Intel® XDK Lab Script

12.10.2014

Michael Roevie Victoria

1.0 Purpose

The purpose of this document is to describe the tutorial on “Developing a simple mobile app using Intel® XDK” for the lab session.

2.0 An Introduction to the Intel® XDK

Please follow the presentation for an introduction to the Intel XDK

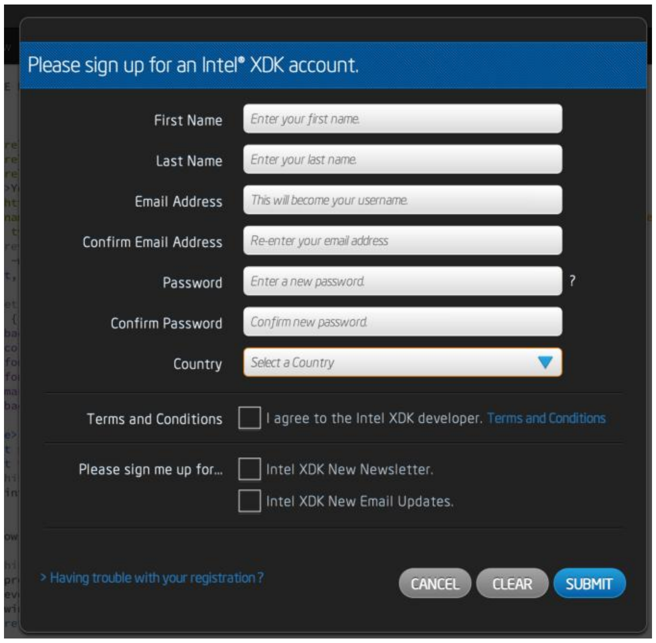
3.0 Hands on lab activities

In this lab we will learn end to end development of a simple mobile app using Intel XDK. The goal of this lab is to explore some device features and build a hybrid app that utilizes simple features like beep, vibrate and playing sound. This activity will provide you hands-on experience with designing the UI using App Designer, programming using Intel XDK APIs to access native features of the device and using the emulator to emulate different devices. You will also learn how to preview your app on the device and build an app package that runs on device.

3.1 Pre-lab activities

1.1. Sign up for an account

- Open XDK New by double clicking on the icon
- Sign up for an account, filling in the necessary details



Please sign up for an Intel® XDK account.

First Name	Enter your first name.
Last Name	Enter your last name.
Email Address	This will become your username.
Confirm Email Address	Re-enter your email address
Password	Enter a new password.
Confirm Password	Confirm new password.
Country	Select a Country
Terms and Conditions	☐ I agree to the Intel XDK developer. Terms and Conditions
Please sign me up for...	☐ Intel XDK New Newsletter.
	☐ Intel XDK New Email Updates.

> Having trouble with your registration?

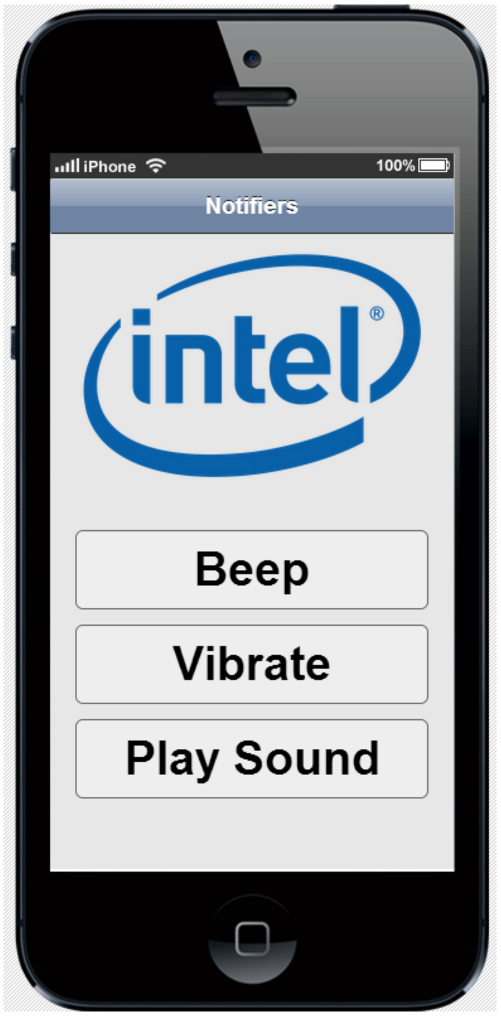
- Then log in with the credentials you created.

1.2. Download App lab on your phone

- If you have either an iPhone or an Android device, search for the free “App Lab” app in the store to download.
- This app will be used later in the lab to preview your app on the device.

4.0 Develop a simple hybrid app “Notifiers”

Our goal is to develop an app that looks like this:



For simplicity we will divide the app development into 5 parts.

