).

images/



Mappa inneholder 4 undermapper, med følgende innhold:

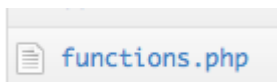
img_inquisitor/: knyttet til autosuggest

mashup/: diverse bilder

symbols/: vær-symboler som benyttes i boksen med værinformasjon fra yr.no. Symbolene er hentet fra <http://www.yr.no/informasjon/1.1940495>

widgets/: ikoner knyttet til boksene

include/



Fila functions.php inneholder mye av den sentrale funksjonaliteten i reiseplanleggeren, og inkluderes av både index.php og de ulike modulene under api/.

scripts/



Her ligger filer med JavaScript-funksjonalitet.

bokser.js sørger for å hente geografiske data via api/getgeo.php.

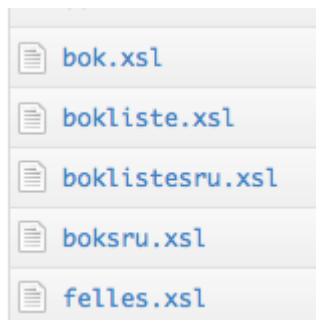
De to AutoSuggest-filene besørger autosuggest-funksjonaliteten. Disse er hentet fra http://www.brandspankingnew.net/archive/2007/02/ajax_auto_suggest_v2.html

googlemaps.js inneholder funksjoner knyttet til

Google Maps og [kart-modulen](#).

jquery.easywidgets.min.js inneholder funksjonalitet knyttet til boksene på høyre siden av reiseplanleggeren, og er hentet fra <http://plugins.jquery.com/project/easywidgets>

xsl/



MARXML transformeres til XHTML for presentasjon i reiseplanleggeren med disse filene.

bok.xsl: postvisning for Z39.50

bokliste.xsl: treffliste for Z39.50.

boksruxsl: postvisning for SRU

boklistesru.xsl: treffliste for SRU

felles.xsl inneholder en del funksjonalitet som er felles for de andre transformasjonene.

api/mod.food.php

```
<?php

/*

Copyright 2009 ABM-utvikling

This file is part of "Podes reiseplanlegger".

"Podes reiseplanlegger" is free software: you can redistribute it and/or
modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

"Podes reiseplanlegger" is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with "Podes reiseplanlegger". If not, see <http://www.gnu.org/licenses/>.

Source code available from:
http://github.com/pode/reiseplanlegger/

*/

include_once('../config.php');

if (!$config['modules']['food']['enabled']) {
    exit;
}

include_once('../include/functions.php');

/*
Henter ut reiseskildringer fra katalogen, basert pp land
*/

if (!empty($_GET['country']) && !empty($_GET['bib'])) {

    if (get_type($_GET['bib']) == 'z39.50') {
        if ($zresult = z_search("eo={$_GET['country']} and eo=matlaging",
$config['modules']['food']['limit'])) {
            echo $zresult;
        } else {
            echo $config['msg']['zero_hits'];
        }
    } else {
        if ($result = sru_search("dc.subject = {$_GET['country']} and dc.subject
= matlaging", $config['modules']['food']['limit'])) {
```

```

    echo $result;
  } else {
    echo $config['msg']['zero_hits'];
  }
}

}

?>

```

api/mod.map.php

```
<?php
```

```
/*
```

Copyright 2009 ABM-utvikling

This file is part of "Podes reiseplanlegger".

"Podes reiseplanlegger" is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

"Podes reiseplanlegger" is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with "Podes reiseplanlegger". If not, see <<http://www.gnu.org/licenses/>>.

Source code available from:
<http://github.com/pode/reiseplanlegger/>

```
*/
```

```
include_once('../config.php');
```

```
if (!$config['modules']['map']['enabled']) {
    exit;
}
```

```
echo "\t<div id=\"map\"></div>\n";
getMap($_GET['lat'], $_GET['lon'], "280", "280", "12");
```

```
/*
```

funksjon som skriver kart fra Google Maps.

