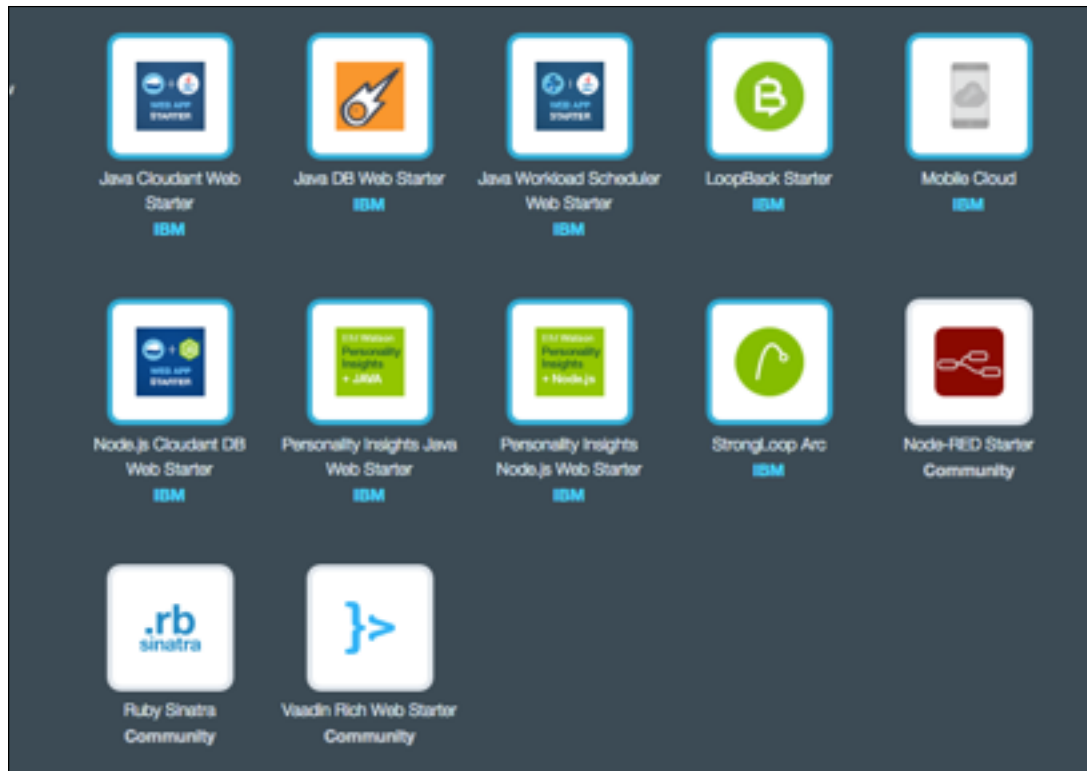


Hvordan bruke telefon som GPS på 10 minutter?

Start med å lage et Node-Red prosjekt i Bluemix

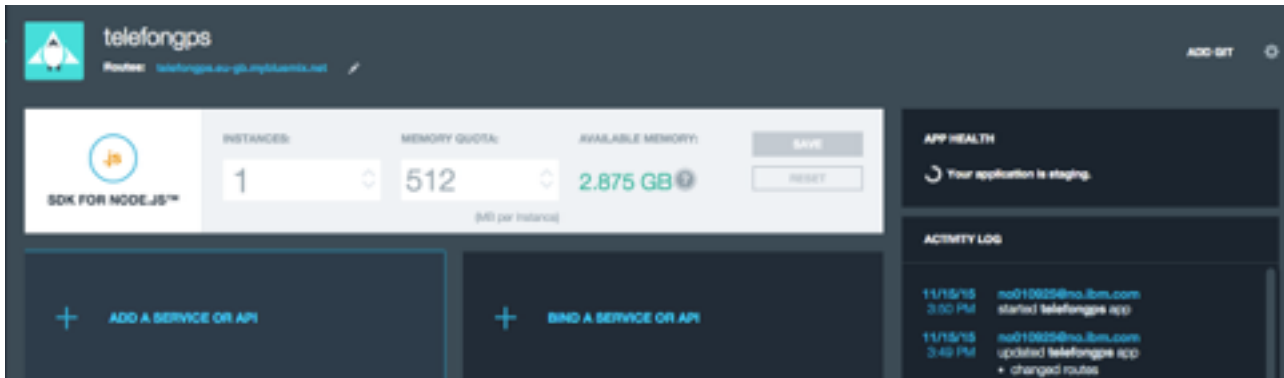
Oslo - 15. november 2015



Velg et navn på din app..