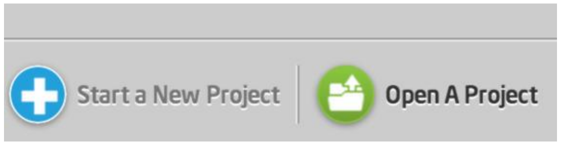
1. Create the UI
2. Add functionality
3. Preview the app in Emulator
4. Preview the app on device
5. Build a package for app store

4.1 Create the UI

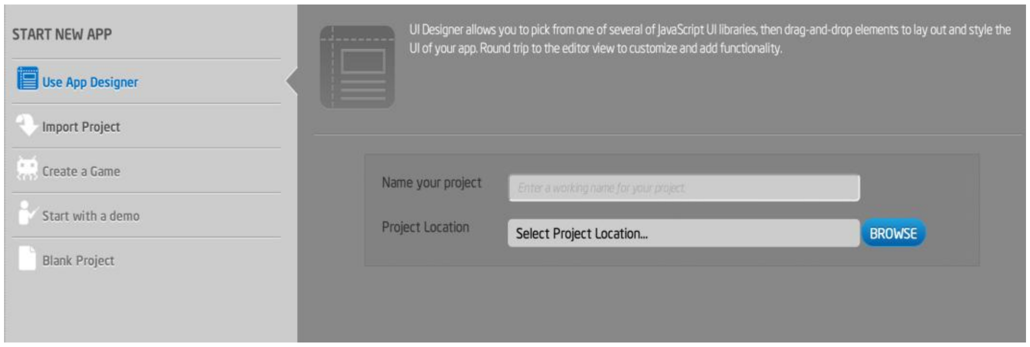
To build UI that is responsive across multiple devices and form factors, we use libraries or frameworks like jQuery Mobile, Twitter Bootstrap or App Framework. To make design phase easier Intel XDK has a built-in tool called App Designer, which lets you drag and drop components for your UI. You can choose one of the three libraries mentioned above and App Designer will generate code using that library for your UI. For this lab we will build the UI with App Designer using App Framework. App Framework is Intel's JavaScript library that is specifically designed for mobile UIs.

4.1.1 Start a new Project

- Open XDK New by clicking on the icon.
- Under the Projects tab click on the Start a New Project to create a new project



- Select Use App Designer for this project.
- Fill in the name of your project and select the folder where you want this project to be saved.

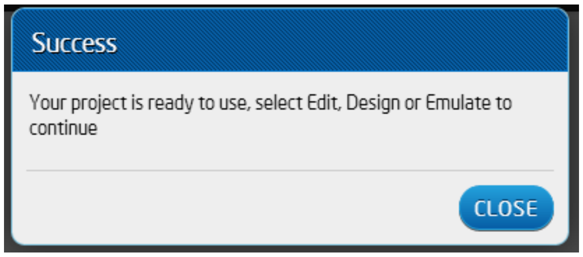


UI Designer allows you to pick from one of several of JavaScript UI libraries, then drag-and-drop elements to lay out and style the UI of your app. Round trip to the editor view to customize and add functionality.

Name your project: Enter a working name for your project

Project Location: Select Project Location...

- Create the project. When you see the following message box, you will have a project created in your folder with default files Index.html and project.xdk

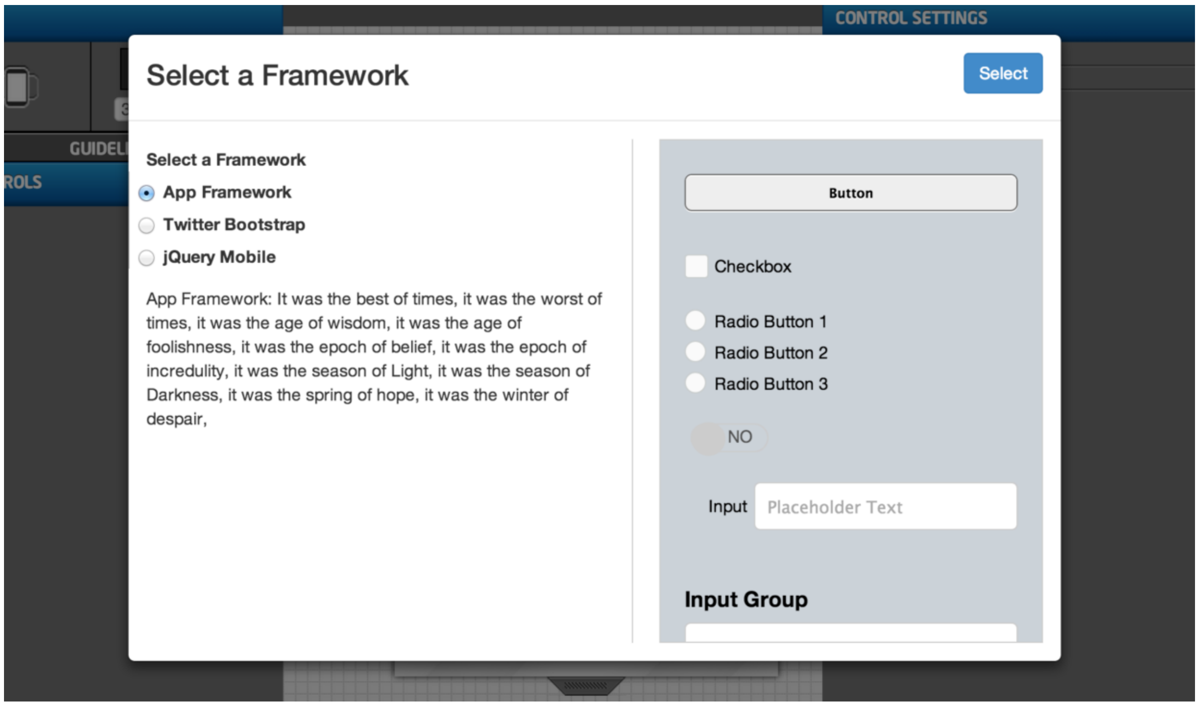


Success

Your project is ready to use, select Edit, Design or Emulate to continue

4.1.2 Using App Designer

- Click on index.html in the file manager and click on App Designer tab, this will open up App Designer for you to design your UI.
- Select App Framework to start with.

