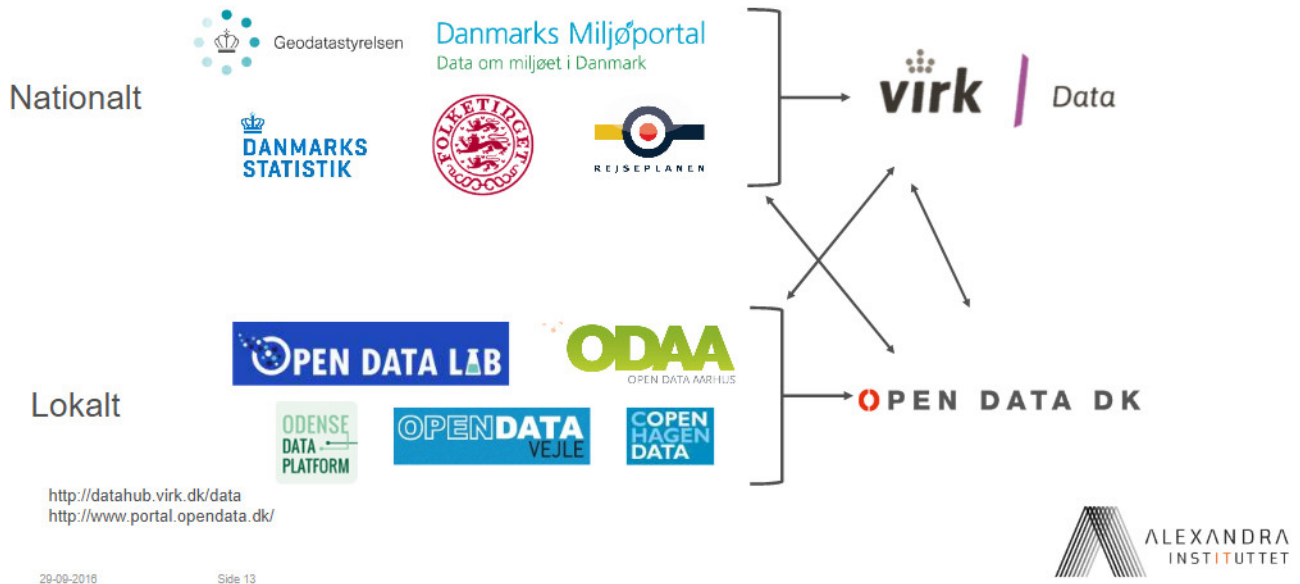


## Linkliste over webservices

Se nederst i dokumentet her: 3 eksempler på kode til at hente data fra en webservice (XMLHttpRequest, Fetch og jQuery Ajax).

## UDBUD AF ÅBNE DATA I DANMARK



## Mashup!

- [Google - best mashup map](#)
- [9 of the Best Web Mashups That Once Ruled Online](#)

## TJEK DENNE

[A query language for your API - GraphQL](#) ... set på UmbracoFestival okt 2018 - se evt. slides (hvis jeg fik nogen)

GOOGLEMAP (okt 2018) [Dansk it-virksomhed lancerer gratis alternativ til Google Maps](#)

## Når du bruger en webservice - vær opmærksom på CORS

**CORS (Cross-Origin Resource Sharing)** handler kort fortalt om sikkerhed og kontrol i forhold til det at dele "ressourcer" på tværs af forskellige domæner (oprindelser/origins). Som når man fra origin/domænet [localhost](#) eller [mitdomæne.dk](#) hente ressourcer/data fra et andet origin/domæne fx [swapi.co](#) (StarWars-data). Når man med JavaScript kalder en webservice - fx de der StarWars-data - så sker det jo hos klienten/brugeren af dit website.