Parametere:

\$lat og \$lon angir lengde- og breddegrad kartet skal vise.
 \$width og \$height forteller hva dimensjonene på kartet skal være.
 \$zoom angir hvilket zoom-nivå kartet skal ha i utgangspunktet
 */

```

function getMap($lat, $lon, $width, $height, $zoom)
{

echo "\t\t<script type=\"text/javascript\">
map($lat,$lon,$width,$height,$zoom);
</script>\n";

?>
<label for="informasjon">Informasjon</label>
<input id="informasjon" type="checkbox"
onclick="switchLayer(this.checked,wiki)" />
<label for="bilder"> Bilder</label>
<input id="bilder" type="checkbox" onclick="switchLayer(this.checked,pano)"
/>
<label for="filmer">Filmer</label>
<input id="filmer" type="checkbox" onclick="switchLayer(this.checked,tube)"
/>
<?php
}

?>

```

api/mod.snl.no

```

<?php

/*

Copyright 2009 ABM-utvikling

This file is part of "Podes reiseplanlegger".

"Podes reiseplanlegger" is free software: you can redistribute it and/or
modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

"Podes reiseplanlegger" is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with "Podes reiseplanlegger". If not, see <http://www.gnu.org/
licenses/>.

Source code available from:
http://github.com/pode/reiseplanlegger/

*/

include_once('../config.php');
if (!$config['modules']['snl']['enabled']) {
    exit;
}
include_once('../include/functions.php');

```

```

/*
Viser lenker til artikler fra SNL

Info fra SNL:

Man søker i leksikonet med følgende url:
http://www.snl.no/.search?query=test&format=xml&size=0&y=0
query er spørreordet, x og y er antall ønskede svar og startpunkt,
format er hvilket format dere ønsker resultatet i. Vi støtter
formatene html, xml og json.

Strengen kan også ta parameteren authorized som kan være 0 eller 1,
avhengig av om man vil søke etter autorisert innhold, uautorisert,
eller begge deler (i siste tilfellet utelater man parameteren).

NB! Det ser ikke ut som x- og y-parameterne i URLen til SNL funker!
Ny info: parameteren size skal avgrense antall treff

*/

if (!empty($_GET['q'])) {

    $data = json_decode(file_get_contents("http://www.snl.no/.search?query="
    . $_GET['q'] . "&format=json&size=" . $config['modules']['snl']['limit'] .
    "&y=0"));

    if ($data) {
        foreach ($data->result->list as $item) {
            echo('<p><a href="http://snl.no/' . $item->link . '">' . $item->title
            . '</a>', ' . strip_tags($item->shortview) . '<br /><a href="http://snl.no/'
            . $item->link . '">Les mer i Store norske leksikon</a></p>');
        }
    }
}

?>

```

api/mod.weather.php

```

<?php

/*

Copyright 2009 ABM-utvikling

This file is part of "Podes reiseplanlegger".

"Podes reiseplanlegger" is free software: you can redistribute it and/or
modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

"Podes reiseplanlegger" is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of

```

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with "Podes reiseplanlegger". If not, see <<http://www.gnu.org/licenses/>>.

Source code available from:
<http://github.com/pode/reiseplanlegger/>

```
*/  
  
include_once('../config.php');  
  
if (!$config['modules']['weather']['enabled']) {  
    exit;  
}  
  
//henter været basert på lengde-, breddegrad og tidssone  
$weather = getWeather($_GET['lat'], $_GET['lon'], $_GET['timeZone']);  
  
//skriver ut været  
echo "\t\t\tVind: ".$weather['windSpeed']."<br />\n";  
echo "\t\t\tTemperatur: ".$weather['temperature']." &deg; C<br /><br />\n";  
echo "\t\t</td>\n";  
echo "\t\t<td>\n";  
echo "\t\t\t&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~\n";  
src=\"\"".$weather['symbol']."' alt=\"$v&aelig;rsymbol\"/><br />\n";  
echo "\t\t</td>\n";  
echo "\t\t</tr>\n";  
echo "\t</table>\n";  
  
//skriver ut lenke til yr slik lisensen krever  
echo "V&aelig;rvarsel fra <a href=\"$http://www.yr.no/\">yr.no</a>, levert  
av Meteorologisk institutt og NRK<br /><br />\n";  
  
/*  
funksjon som henter returnerer vindinfo, tempinfo og v&aelig;rsymbol  
for gitt lengde- og breddegrad ($lat og $lon). $timeZone brukes  
for å bestemme om det er natt eller dag  
*/  
function getWeather($lat, $lon, $timeZone)  
{  
    //stammen i URL'en der symbolene befinner seg  
    $symbolPath = "images/symbols/";  
  
    /*  
    arrayer som forteller hvilket symbol som tilsvarer hvilket i  
    XML'en. alle filer som slutter på 'd' har sol, de med 'n'  
    viser måne  
    */  
    $symbolsDay = array(1 => "01d.png",  
        2 => "02d.png",  
        3 => "03d.png",  
        4 => "04.png",  
        5 => "05d.png",  
        6 => "06d.png",  
        7 => "07d.png",
```

```

8 => "08d.png",
9 => "09.png",
10 => "10.png",
11 => "11.png",
12 => "12.png",
13 => "13.png",
14 => "14.png",
15 => "15.png",
16 => "16.png",
17 => "17.png",
18 => "18.png",
19 => "19.png");

$symbolsNight = array(1 => "01n.png",
2 => "02n.png",
3 => "03n.png",
4 => "04.png",
5 => "05n.png",
6 => "06n.png",
7 => "07n.png",
8 => "08n.png",
9 => "09.png",
10 => "10.png",
11 => "11.png",
12 => "12.png",
13 => "13.png",
14 => "14.png",
15 => "15.png",
16 => "16.png",
17 => "17.png",
18 => "18.png",
19 => "19.png");

//henter XML med varsel for gitt lengde- og breddegrad fra yr
$xml = simplexml_load_file("http://api.yr.no/weatherapi/locationforecast/
1.6/?lat=$lat;lon=$lon");

//henter ut hvilken time det er basert på tidssone
date_default_timezone_set($timeZone);
$hour = date("G")+0;

//henter ut windSpeed-nodene med XPath
$windData = $xml->xpath("/weatherdata/product/time/location/windSpeed");

//henter ut attributter i første windSpeed-node
foreach ($windData[0]->attributes() as $a => $b)
{
//lager attributten med navn 'name' i $windSpeed
if ($a=='name')
$windSpeed = $b;
}

//tilsvarende windSpeed
$temperatureData = $xml->xpath("/weatherdata/product/time/location/
temperature");

foreach ($temperatureData[0]->attributes() as $a => $b)
{