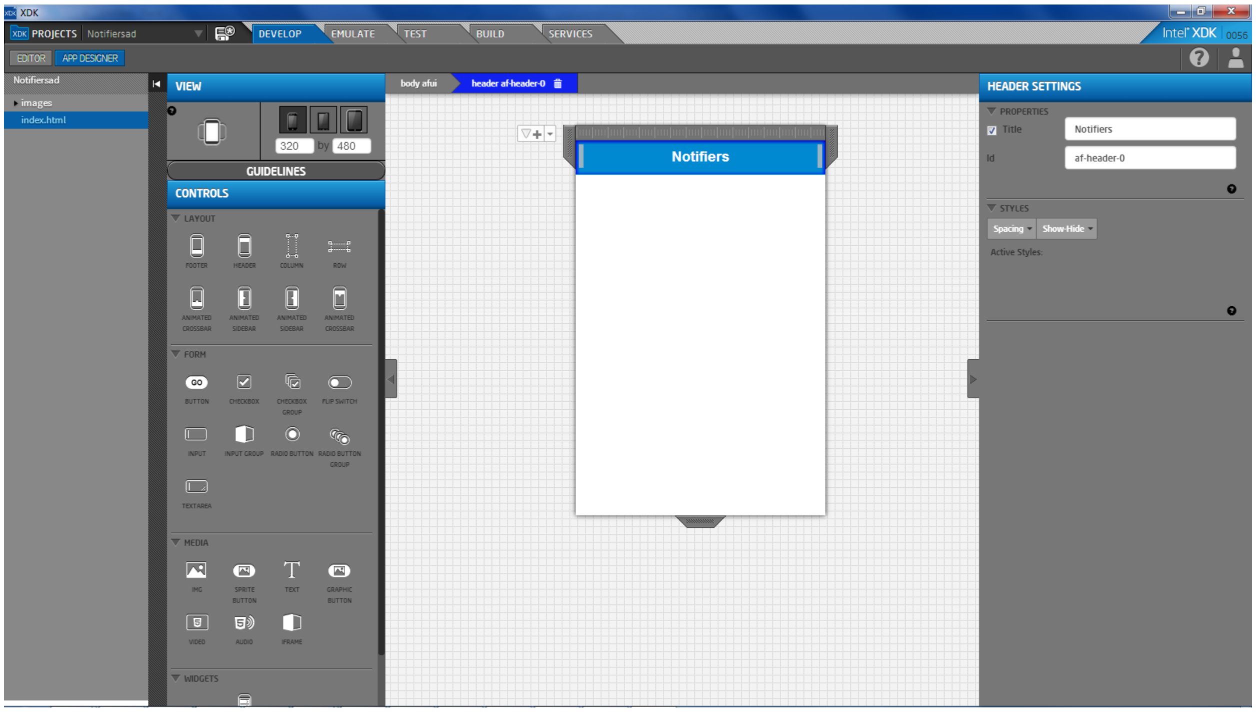


Select a Framework

Select a Framework
- [✓ App Framework]
- Twitter Bootstrap
- jQuery Mobile

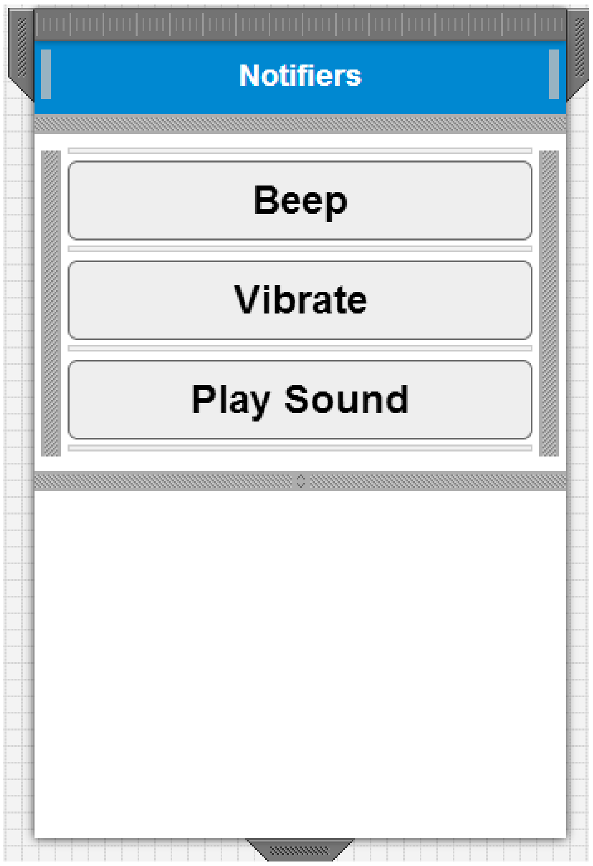
App Framework: It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair,

- From the controls panel drag and drop a header component into the design panel.
- Add a title by selecting the Title on the Properties menu to the right.