Et sådan setup indebærer visse risici, og derfor kan det med StarWarsAPI'et kun lade sig gøre, fordi gutterne bag swapi.co har enabled/aktiveret CORS med `Access-Control-Allow-Origin: *` Stjernen betyder, at alle origins/domæner må hente data fra webservicen. Man kan begrænse adgangen ved at sætte nogle andre regler end `*`. Sådan lidt kort forklaret. Læs mere her:

- [What is CORS](#)
- [Cross-Origin Resource Sharing \(CORS\)](#)

- [Enabling Cross-Origin Requests in ASP.NET Web API 2](#)

Og test evt. selv CORS her: [test-cors.org](http://test-cors.org)

Test fx med <https://swapi.co/api/people/1> ... du får OK som svar (alle må bruge løs). Og test så med <https://live.midttrafik.dk/getbuses.php?lat=56.21325964347735&lon=10.68502922058109&radius=10000> ... svarer med Error og 0/nul.

## Et par eksempler på websites som lever af at benytte (andres) webservices

- [Den norske vejrtjeneste Yr](#) benytter **Geonames** og **webcams.travel**. Og leverer så vidt jeg kan se selv en webservice - **Free weather map**. Tjek selv et stykke nede på Yr-siden.
- [DinGeo](#) viser en masse data for en specifik adresse (tjek selv med din egen adresse fx). Den trækker på en masse offentlige tilgængelige datakilder. Læs selv om det på siden "Om DinGeo".
- [AutoUncle](#) trækker også på forskellige webservices - fx Bilbogen, nummerpladeopslag, Motorregistret mv.

## API

[API Guide](#)

## Lidt tuts

- [Getting Started with ASP.NET Web API 2 \(C#\)](#)
- [Enabling Cross-Origin Requests in ASP.NET Web API 2](#)
- [See ASP.NET Web API in action with these samples](#)
- [XML Web Services - W3Schools](#)
- [Living Standard — Last Updated 12 January 2018](#)

## Diverse

- [Hvor lang må en url/query string max være?](#)
- [W3Schools - AJAX - Server Response - om xmlhttprequest](#)

## Link til og eksempler på webservices

## Open Data - Aarhus, Danmark, EU, Verden ....

- [OpenDataAarhus - dataset](#)
- [Det sker i Aarhus](#)
- [Copenhagen Data](#)
- [Open Data DK](#)
- [mauran/API-Danmark](#)
- [European Union Open Data Portal \(EU ODP\)](#) ... webinar om brug af disse data [EU Datathon 2017 - Webinar 3 \(Financial transparency system, Whoiswho, ...\)](#)
- [Open Data Handbook - Resources](#)

Årlig konkurrence i EU (med præmier) om brug af EU Open Data: [EU Datathon competition](#) - [Datasets by subject](#)

## OPEN DATA SCHOOL

- [Virk Data Open Data School](#)

## CARTO

- [Carto ... Carto documentation](#)
- [CARTO.js](#)

## GeoNames

The GeoNames geographical database covers all countries and contains over eleven million placenames that are available for download free of charge.

- [GeoNames](#)
- [GeoNames WebServices overview](#)
- [DK - countryInfo](#) og [med mbrvidendjurs-account](#)
- [Geonames opret bruger](#)
- [Opslag ex - postalCodeSearchJSON](#)
- [Opslag ex - postalCodeSearchJSON](#)
- [Opslag ex - postalCodeSearchJSON](#)
- [Earthquakes](#)

## DAWA

Danmarks Adressers Web API (DAWA) udstiller data og funktionalitet vedrørende Danmarks adresser, adgangsadresser, vejnavne samt postnumre. DAWA anvendes til at etablere adressefunktionalitet i it-systemer. Målgruppen for nærværende website er udviklere, som ønsker at indbygge adressefunktionalitet i deres it-systemer. DAWA er en del af AWS Suiten.

- [Danmarks Adressers Web API - DAWA](#)
- [Generelt om Web API'et](#)
- [Adresseopslag](#)

## Digitaliser.dk

Digitaliser.dk er en social netværksplatform for erfaringsudveksling og videndeling om offentlig digitalisering af Danmark, hvor du blandt andet finder vejledninger, software eller driftsinformationer.