Velg en http input node en template og en http response - koble de sammen slik som vist og gi siden et navn... feks startside...

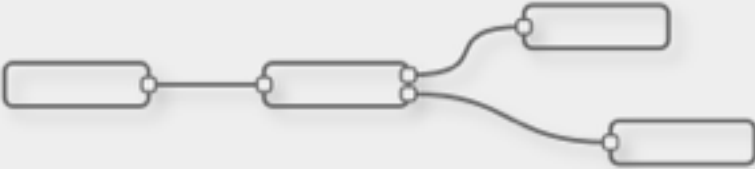
Vent til applikasjoner er ferdig med Staging og klikk på linken øverst på oversikten.



Klikk på Node-red Flow editor

Node-RED in BlueMix

A visual tool for wiring the Internet of Things



Node-RED provides a browser-based editor that makes it easy to wire together flows that can be deployed to the runtime in a single-click.

The version running here has been customised for the BlueMix cloud environment.

More information about Node-RED, including documentation, can be found at nodered.org.

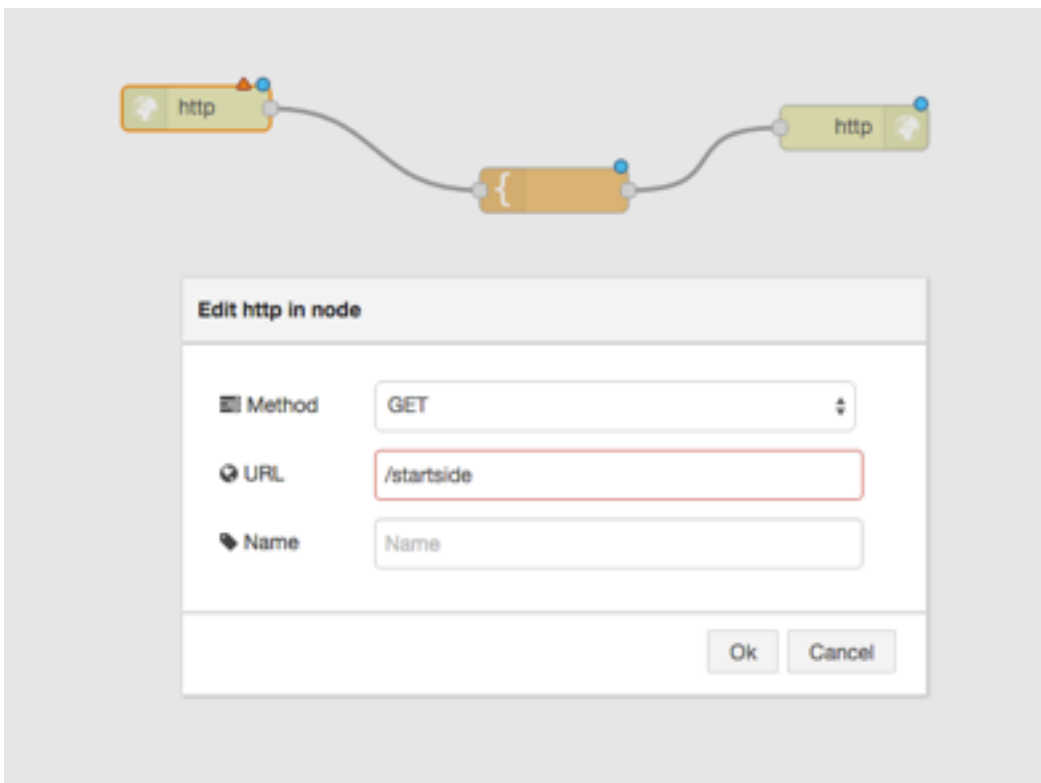
[Go to your Node-RED flow editor](#)

[Learn how to password-protect your instance](#)

[Learn how to customise Node-RED](#)

Du får nå opp arbeidsflaten vi skal jobbe i.

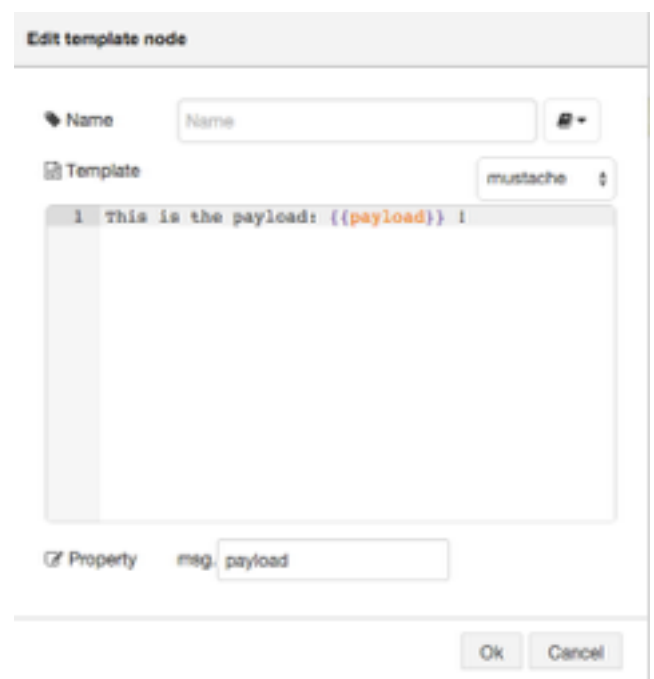
Det første vi skal gjøre er å lage en web side. Velg fra node listen til venstre - en http input - en template og en http response node. Koble de sammen som vist under. Dobbelt-klikk på http-input noden og gi siden et navn f.eks. startside



Trykk OK og trykk Deploy oppe til høyre i arbeidsflaten - test at siden virker - åpne en ny nettleser vindu på adresse <http://<dinappnavn>.mybluemix.net/startside>

Hvis ok - gå tilbake til Node-red og åpne templatene ved et dobbelklikk..

Her skal vi skrive html koden som skal starte GPS'n på telefonen din...



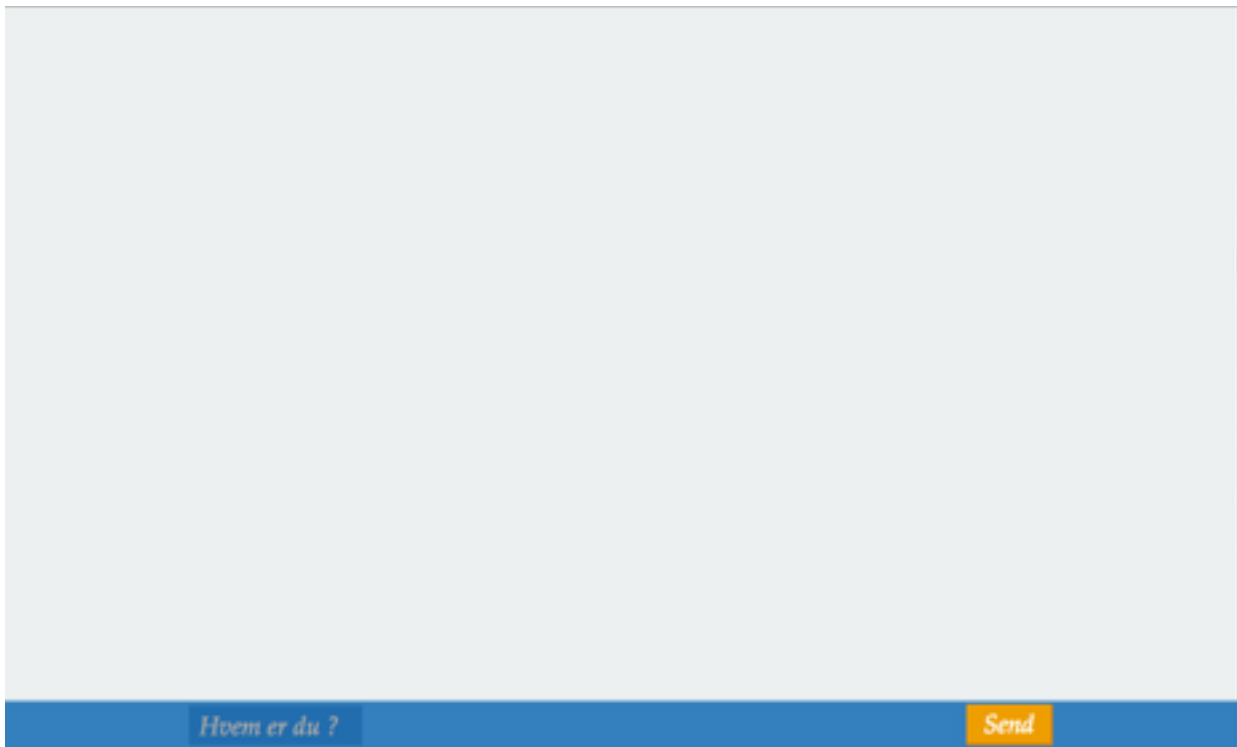
Detter er en start side som gir deg en struktur og et lite eksempel

```
<html>
  <head>
    <meta name="viewport" content="width=320, initial-scale=1">
    <title>GPS Klient</title>
  </head>
  <body>
    <div id="wrapper">
      <div id="userinfo" class="content"></div>
      <div id="footer">
        <div class="content">
          <input type="text" id="user" placeholder="Hvem er du ?" />
          <input type="button" disabled id="send_btn" value="Send" onclick="sendMessage()">
        </div>
      </div>
    </div>
  </body>
</html>
```

For at det skal se litt finere ut kan det legges til litt styling... lim inn koden under head seksjonen...

```
<style type="text/css">
  * {font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;font-style: italic;font-size: 24px;}
  html, body, #wrapper {margin: 0;padding: 0;height: 100%;}
  #wrapper {background-color: #ecf0f1;}
  #footer {box-sizing: border-box;position: fixed;bottom: 0;height: 50px;width: 100%;background-color: #2980b9;}
  #footer .content {padding-top: 4px;position: relative;}
  #user { width: 20%; }
  #message { width: 68%; }
  #send_btn {width: 10%;position: absolute;right: 0;bottom: 0;margin: 0;}
  .content {width: 70%;margin: 0 auto;}
  input[type="text"],
  input[type="button"] {border: 0;color: #fff;}
  input[type="text"] {background-color: #146EA8;padding: 3px 10px;}
  input[type="button"] {background-color: #f39c12;border-right: 2px solid #e67e22;border-bottom: 2px solid #e67e22;min-width:
70px;display: inline-block;}
  input[type="button"]:hover {background-color: #e67e22;border-right: 2px solid #f39c12;border-bottom: 2px solid #f39c12;cursor:
pointer;}
  .system,
  .username {color: #aaa;font-style: italic;font-family: monospace;font-size: 16px;}
  @media(max-width: 1000px) {.content { width: 90%; }}
  @media(max-width: 780px) {#footer { height: 91px; }#chat_box { padding-bottom: 91px; }
  #user { width: 100%; }
  #message { width: 80%; }
  }
  @media(max-width: 400px) {
    #footer { height: 135px; }
    #chat_box { padding-bottom: 135px; }
    #message { width: 100%; }
    #send_btn {position: relative;margin-top: 3px;width: 100%;}
  }
</style>
```

Etter det burde siden ligne på dette:



Det som mangler nå er å legge inn et java script som starter GPS på telefonen.
Java Script kan legges inn på ulike måter in en html side men her kan vi enkelt gjøre det slik:

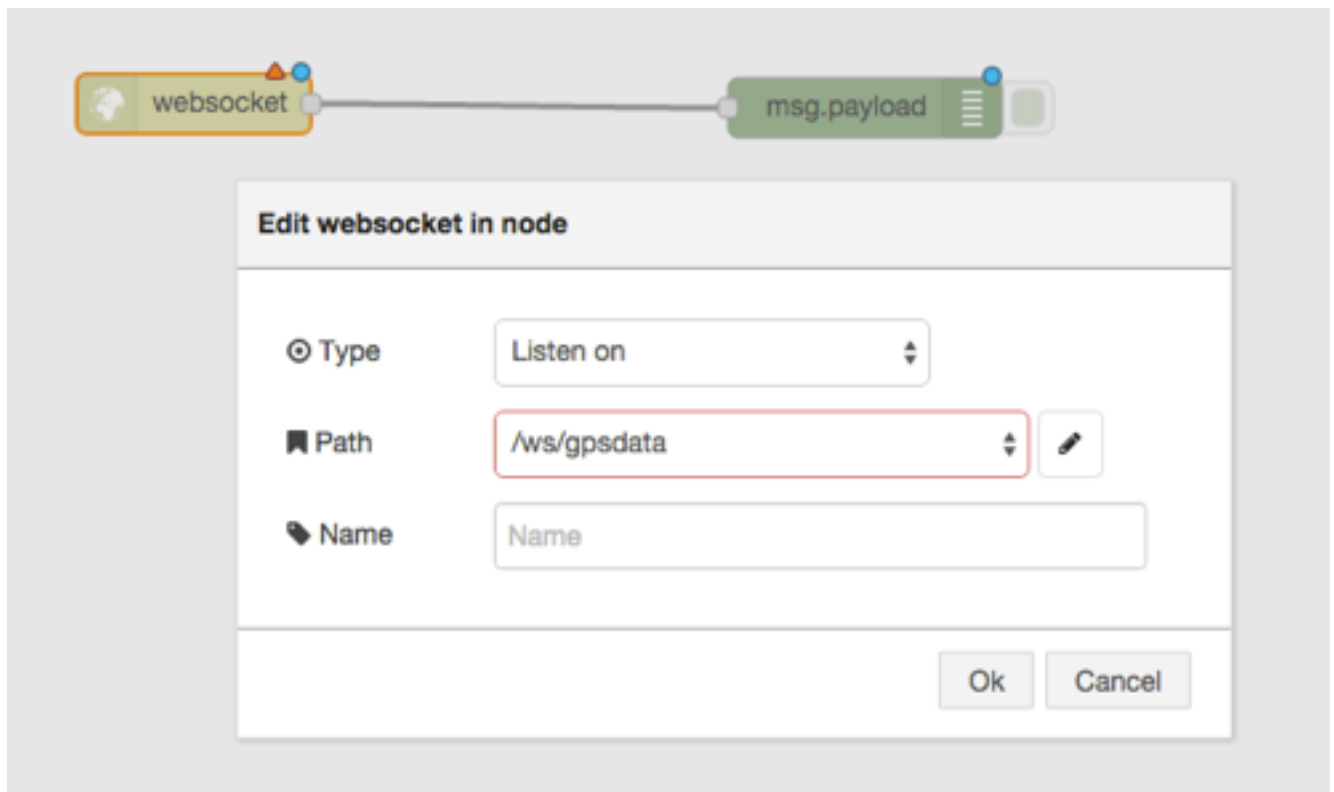
```
<script type="text/javascript">

if (navigator.geolocation){
  console.log("geopos");
  navigator.geolocation.getCurrentPosition(onSucess);
}
else {
  alert("Støtter ikke geoPos");
  exit();
}
function onSucess(position){
  console.log("Sucess");
  lat = position.coords.latitude;
  lon = position.coords.longitude;
  console.log("Dette er din geopossisjon: " +lat+'.'+lon);
  document.getElementById("send_btn").disabled = false;
}
</script>
```

Lim dette inn i bunnen av body delen din...

Test siden og se i loggen i nettleseren din.. du skal se console.log - geopos og success hvis det virker...

Det som nå gjenstår er å sende data fra din telefon tilbake til Node-red serveren. Til dette tenker vi å bruke WebSocket - en lett fin toveis kommunikasjons protokoll. Vi trenger en websocket klient på telefonen og en websocket listner på serveren. I Node-red legger du til en websocket input node og kobler den til en debug node...



Noe slikt som over.. Jeg valgte /ws/gpsdata som sti men dette er valgfritt bare du bruker den samme stien senere...

En enkel websocket klient ser slik ut:

```
websocket = new WebSocket(wsUri);  
websocket.onopen = function(evt) { };  
websocket.onclose = function(evt) { };  
websocket.onmessage = function(evt) { };  
websocket.onerror = function(evt) { };
```

Her kan vi legge opp logikk og feilhåndtering enkelt onopen blir kalt hvis det etableres en webSoket mot wsUri, onclose hvis den dropper og onerror hvis det blir noe feil... Lag scriptet slik at GPS først startes hvis vi klarer å etablere en forbindelse til serveren..

Feks:

```
var wsUri = "ws://{req.headers.host}/ws/gpsdata";
var ws = new WebSocket(wsUri); // setter opp og starter en websocket forbindelse

ws.onopen = function(ev) { connected(); }; // hvis websoket åpnes kalles funksjon connected
ws.onclose = function(ev) { console.log('Disconnected'); } // skriver ut Disconnected hvis sockert feiler
ws.onmessage = function(ev) { }
ws.onerror = function(ev) { console.log("WebSoket Error!"); }

function connected(message) { // Her startes GPS }
```

Videre trenger vi en send funksjon når knappen trykkes:

```
function sendMessage() { // kjøres når du trykker på send knappen
    var payload = { // lager meldingen som skal sendes
        Latitude: lat, Longitude: lon, // legger ved geodata
        User: document.getElementById('user').value, //henter bruker navnet fra websiden
        ts: (new Date()).getTime() // sender med et tids merke..
    };
    ws.send(JSON.stringify(payload)); // selve websocket send funksjonen
    console.log("Medling sendt", JSON.stringify(payload));
    payload = "";
};
```

Eksempel følger på neste side eller her : https://github.com/ethilesen/kodeklubben/blob/master/gps_side.html

```

<html>
<head>
  <meta name="viewport" content="width=320, initial-scale=1">
  <title>GPS Klient</title>
  <style type="text/css">
    * {font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;font-style: italic;font-size: 24px;}
    html, body, #wrapper {margin: 0;padding: 0;height: 100%;}
    #wrapper {background-color: #ecf0f1;}
    #footer {box-sizing: border-box;position: fixed;bottom: 0;height: 50px;width: 100%;background-color: #2980b9;}
    #footer .content {padding-top: 4px;position: relative;}
    #user { width: 20%; }
    #message { width: 68%; }
    #send_btn {width: 10%;position: absolute;right: 0;bottom: 0;margin: 0;}
    .content {width: 70%;margin: 0 auto;}
    input[type="text"],
    input[type="button"] {border: 0;color: #fff;}
    input[type="text"] {background-color: #146EA8;padding: 3px 10px;}
    input[type="button"] {background-color: #f39c12;border-right: 2px solid #e67e22;border-bottom: 2px solid #e67e22;min-width: 70px;display: inline-block;}
    input[type="button"]:hover {background-color: #e67e22;border-right: 2px solid #f39c12;border-bottom: 2px solid #f39c12;cursor: pointer;}
    .system,
    .username {color: #aaa;font-style: italic;font-family: monospace;font-size: 16px;}
    @media(max-width: 1000px) { .content { width: 90%; }}
    @media(max-width: 780px) { #footer { height: 91px; } #chat_box { padding-bottom: 91px; }
      #user { width: 100%; }
      #message { width: 80%; }
    }
    @media(max-width: 400px) {
      #footer { height: 135px; }
      #chat_box { padding-bottom: 135px; }
      #message { width: 100%; }
      #send_btn {position: relative;margin-top: 3px;width: 100%;}
    }
  </style>
</head>
<body>
  <div id="wrapper">
    <div id="userinfo" class="content"></div>
    <div id="footer">
      <div class="content">
        <input type="text" id="user" placeholder="Hvem er du ?" />
        <input type="button" disabled id="send_btn" value="Send" onclick="sendMessage()">
      </div>
    </div>
  </div>
<script>
var wsUri = "ws://{{req.headers.host}}/ws/gpsdata";
var ws = new WebSocket(wsUri); // setter opp og starter en websocket forbindelse
var user = "EspenT";
var lat,lon;
ws.onopen = function(ev) { connected(); }; // hvis websoket åpnes kalles funksjon connected
ws.onclose = function(ev) { console.log('Disconnected'); } // skriver ut Disconnected hvis sockert feiler
ws.onmessage = function(ev) { }
ws.onerror = function(ev) { console.log("WebSoket Error!"); }

function connected(message) { // Her startes GPS
if (navigator.geolocation){
  console.log("geopos");
  navigator.geolocation.getCurrentPosition(onSucess);
}
else {
  alert("Støtter ikke geoPos");
  exit();
}
function onSucess(position){
  console.log("Sucess");
  lat = position.coords.latitude;
  lon = position.coords.longitude;
  console.log("Dette er din geoposisjon: " +lat+'.'+lon);
  document.getElementById("send_btn").disabled = false;
}
}

function sendMessage() { // kjøres når du trykker på send knappen (<input type="button" disabled id="send_btn" value="Send" onclick="sendMessage()">)
  var payload = {
    Latitude: lat,Longitude: lon,
    User: document.getElementById('user').value,
    ts: (new Date()).getTime()
  };

  ws.send(JSON.stringify(payload));
  console.log("Medling sendt",JSON.stringify(payload));
  payload = "";
};
</script>
</body>
</html>

```


Komplett kode eksempel finner du på neste side....

Nå har du sende gps data fra telefonen din tilbake til Node-Red servern. Disse dataene kan brukes til så mangt - men en liten test kan jo være å plote de på et kart for å se hvor du er...

For å gjøre dette lager du en ny webside..

Velg en http Input node - en template node og en http response node. Koble de sammen som sist og gi siden et navn feks kartside.

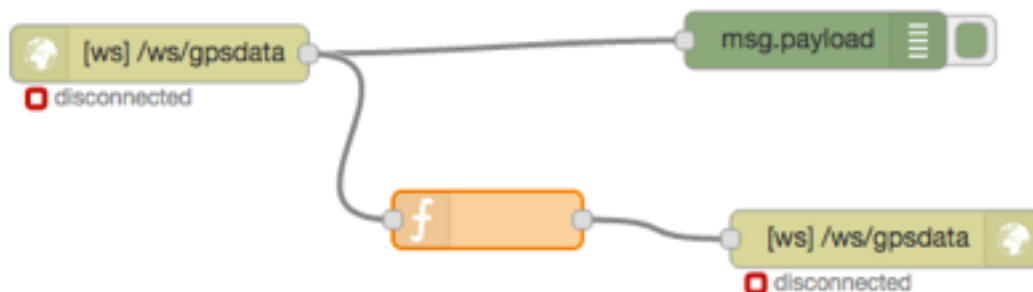


Lag en WebSocket klient som kobler til den websocket vi har brukt tidligere /ws/gpsdata

I onmessage må du lage noe kart plotting. Her finnes det flere løsninger - google maps, openStreet map ol... Du finner fine eksempler på nettet.

En ting til du må huske på - for å kunne sende websocket meldinger til mer enn en mottaker så må vi fjerne den spesifikke sessionsid som opprettes mellom sender og mottaker.

Velg en funksjons node og koble den til websoket input noden din. Velg en websoket out node - spesifiser den til å sende ut på /ws/gpsdata.



Her finner du en komplett kartside - <https://github.com/ethilesen/kodeklubben/blob/master/kartside.html>

```
<html>
<head>
  <title>Node-Red GeoSporing Kart</title>
  <script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=true"></script>
  <script type="text/javascript" src="http://yourjavascript.com/4594301102/gmaps.js"></script>
  <style type="text/css" media="screen">
    #map {position:absolute;top: 0; bottom: 0; left: 0; right: 0;}
  </style>
</head>
<body>
  <div id="map"></div>
  <script type="text/javascript">
    var wsUri = "ws://" + {{req.headers.host}} + "/ws/gpsdata";
    var map;
    var ws;
    map = new GMaps({
      div: '#map',
      lat: 58.699776,
      lng: 10.546875,
      zoom: 6
    });
    ws = new WebSocket(wsUri);
    ws.onopen = function(){ console.log("Connected websocket");};
    ws.onerror = function(){ console.log("Websocket error"); };
    ws.onmessage = function(evt){
      console.log("Message recieved!", evt.data);
      var latlng = JSON.parse(evt.data);
      map.removeMarkers();
      console.log("Got marker at " + latlng.Latitude + ", " + latlng.Longitude + " from : " + latlng.User);
      map.setMapTypeId(google.maps.MapTypeId.HYBRID);
      map.setZoom(80);
      map.setCenter(latlng.Latitude, latlng.Longitude);
      map.addMarker({
        lat: latlng.Latitude,
        lng: latlng.Longitude,
        title: 'Navn: ' + latlng.User + '\n' + 'Latitude: ' + latlng.Latitude + '\n' + 'Longitude: ' +
latlng.Longitude + '\n' + 'Time: ' + latlng.ts,
        click: function(e) {
latlng.Latitude + '\n'
latlng.Longitude + '\n'
          alert('Denne device spores....' + '\n' + 'Navn: ' + latlng.User + '\n' + 'Latitude: ' +
latlng.Latitude + '\n' + 'Longitude: ' + latlng.Longitude + '\n' + 'Time: ' + latlng.ts );
        }
      });
    };
  </script>
</body>
</html>
```

Det er kanskje dumt å ha to sider ? - prøv å utvide gps klienten til også å vise kartet...
De neste tre sidene er et eksempel...
https://github.com/ethilesen/kodeklubben/blob/master/gps_kart_side.html

```
<html>
<head>
  <meta name="viewport" content="width=320, initial-scale=1">
  <title>GPS Kart Klient</title>
  <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.4/jquery.min.js"></script>
  <script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=true"></script>
  <script type="text/javascript" src="http://yourjavascript.com/4594301102/gmaps.js"></script>
  <style type="text/css">
    * {font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;font-style: italic;font-size: 24px;}
    html, body, #wrapper {margin: 0;padding: 0;height: 100%;}
    #wrapper {background-color: #ecf0f1;}
    #footer {box-sizing: border-box;position: fixed;bottom: 0;height: 50px;width: 100%;background-color: #2980b9;}
    #footer .content {padding-top: 4px;position: relative;}
    #user { width: 20%; }
    #message { width: 68%; }
    #send_btn {width: 10%;position: absolute;right: 0;bottom: 0;margin: 0;}
    .content {width: 70%;margin: 0 auto;}
    input[type="text"],
    input[type="button"] {border: 0;color: #fff;}
    input[type="text"] {background-color: #146EA8;padding: 3px 10px;}
    input[type="button"] {background-color: #f39c12;border-right: 2px solid #e67e22;border-bottom: 2px solid
#e67e22;min-width: 70px;display: inline-block;}
    input[type="button"]:hover {background-color: #e67e22;border-right: 2px solid #f39c12;border-bottom: 2px
solid #f39c12;cursor: pointer;}
    .system,
    .username {color: #aaa;font-style: italic;font-family: monospace;font-size: 16px;}
    @media(max-width: 1000px) {.content { width: 90%; }}
    @media(max-width: 780px) {#footer { height: 91px; }#chat_box { padding-bottom: 91px; }
      #user { width: 100%; }
      #message { width: 80%; }
    }
    @media(max-width: 400px) {
      #footer { height: 135px; }
      #chat_box { padding-bottom: 135px; }
      #message { width: 100%; }
      #send_btn {position: relative;margin-top: 3px;width: 100%;}
    }
    #map {height: 100%}
  </style>
</head>
```

```

<body>
  <div id="wrapper">
    <div id="map"></div>
    <div id="footer">
      <div class="content">
        <input type="text" id="user" placeholder="Hvem er du ?" />
        <input type="button" disabled id="send_btn" value="Send" onclick="sendMessage()">
      </div>
    </div>
  </div>
</div>
<script>
  var wsUri = "ws://{req.headers.host}/ws/gpsdata";
  var ws = new WebSocket(wsUri); // setter opp og starter en websocket forbindelse
  var lat,lon;
  var map;
$(document).ready(function(){

  startws();
});

function initmap(){ // init map
  map = new GMaps({
    div: '#map',
    lat: 58.699776,
    lng: 10.546875,
    zoom: 4
  });
};

function startws(){ // start websocket
ws.onopen = function(ev) { connected(); }; // hvis websoket åpnes kalles funksjon connected
ws.onclose = function(ev) { console.log("Disconnected");} // skriver ut Disconnected hvis sockert feiler
ws.onmessage = function(ev) {

  console.log("message arrived!")
  var data = JSON.parse(ev.data);
  zoom = 20;
  map.setMapTypeId(google.maps.MapTypeId.HYBRID);
  map.setZoom(zoom);
  map.setCenter(data.Latitude, data.Longitude);

  map.addMarker({ // setter en makør, men informasjon i mouseover/klikk
    lat: data.Latitude,
    lng: data.Longitude,
    title: 'Navn: ' + data.User + '\n' + 'Latitude: ' + data.Latitude + '\n' + 'Longitude: ' + data.Longitude + '\n',
    click: function(e) {
      alert('Denne device spores....' + '\n' + 'Navn: ' + data.User + '\n' + 'Latitude: ' + data.Latitude + '\n' +
'Longitude: ' + data.Longitude + '\n' );
    }
  });
}

}

```

```

ws.onerror = function(ev) {console.log("WebSoket Error!"); }

function connected(message) { // Her startes GPS

if (navigator.geolocation){
  console.log("geopos");
  navigator.geolocation.getCurrentPosition(onSucess);
}
else {
  alert("Støtter ikke geoPos");
  exit();
}
function onSucess(position){
  console.log("Sucess");
  lat = position.coords.latitude;
  lon = position.coords.longitude;
  console.log("Dette er din geoposisjon: " +lat+":"+lon);
  document.getElementById("send_btn").disabled = false; // enabler send knappen i ux...
  initmap(); // laster kart når client er connected og vi har en gps possisjon
}
}
}

function sendMessage() { // kjøres når du trykker på send knappen (<input type="button" disabled id="send_btn"
value="Send" onclick="sendMessage()">)
  var payload = {
    Latitude: lat,Longitude: lon,
    User: document.getElementById('user').value,
    ts: (new Date()).getTime()
  };

  ws.send(JSON.stringify(payload));
  console.log("Medling sendt",JSON.stringify(payload));
  payload = "";

};
</script>
</body>
</html>

```