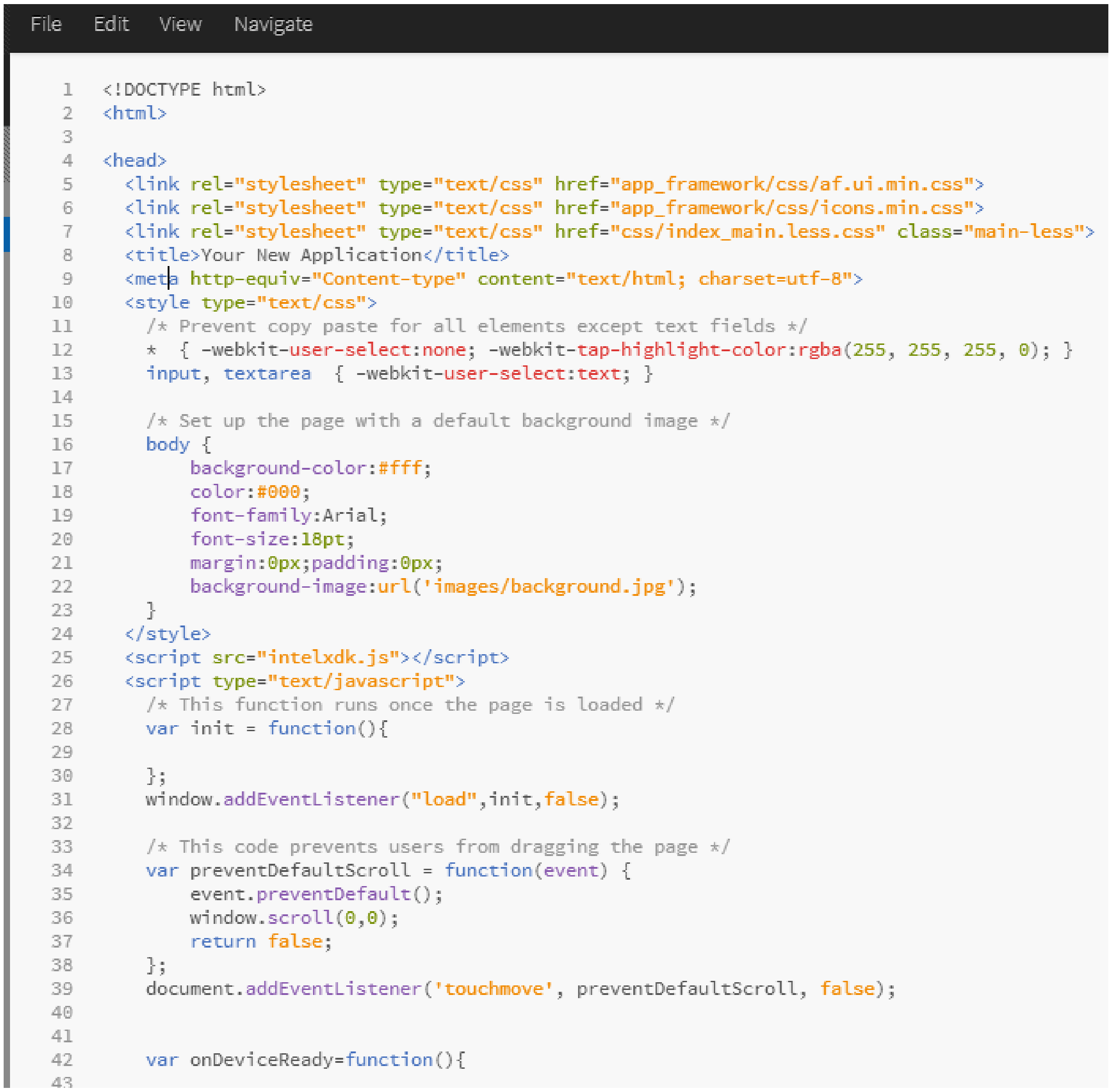


Title: Notifiers
Id: af-header-0

- Drag and drop a button. Change the text of label under the properties to “Beep”
- Similarly add two more buttons for “Vibrate” and “Play Sound”



- Press Ctrl S to save any UI changes
- View the source using Brackets Editor



```
<!DOCTYPE html>
<html>

<head>
  <link rel="stylesheet" type="text/css" href="app_framework/css/af.ui.min.css">
  <link rel="stylesheet" type="text/css" href="app_framework/css/icons.min.css">
  <link rel="stylesheet" type="text/css" href="css/index_main.less.css" class="main-less">
  <title>Your New Application</title>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <style type="text/css">
    /* Prevent copy paste for all elements except text fields */
    *  { -webkit-user-select:none; -webkit-tap-highlight-color:rgba(255, 255, 255, 0); }
    input, textarea  { -webkit-user-select:text; }

    /* Set up the page with a default background image */
    body {
        background-color:#fff;
        color:#000;
        font-family:Arial;
        font-size:18pt;
        margin:0px;padding:0px;
        background-image:url('images/background.jpg');
    }
  </style>
  <script src="intelxdk.js"></script>
  <script type="text/javascript">
    /* This function runs once the page is loaded */
    var init = function(){

    };
    window.addEventListener("load",init,false);

    /* This code prevents users from dragging the page */
    var preventDefaultScroll = function(event) {
        event.preventDefault();
        window.scroll(0,0);
        return false;
    };
    document.addEventListener('touchmove', preventDefaultScroll, false);


    var onDeviceReady=function(){
```

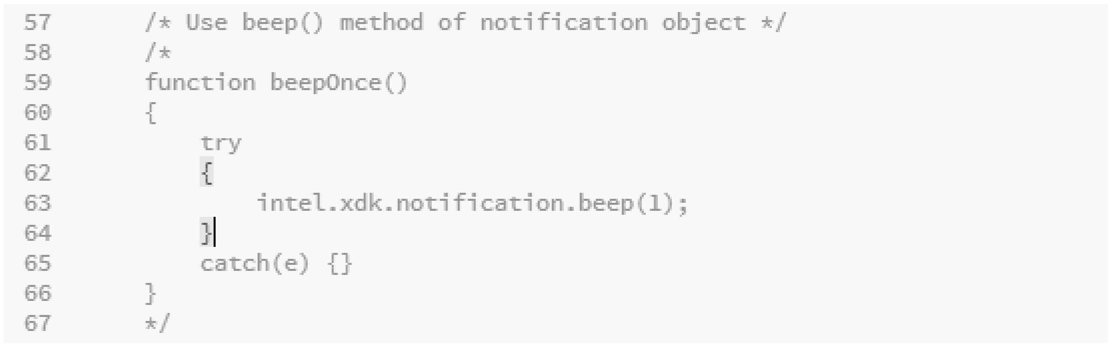
4.2 Add functionality

For the sake of consistency and completeness, please download the project from github that has been created beforehand. It will provide a skeleton for you to insert the functionality at proper places and it also has all the assets necessary for this project.