- [Digitaliser.dk - forsiden](#)

Alt indhold på Digitaliser.dk ligger i grupper. Grupperne har udgangspunkt i en fælles interesse om digitalisering i det offentlige, fx et projekt, et driftssamarbejde eller noget helt andet. Alle kan oprette en gruppe på Digitaliser.dk.

Scroll ned til fx "**Teknisk indhold**" - her er det opdelt i Datakataloget, DATA, XML, SOFTWARE. Tjek fx [xml Velkommen til Datakataloget](#)

- Grupper: [Grupper](#)

## Film

- [IMDb - Internet Movie Database](#)
- [IMDb - interface \(api\)](#)
- [OMDb - The Open Movie Database](#)
- [OMDb - koster 1\\$ pr. måned for billigste API-key](#)
- [Åbne data - Brug Filmdatabasens indhold](#)
- [Dokumentation DFI API](#)
- [Ex på webservice-url - der bliver spurgt efter brugernavn og password ...](#) Brugernavn: `F005936` og Password: `JRbT1fWVMH0bm3n`

**Fx - film som indeholder ordet "børn" og som er fra 2008-2010:**

```
http://nationalfilmografien.service.dfi.dk/movie.svc/json/list?titlecontains=børn&startyear=2008&endyear=2010
```

**Kan tage følgende parametre fx:**

```
titlecontains={titlecontains}&titlestartswith={titlestartswith}&freesearch={freesearch}&category={category}
```

## Rapport om brug af åbne data

Blot til info og almen interesse ... læs her: [Afrapportering - Analyse af efterspørgsel og markedstendenser indenfor offentlige data - juni 2017](#)

## OpenWeatherMap

- [OpenWeatherMap](#)

Fx `api.openweathermap.org/data/2.5/weather?lat=35&lon=139&APPID=[din-app-key]`

Også med city: Parameters: q city name and country code divided by comma, use ISO 3166 country codes ...:

```
http://api.openweathermap.org/data/2.5/weather?q=aarhus,da&APPID=[din-app-key]
```

Tilføj `units=metric` for at få celcius ... læs/vis her:

- [OpenWeatherMap Units format](#)

Bemærk også icon-id i resultatet/json. Tjek owm egen ikoner - så er det tydeligt, hvordan icon-id skal bruges fx:

```
var vejrikon = "<img src='http://openweathermap.org/img/w/" + data.weather[0].icon + ".png' />";
```

Tjek fx denne fikse måde at (gen)bruge werbserviceen på:

```
let apis = {
  currentWeather: {
    api: "http://api.openweathermap.org/data/2.5/forecast/daily?lat=",
    parameters: "&mode=xml&units=metric&cnt=6&APPID=/*api key*/",
    url: (lat, lon) => {
      return apis.currentWeather.api + lat + "&lon=" + lon +
        apis.currentWeather.parameters
    }
  }
};

function getCurrentLoc() {
  return new Promise((resolve, reject) => navigator.geolocation
    .getCurrentPosition(resolve, reject))
}

function getCurrentCity(location) {
  const lat = location.coords.latitude;
  const lon = location.coords.longitude;
  return fetch/apis.currentWeather.url(lat, lon))
    .then(response => response.json())
    .then(data => console.log(data))
}

getCurrentLoc()
  .then( coords => getCurrentCity(coords))
```

## ThingSpeak

- [ThingSpeak - vælg fx Channels](#) ... IOT sender fra forskellige kanaler (link fra Michael) ...
- [ThingSpeak Apps](#) ...
- [MrCoffeePlant](#) ... og tjek webservice2017, hvor der er lavet en side, som er starten på en visualisering af temp, humidity mv. [json-data her](#)
- [Hamster Wheel](#)

HUSK ... du kan søge på tags. Og når du har fundet nogle spændende data, så tjek lige, hvornår de sidst er opdaterede - visse channels har stået stille i nogle år, og det er der jo ikke så meget sjov ved.