```

```

if ($a=='value')
$temperature= $b;
}

//tilsvarende windSpeed
$symbolData = $xml->xpath("/weatherdata/product/time/location/symbol");

foreach ($symbolData[0]->attributes() as $a => $b)
{
if ($a=='number')
$symbol = $b;
}

/*
velger symbol på grunnlag av hvilken tid på dagen det er.
her har vi valgt å vise sol hvis timen er fra og med 6 til
og med 19
*/
if ($hour<20 and $hour>=6)
{
$imageLink = $symbolPath.$symbolsDay[$symbol+0];
}
else
{
$imageLink = $symbolPath.$symbolsNight[$symbol+0];
}

/*
returnerer en array med windSpeed(tekst), temperature(tall)
og symbol(bildelink)
*/
return array("windSpeed" => $windSpeed, "temperature" => $temperature,
"symbol" => $imageLink);
}

?>

```

scripts/bokser.js

```

/*

Copyright 2009 ABM-utvikling

This file is part of "Podes reiseplanlegger".

"Podes reiseplanlegger" is free software: you can redistribute it and/or
modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

"Podes reiseplanlegger" is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License

```

along with "Podes reiseplanlegger". If not, see <<http://www.gnu.org/licenses/>>.

Source code available from:
<http://github.com/pode/reiseplanlegger/>

```
*/

// Sjekk om det er gitt noen verdi for "place" i URLen (query string)
if (getQueryVariable('place')) {

// Dette er en jQuery funksjon som automatisk kjøres når hele siden er
lastet inn
// (dvs, egentlig når DOMen er klar)
$(function() {

    // Gjør boksene til "widgets"
    // useCookies : true gjør at det brukes cookies for å huske tilstanden
    til boksene
    // mellom sesjoner
    $.fn.EasyWidgets({

        behaviour : {
            useCookies : true
        },

        i18n : {
            editText : '',
            closeText : '',
            collapseText : '',
            cancelEditText : '',
            extendText : ''
        }
    });

    // Vis boksen som inneholder navn på sted, evt mulighet for å velge sted
    $("#place").css({'visibility' : 'visible'});

    // Hent inn verdier fra URLen (query string) Denne funksjonen er definert
lenger ned i fila
    var place = getQueryVariable('place');
    // var placeEng = place;
    var geoId = getQueryVariable('geoId');

    // Bruk Google Translate for å oversette navnet som vi antar er på norsk,
til engelsk
    // (Det funker å søke med norske navn mot hoved-fila fra GeoNames, men
ikke mot fila
    // som gir oss koblingen mellom land og hovedstad.)
    // google.language.translate(place, "no", "en", function(result) {
    // if (!result.error) {
    // placeEng = result.translation;
    // }
}
```



```

// });

// Hent nødvendige geodata fra serveren
// Ved suksess vil getgeo.php returnere et array som inneholder et
element "result" som vil
// være antallet treff. Vi bruker dette for å bestemme videre gang i
skriptet
$.getJSON("api/getgeo.php", { geoId: geoId, place: place },
function(json){

    // Fjern innholdet i boksen som viser sted/tid
    $("#place").find(".widget-content").text("");

    if (json == 0) {

        // INGEN TREFF, skriver ut en feilmelding

        $("#place").find(".widget-content").append("<p>Beklager, 0 treff.
Pr&oslash;v &aring; s&oslash;ke etter et annet stedsnavn.</p>");

        // TODO: Kunne man fått til å foreslå alternative skrivemåter?

    } else if (json.length == 1) {

        // ETT TREFF - lager bokser med info om stedet/landet

        // Vi vil bare ha data fra det første og eneste stedet
        var place = json[0];

        // Hent ut koordinater
        var lat = place.lat;
        var lon = place.lng;

        if (place.fcode == 'PCLI' || place.fcode == 'PCLD') {
            // Vi har et land
            $("#place").find(".widget-content").append("<h3>" + place.name + "</h3>");
            $("#place").find(".widget-content").append("<p>Hovedstad: " +
            place.placeInfo.capital + "</p>");
            $("#place").find(".widget-content").append("<p>Befolkning: " +
            addCommas(place.placeInfo.population) + "</p>");
            $("#place").find(".widget-content").append("<p>Valuta: " +
            addCommas(place.placeInfo.currencyCode) + "</p>");
            // Sett koordinatene til hovedstadens koordinater
            lat = place.placeInfo.capital_long.lat
            lon = place.placeInfo.capital_long.lng
        } else {
            // Vi har et sted
            $("#place").find(".widget-content").append("<h3>" + place.name + ", " +
            place.countryName + "</h3>");
            $("#place").find(".widget-content").append("<p>Befolkning: " +
            addCommas(place.population) + "</p>");
            $("#place").find(".widget-content").append("<p>Valuta: " +
            addCommas(place.placeInfo.currencyCode) + "</p>");
        }

        // Lag en liste med navn på språkene
        var langs = '';