Unzip the project and open the project using “Open A Project” icon under Projects tab .

1.3. Open and Edit project

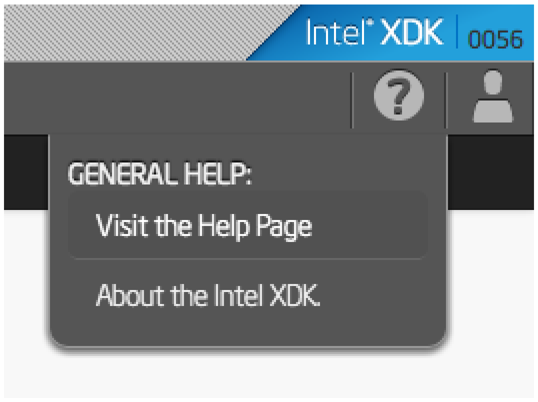
- Select the “Notifiers” project in the Projects tab
- This is the same UI as we created earlier, but with some extra skeleton for adding the functionality.
- Uncomment code for beep.



```
/* Use beep() method of notification object */
/*
function beepOnce()
{
    try
    {
        intel.xdk.notification.beep(1);
    }
    catch(e) {}
}
*/
```

- Use emulator to emulate the beep functionality.

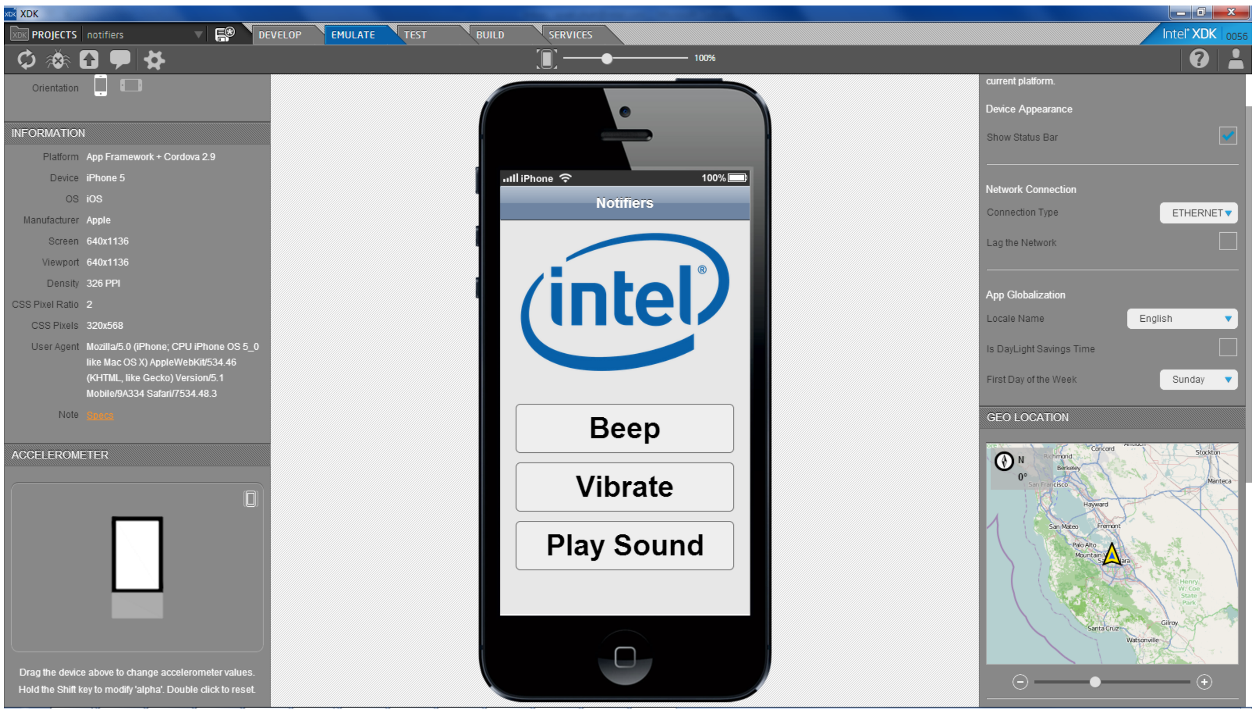
- Lookup API for vibrate in the documentation by clicking “?” at the top right of XDK.



- Add code for vibrate on your own
- Add code for playing sound on your own. APIs will be explained.
- Solution files on github (link and QR code will be given)

4.3 Preview in Emulator

- View the project in the emulator.
- Change devices to see how the app looks on different devices



INFORMATION

Platform	App Framework + Cordova 2.9
Device	iPhone 5
OS	iOS
Manufacturer	Apple
Screen	640x1136
Viewport	640x1136
Density	326 PPI
CSS Pixel Ratio	2
CSS Pixels	320x568
User Agent	Mozilla/5.0 (iPhone; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
Note	Specs

ACCELEROMETER

Drag the device above to change accelerometer values. Hold the Shift key to modify 'alpha'. Double click to reset.

current platform.