## EnergiNet i Danmark

Danmarks EnergiNet stiller data til rådighed - visse er gratis, andre kræver login: Tjek dem ud her:

- [Energi Data Service](#)

Tjek fx her [Electricity Balance Non-Validated Data](#) og klik på knappen "DATA API" ...

## Data til kort/maps

- [OpenStreetMap](#)
- [OpenLayers](#)
- [Open Layers - tutorials](#)
- [Open Layers - dokumentation og FAQ](#)
- [Open Layers - API](#)
- Download masser af eksempler her [Downloads for the v5.2.0 release](#) ... hent versionen "v5.2.0.zip Includes all of the above plus examples, API docs, and sources."
- [wiki OpenSeaMap](#) og [OpenSeaMap - die freie Seekarte](#) og dybder? [Kategorie:Depth-DB](#)

## - Hvad er det her? [Ocean Basemap](#)

- [MapQuest Developer](#)
- [Datasæt med danske postnumre – klar til dit mashup helt kvit og frit](#)

Google maps (kræver desværre en billing-plan/kreditkortoplysninger for at kunne benyttes fra foråret 2018)

- [Google Maps JavaScript API - GUIDE/tutorial](#)

## Rejseplanen

- [Rejseplanen Labs](#) ...

Fx:

```
Json:
http://xmlopen.rejseplanen.dk/bin/rest.exe/location?input=ebeltoft&format=json

Xml:
http://xmlopen.rejseplanen.dk/bin/rest.exe/location?input=ebeltoft

http://xmlopen.rejseplanen.dk/bin/multiDepartureBoard?id1=8600626&id2=5548&date=23.01.18&time=22:02&useToG

http://xmlopen.rejseplanen.dk/bin/rest.exe/departureBoard?id=8600626&date=23.01.18&time=20:02
```

Pga. af CORS, kan det blive nødvendigt med en proxy ... fx [Cors Anywhere](#) ... læs fx om CORS her [How to fix CORS problems](#)

## Andre vigtige/sjove/avancerede

- [twitch Developers](#) kræver login og evt. henvisning til hjemmeside (skriv evt. bare "localhost")
- [Webcam Web Service API - worldwide](#)
- [Steam Web API Documentation](#)
- [Twitter API/Developer - Docs](#)
- [Build epic experiences with Blizzard's APIs](#)
- [Instagram API - Hello Developers](#)
- [PokéAPI - The RESTful Pokémon API](#)
- [the Riot Games API](#)
- [facebook for Developers - APIs and SDKs](#)
- [SWAPI - The Star Wars API](#)
- [Google Time Zone API](#)
- [Tronald Dump](#)

- [Hent data fra fødevaredata](#)
- [Coolrunner](#)
- [Open Data Handbook](#)
- [VIRK - bl.a. Rejseplanen](#)
- [KORTFORSYNINGEN - Webtjenesteliste](#)
- Valuta: [Fixer](#)
- Valuta-converter [Online Currency Converter](#)
- Søfartsstyrelsen (er vist ikke gratis) [AIS-data](#)
- [MediaWiki API:Main page](#)
- PostNord API [API - General Information](#) - kræver at du opretter en bruger for at du kan få den nødvendige key
- [MashApe - marketplace](#) og [RapidAPI Marketplace](#) Og en beskrivelse: [Building A Movie Quote Randomizer with Vanilla JS and Mashape API](#)

## Div værktøjer

- [Gelato - Create Super Sweet Developer Portals](#)
- [Mashape/kong ... FAQs - Mashape Marketplace](#)
- [Deliver Better APIs - The Analytics Platform for APIs, Microservices and Serverless Software](#)

## Hent data fra webservice på 3 måder

Der findes en masse varianter over disse, men her er i hvertfald 3 bud på, hvordan du kan hente data fra en webservice. De gør alle tre nøjagtigt det samme, og skal alle have en `id=resultat` i `body` fx:

```
<h1 id="resultat"></h1>
```