```

```

jQuery.each($(place.placeInfo.languages_long), function() {
langs = langs + this + ",";
});

// Gjør widgetene synlige
$(".widget").css({'visibility' : 'visible'});

// Gå igjennom alle widgetene og legg til innhold
jQuery.each($(".widget"), function() {
var this_widget = this;
target = this_widget.id.replace(/widget_/g, "");

// Vi sender den samme informasjonen til modulene, uavhengig av hva de skal
gjøre for noe.
// På denne måten slipper vi å vite noe om hver enkelt modul før vi kaller
dem opp.
$.get("api/index.php", { mod: target,
lat: lat,
lon: lon,
place: place.name,
country: place.countryName,
langs: langs,
q: getQueryVariable('place'),
bib: getQueryVariable('bib')
},
function(data){
$("#" + this_widget.id).find(".widget-content").text("");
$("#" + this_widget.id).find(".widget-content").append(data);
// Sørg for å aktivere eventuelle trekkspill
$(".trekkspill").accordion({ autoHeight: false, active: false });
});

});

} else {

// FLERE TREFF - vis liste med mulighet for å velge hvilket sted som
var ment

$("##place").find(".widget-content").append('<h3>Velg sted</h3>');
var place = "place=" + getQueryVariable('place');
var sort = "&sortBy=" + getQueryVariable('sortBy');
var order = "&order=" + getQueryVariable('order');
var bib = "&bib=" + getQueryVariable('bib');
jQuery.each(json, function() {

var name = this.name;
var cname = this.countryName;
var geoId = "&geoId=" + this.geonameId;
var adminname = '';
if (this.adminName1) {
adminname = ", " + this.adminName1;
}

placename = '';
if (this.fcode == 'PCLI') {
// Land
placename = this.countryName;

```

```

    } else {
    // Sted
    placename = name + adminname + ", " + cname;
    }

$("#place").find(".widget-content").append('<p><a href="?' + place + geoId
+ sort + order + bib + '">' + placename + '</a></p>');

    });

}
});

});

}

// From: http://www.mredkj.com/javascript/nfbasic.html
function addCommas(nStr) {
nStr += '';
x = nStr.split('.');
x1 = x[0];
x2 = x.length > 1 ? '.' + x[1] : '';
var rgx = /(\d+)(\d{3})/;
while (rgx.test(x1)) {
x1 = x1.replace(rgx, '$1' + '.' + '$2');
}
return x1 + x2;
}

// From: http://www.webdeveloper.com/forum/showthread.php?t=166692
function getQueryVariable(variable)
{
var query = window.location.search.substring(1);
var vars = query.split("&");
for (var i=0; i<vars.length; i++)
{
pair = vars[i].split("=");
if (pair[0] == variable)
{
str_arr = pair[1].split("+");
return str_arr.join(" ");
}
}
return "";
}

```