Device Appearance

Show Status Bar	✓

Network Connection

Connection Type	ETHERNET
Lag the Network	

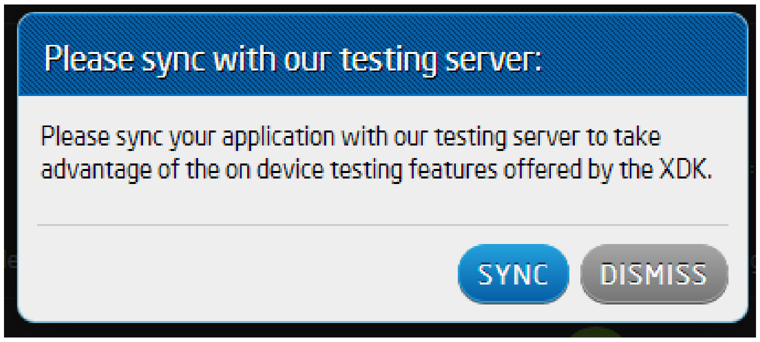
App Globalization

Locale Name	English
Is DayLight Savings Time	
First Day of the Week	Sunday

4.4 Preview on Device

To preview your app on your device:

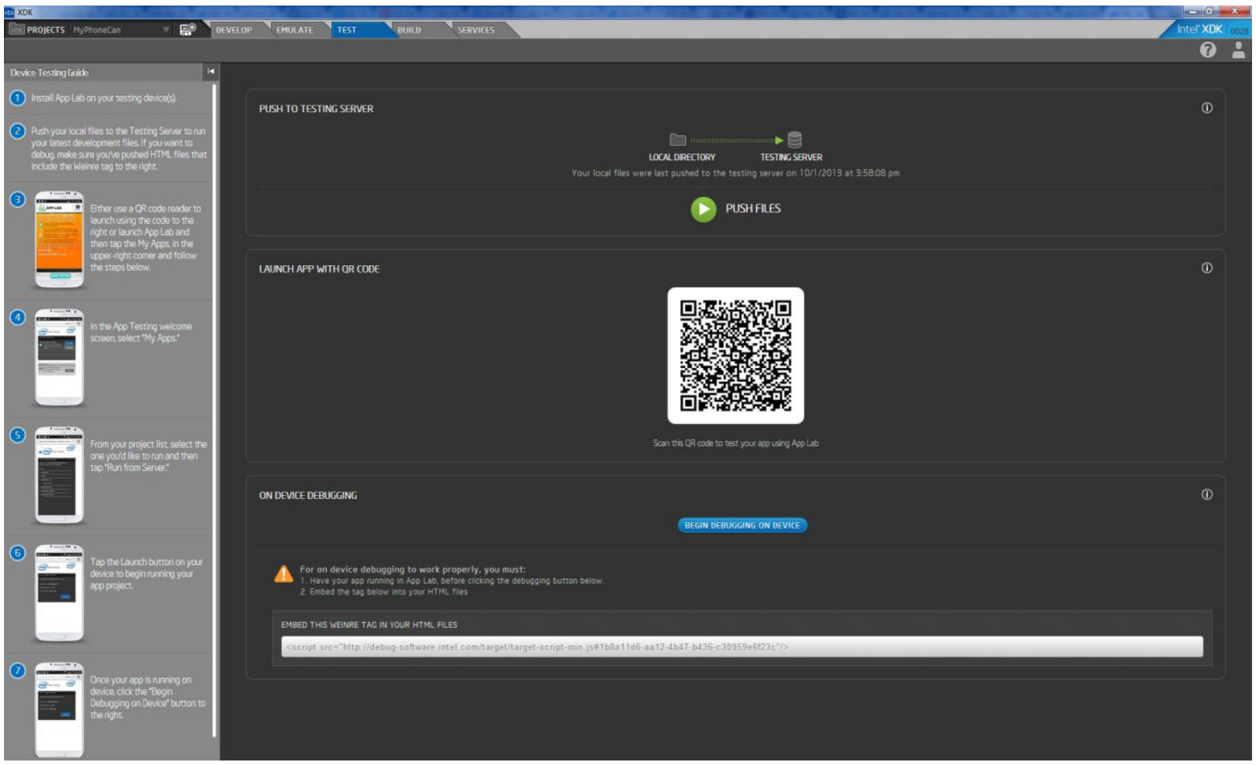
- Sync the project to the cloud by navigating to the Test Tab.



Please sync with our testing server:

Please sync your application with our testing server to take advantage of the on device testing features offered by the XDK.

- On your phone log into the App Lab app with your XDK credentials and follow the steps on the Test tab.
- Select the app to preview
- Alternatively you can scan the QR code displayed in the Test tab with your phone and it will launch the app on your phone.



Device Testing Guide

1. Install App Lab on your testing device(s).
2. Push your local files to the Testing Server to run your latest development files. If you want to debug, make sure you've pushed HTML files that include the Weinre tag to the right.
3. Either use a QR code reader to launch using the code to the right or launch App Lab and then tap the My Apps, in the upper-right corner and follow the steps below.
4. In the App Testing welcome screen, select "My Apps."
5. From your project list, select the one you'd like to run and then tap "Run from Server."
6. Tap the Launch button on your device to begin running your app project.
7. Once your app is running on device, click the "Begin Debugging on Device" button to the right.

PUSH TO TESTING SERVER

Your local files were last pushed to the testing server on 10/1/2013 at 3:58:08 pm

LAUNCH APP WITH QR CODE

Scan this QR code to test your app using App Lab

ON DEVICE DEBUGGING

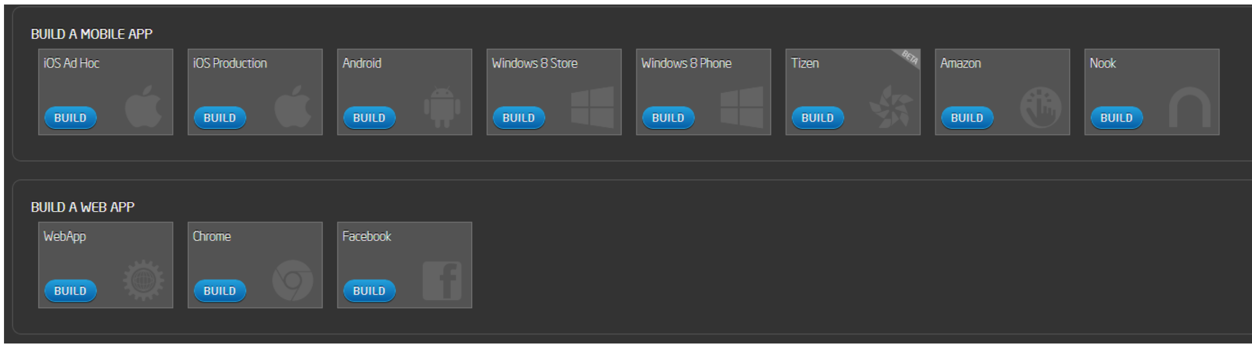
For on device debugging to work properly, you must:
1. Have your app running in App Lab, before clicking the debugging button below.
2. Embed the tag below into your HTML files

EMBED THIS WEINRE TAG IN YOUR HTML FILES

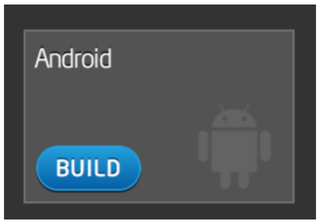
```
<script src="http://debug-software.intel.com/target/target-script-min.js#1b8a11d6-aa12-4b47-b436-c30959e6f23c"/>
```

4.5 Build a Package

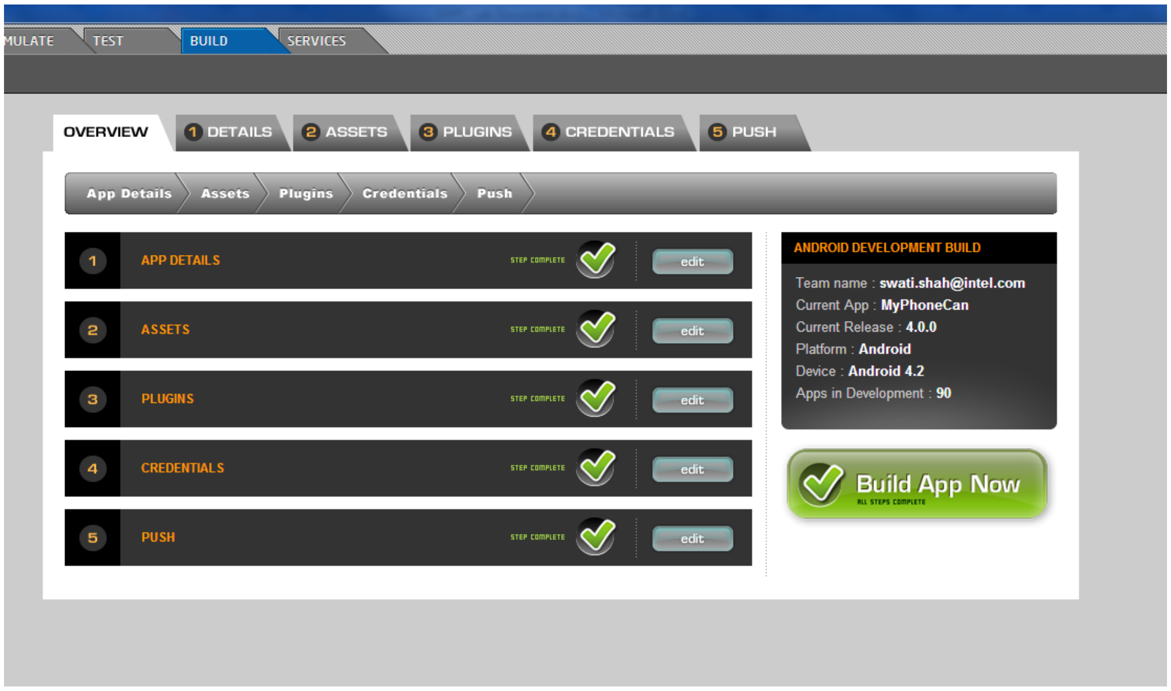
- Go to the Build tab



- Build an example Android* app by selecting Android button.



- Follow the steps.



ANDROID DEVELOPMENT BUILD

Team name : swati.shah@intel.com
Current App : MyPhoneCan
Current Release : 4.0.0
Platform : Android
Device : Android 4.2
Apps in Development : 90

- After the build is complete, you will receive an email pointing to the download link for your built app.
- Download the app and install it.

5.0 Task 2: “Rolling Can” accelerometer app

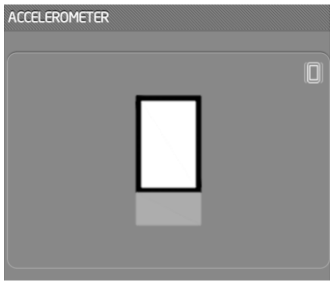
This part of the lab will show you how to start with an existing demo available in the XDK New. We will explore the accelerometer API using this example.