## XMLHttpRequest

Den gode gamle, som virker i de fleste (også ældre) browsere [XMLHttpRequest](#). Hvis du vil være sikker på også at få gamle versioner af IE med, så erstat `var request = new XMLHttpRequest;` med:

```
var request = window.ActiveXObject ? new ActiveXObject('Microsoft.XMLHTTP') : new XMLHttpRequest;
```

Og koden:

```
<script>

    window.onload = kaldWebservice;

    // Kontakt webservice
    function kaldWebservice() {

        // Url'en til webservicekaldet
        var wsurl = "https://swapi.co/api/planets/1/";

        var request = new XMLHttpRequest;

        request.onreadystatechange = function () {

            if (request.readyState == 4 && this.status == 200) {
                udskrivData(JSON.parse(this.responseText));
            }
        };

        request.open("GET", wsurl, true);
        request.send();
    }

    // Udskriv data i html
    function udskrivData(data) {
```

```
        console.log(data);
        document.getElementById("resultat").innerHTML = data.name;
    }

</script>
```

## jQuery Ajax

Og hvis du er glad for jQuery eller alligevel benytter det andre steder i dit projekt, kunne du jo overveje denne version. HUSK at linke til jQuery-lib:

### Koden:

```
<script src="https://code.jquery.com/jquery-3.3.1.js" crossorigin="anonymous"></script>
<script>
    $(document).ready(function () {

        var wsurl = "https://swapi.co/api/planets/1/";

        $.ajax({
            type: "GET",
            dataType: "json",
            url: wsurl,
            success: function (data) {
                udskrivData(data);
            },
            error: function (jqXHR, textStatus, errorThrown) {
                console.log("Noget gik galt: " + jqXHR + "--" + textStatus + "--" + errorThrown);
            }
        });

        function udskrivData(data) {
            console.log(data);
            $("#resultat").html(data.name);
        }
    });
</script>
```

## Fetch

Fetch er noget nyere - og lavet specielt til JavaScript ... ikke alle browsere forstår fetch-måde, især ældre browsere kan have problemer.

We all remember the dreaded XMLHttpRequest we used back in the day to make requests, it involved some really messy code, it didn't give us promises and let's just be honest, it wasn't pretty JavaScript, right? Maybe if you were using jQuery, you used the cleaner syntax with jQuery.ajax(). Well JavaScript has it's own built-in clean way now. Along comes the Fetch API a new standard to make server request jam-packed with promises and all those things we learned to love over the years... [How to Use the JavaScript Fetch API to Get Data](#)

```
<script>

    window.onload = kaldWebservice;

    // Kontakt webservice
    function kaldWebservice() {

        // Url'en til webservicekaldet - request
        var wsurl = "https://swapi.co/api/planets/1/";

        // FETCH-METODEN til at kalde en webservice:
        fetch(wsurl, {
            method: 'get'
        }).then(function (response) {
            return response.json();
        });
    }
</script>
```

```
    }).then(function (json) {
        udskrivData(json);
    }).catch(function (error) {
        console.log(error);
    });

}

// Udskriv data i html
function udskrivData(data) {
    console.log(data);
    document.getElementById("resultat").innerHTML = data.name;
}

</script>
```

[Handling Failed HTTP Responses With fetch\(\)](#)

## All about layers

Forsiden af OL Api ... subclasses til Layers (3 i videoen - men 4 i nyere version?) ...

### Layers

Layers are lightweight containers that get their data from [sources](#).

[ol/layer/Tile](#)

[ol/layer/Image](#)

[ol/layer/Vector](#)

[ol/layer/VectorTile](#)

Tile er det vi lige har testet. Image - som tile men i stedet for tiles er det ét stort image. Vector-layer - se næste modul.

Tiles - pregenerated and cached on the server. Og derfor en limitation på zoomlevels, extends mv. Image-layer created on the fly ... så zoomlevels og extends er meget mere fleksible.

Data for the layer - known as the source. Source-classes ex kan ses her: