Oppgaver Fagkveld Cross Plattform utvikling

# Oppsett

Se vedlagte filer eller se online manualer:

[Android](http://docs.phonegap.com/en/2.7.0/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android)  
De som vil prøve seg med iPhone finner en guide [her](http://docs.phonegap.com/en/2.7.0/guide_getting-started_ios_index.md.html#Getting%20Started%20with%20iOS)

# Hello World

Skriv et Hello World-program med PhoneGap.   
  
Erstatt teksten “Device is Ready” med “Hello, World!” i p-taggen på linje 33 i index.html .

# 2 Integrasjon mot interne komponenter

## 2a) Geolocation

Når PhoneGap er klar til å brukes fyres eventet *deviceready* av.   
Ved å lytte på dette eventet kan man forsikre seg om at koden man vil kjøre ikke blir kjørt før PhoneGap er klar.   
I *js/index.js* blir eventet «bindet» til funksjonen *onDeviceReady* som igjen kaller på funksjonen *receivedEvent*. Her kan vi legge den koden vi vil skal kjøre når PhoneGap er klar.

*main.js* inkluderer en del hjelpemetoder.  
**Inkluder** *main.js* **i** *index.html* **over der** *index.js* **blir inkludert slik at metodene i** *main.js* **kan kalles fra** *index.js*

En av hjelpemetodene i *main.js* gir deg mobilens lokasjon.  
**Kall denne metoden fra** *index.js* **når PhoneGap er klar.**

**Deploy applikasjonen til mobilen din eller en emulator.**   
Du bør få opp en alert som forteller om mobilens posisjon.

Alerten rapporterer bare breddegrad og lengdegrad.  
**Endre koden i hjelpemetoden i main.js slik at den også rapporterer høyden over havet.   
Deploy og sjekk at du får opp både breddegrad, lengdegrad og høyde over havet.**

NB: Ikke alle mobiler er like gode på høyde over havet, så det kan være det kommer ut som *null*

## 2b) Vis lokasjon i kart

I denne oppgaven skal vi vise et kart og markere vår posisjon i kartet, basert på posisjonen vi fikk fra PhoneGap i forrige oppgave.

**Last ned** *map.html* **fra wiki-siden/minnepinne og legg den i samme mappe som** *index.html* **Legg** *map.js* **i** *js***-mappen.Lag en link i diven med klasse** *app* **i** *index.html* **som tar deg til** *map.html***I** *map.html* **er det en div med id** *map\_canvas.* Vi skal bruke denne til å vise kartet i.  
**Gi diven en høyde og bredde på 100 %**

For å lage et kart trenger vi en lokasjon representert som et *google.maps.LatLng*-objekt.  
Du lager et slikt objekt med din breddegrad og lengdegrad på denne måten:

**var** myLocation = new google.maps.LatLng(<din\_breddegrad>, <din\_lengdegrad>);

**I** *map.js* **ligger det en funksjon for å vise kartet og din posisjon i kartet:**

**var** showMyPositionInMap = **function(**position**)** {

**var** myLocation = new google.maps.LatLng(<din\_breddegrad>, <din\_lengdegrad>);

**var** map = new google.maps.Map(document.getElementById('map\_canvas'), {

mapTypeId: google.maps.MapTypeId.ROADMAP,

center: myLocation,

zoom: 17

});

**var** marker = new google.maps.Marker({

position: myLocation,

map: map,

title: "You are here"

});

marker.setMap(map);

};

**Kall på funksjonen din fra** *suc***-funksjonen****i map.js**

# 2c ) notfication.alert

PhoneGap har sin egen alert i [Notification-APIet](http://docs.phonegap.com/en/2.7.0/cordova_notification_notification.md.html#Notification).   
Her kan du selv bestemme overskriften på alerten hvis du vil ha noe annet enn «Alert».  
Du kan også definere en egen tekst for knappen man bruker til å dismisse alerten og du kan angi en callback-funksjon som blir kalt når alerten dismisses.

Notification-APIet har også annen kul funksjonalitet som blant annet gjør det mulig å få mobilen til å pipe eller vibrere.

**Slutt å kalle** *main.js* **sin** *getLocation***. Lag i steden en** *notification.alert* **i** *map.js***.   
Gi alertboksen en annen overskrift og knappen en annen tekst.   
Gjør om på koden slik at kartet ikke vises før alertboksen er dismisset.**

# 3 Bygg applikasjonen online med PhoneGap Build

En av de kulere tingene med PhoneGap er at man kan bygge applikasjonen online, scanne en QR-kode og få den inn på sin egen mobil.

PhoneGap Builder kobler seg mot Github for å hente koden din.

## Github

**Gå til** [**Github**](https://github.com/) **og lag et nytt public repository på din egen bruker.**

**Åpne en terminal med tilgang til git-funksjonalitet (f.eks. Git Bash)  
Gå inn i prosjektmappen din.  
Skriv følgende:**

git init

git add assets/www

git rm –cached assets/www/cordova-2.7.0.js

git commit -m "first commit"

git remote add origin https://github.com/<ditt\_brukernavn>/<navnet\_på\_ditt\_repo>.git

git push -u origin master

PhoneGap Build trenger bare webdelen av koden. Derfor pusher vi bare *assets*-mappen.  
Alle andre filer kan legges til i *.gitignore* hvis man ønsker, slik at det blir lettere å adde og commite filer.

*cordova-2.7.0.js* er plattformspesifikk så vår versjon vil bare fungere for den plattformen vi valgte da vi lagde prosjektet. Det er derfor vi må fjerne den fra mappen.  
PhoneGap Build legger ved riktig *cordova.js*-fil for deg når den bygger, basert på hvilken plattform den bygger for.  
  
**Legg til** <script type="text/javascript" src="cordova.js"></script> **i** *index.html***for å referere til filen som blir lagt til av PhoneGap Build.**Legg det i linjen over <script type="text/javascript" src="cordova-2.7.0.js"></script>

Vi beholder den gamle linjen allikevel, slik at lokale bygg fortsatt vil fungere.

## PhoneGap Build

**Gå til** [**PhoneGap Build**](https://build.phonegap.com/)

**Knytt din github-konto til PhoneGap Build.**

**Legg til en ny app.**

**Velg githubrepositoryet du laget i stad.**

**Trykk Ready to Build (må kanskje refreshe siden for å få opp knappen)**

**Hent koden fra github.**

**Bygg koden.**

**Scan QR-koden og installer appen. (Prøv gjerne på en annen plattform enn den du begynte med også).**

**Kjør appen. Se at den kjører som før.**

It works! Magi!

# 4 Integrer mot FourSquare

I denne oppgaven skal vi benytte oss av Foursquare sitt API for å hente ut Venues (steder). Vi kommer til å bruke jQuery for å kalle APIet. Last ned jQuery fra <http://jquery.com/download/> og referer .js filen i map.html.

## 4a) Hente og vise venues

### I) Hent ut venues fra Foursquare

Hent ut venues fra Foursquare innenfor en radius på 1 km fra din posisjon ved å kalle på:

[https://api.foursquare.com/v2/venues/search?client\_id=RY0OBEAXNHYC53X210PSEDDWW4FM5MR3U31XW2WHSKZTNACU&client\_secret=JK2TFSOXPBEAJCLJ0S4KLY3UTN1U4CYXEG4RV3WL41SKGJWC&v=20130513&ll=<latitude,longitude>&radius=1000](https://api.foursquare.com/v2/venues/explore?client_id=RY0OBEAXNHYC53X210PSEDDWW4FM5MR3U31XW2WHSKZTNACU&client_secret=JK2TFSOXPBEAJCLJ0S4KLY3UTN1U4CYXEG4RV3WL41SKGJWC&v=20130513&ll=%3clatitude,longitude%3e&radius=1000)

Bytt ut <latitude,longitude> med din posisjon. Denne URLen returnerer en json som inneholder attributtet ”response”, som inneholder attributtet ”venues”, som er en array av venues.

Hvert venue inneholder bl.a. attributtene ”name” og ”location”. Location inneholder bl.a. attributtene ”lat” og ”lng”. Dette er de attributtene vi kommer til å forholde oss til i denne oppgaven.

Bruk $.getJSON(url, function(data) {…}) for å kalle på APIet. Data er JSON-objektet som returneres. For å loope gjennom alle venues i responsen kan man bruke $.each(data.response.venues, function(index, venue) {…}).

For en komplett oversikt over venue, se: <https://developer.foursquare.com/docs/responses/venue>

### II) Lag markers

Lag markers for hvert venue og vis dem på kartet.

### III) Infovindu

Når man klikker på en venue på kartet så vil man at det skal dukke opp et info vindu hvor navnet på venue vises. Man må registrere et click event på hver marker, som kartet lytter på. I callback-funksjonen bruker man et InfoWindow-objekt ved å sette innnhold i dette og deretter åpne det i kartet.

var infowindow = new google.maps.InfoWindow();

google.maps.event.addListener(marker, 'click', function () {

infowindow.setContent(<innhold>);

infowindow.open(kart, marker);

});

4b) Lagring av venues

I) Favorittvenues  
Når man har vist venues på kartet, så er det ønskelig å kunne velge noen venues man vil markere som favoritter. Skriv funksjonalitet for å markere et venue som favoritt. Dette kan gjøres ved å velge fra en liste eller lignende.

II) Lagring av favorittvenues

Når man har markert et venue som favoritt så skal den lagres lokalt på telefonen ved å bruke HTML 5 Local storage, som er en enkel key-value lagring. Eksempler på bruk:

window.localStorage.setItem("key", "value")

var value = window.localStorage.getItem("key");

Et venue som er markert som favoritt skal indikere dette på en eller annen måte (f. eks. favorittstjerne) i info vinduet når man klikker på det i kartet.

4c) Legg til bilder på venues

På venues man har markert som favoritter skal det være mulig å legge til bilder. For å ta bildet bruker kaller man på navigator.camera.getPicture( cameraSuccess, cameraError, [ cameraOptions ] );

CameraOptions er valgfri, returnerer URI til bilde som default. Bruk   
{ destinationType : Camera.DestinationType.DATA\_URL } hvis man ønsker bildet som base64 encodet string.

CameraSuccess tar imot bildedata (base64 encoded string eller URI til bildet) som parameter.  
cameraError tar imot en filmelding som parameter.

Bildet skal også lagres lokalt på telefonen. Når det er lagt til et bilde på et venue så skal dette vises når i info vinduet som kommer opp når man trykker på et venue på kartet.