5.1 Start with a demo

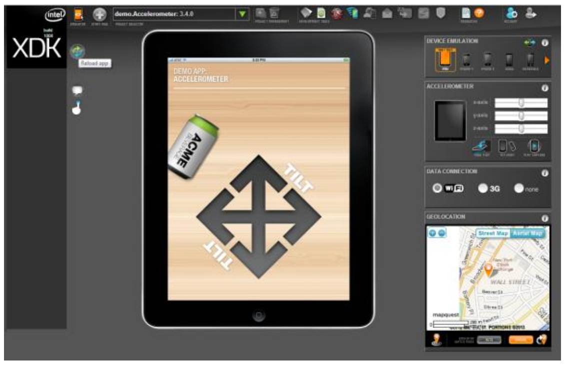
- In the Projects tab, select “Start a new project”
- Select “Start with a demo” and select “Rolling Can” demo

5.2 Preview in Emulator

- View the project in the emulator.
- Change devices to see how the app looks on different devices



- Drag around the device in the accelerometer tab in emulator to emulate accelerometer



5.3 Make changes in the code

- Uncomment the part of code in the index.html. You can find this piece of the code in lines 349-361. Uncomment the part with can_doug.png both at 351 & 360

```
     <img  src="images/can.png" alt=""  />
<!--
     <img  src="images/can_doug.png" alt=""  />
-->

     </div>
     <div  style="position:absolute;left:2px;top:51px;z-index:100;width:164px;height:237px;overflow:hidden" >

      <img id="img_sodalabel" src="images/generic-label.jpg" style="position:absolute;" alt="" />

<!--
      <img id="img_sodalabel" src="images/doug-label.jpg" style="position:absolute;" alt="" />
-->
```

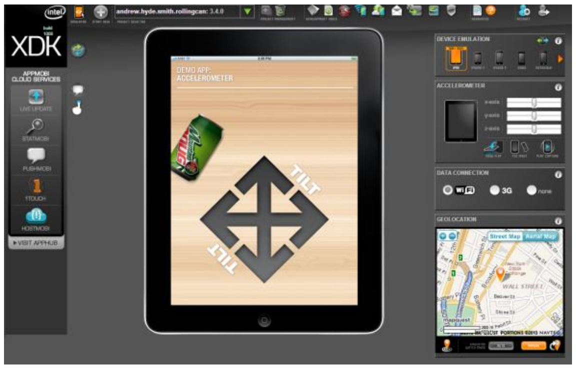
- And comment the code above it – the one with can.png at lines 349 and 358. It should change to look like:

```
<!--
     <img  src="images/can.png" alt=""  />
-->
     <img  src="images/can_doug.png" alt=""  />


     </div>
     <div  style="position:absolute;left:2px;top:51px;z-index:100;width:164px;height:237px;overflow:hidden" >
<!--
      <img id="img_sodalabel" src="images/generic-label.jpg" style="position:absolute;" alt="" />
-->

      <img id="img_sodalabel" src="images/doug-label.jpg" style="position:absolute;" alt="" />
```

- Preview the changes in Emulator by following instructions from 4.2



- You can notice the can image in the demo has changed

5.4 Make further changes to code

- Notice the intel.xdk.watchAcceleration takes in an option opt.
- Play around with increasing the opt.frequency. And preview the changes as visible in the emulator.

```
        var fail = function () { };

        var watchAccel = function () {


            var opt = {};
            opt.frequency = 5;
            //opt.frequency = 100;

            timer = intel.xdk.accelerometer.watchAcceleration(suc, opt);

        }
```

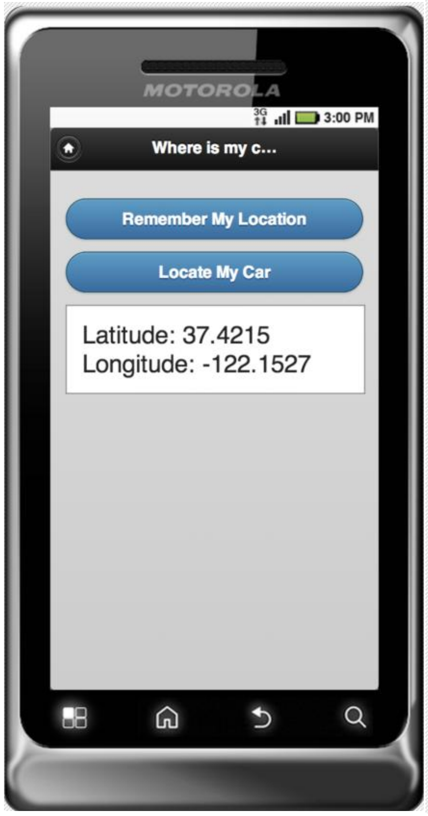
5.5 Preview in App Device

- Follow the instructions in 4.4

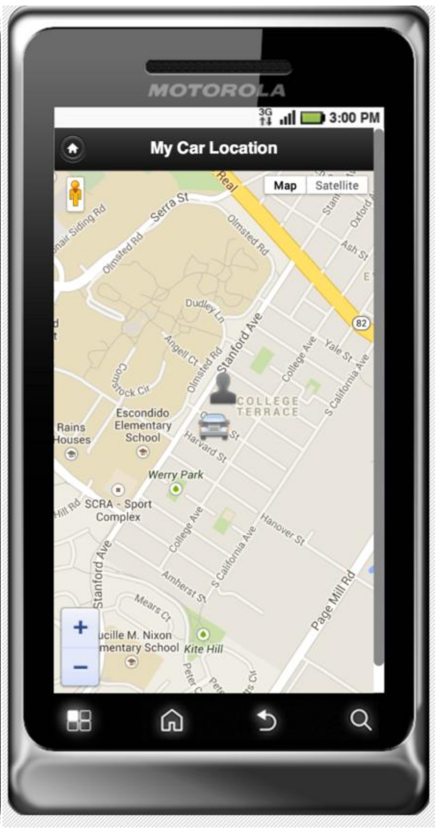
5.6 Build a Package

- Follow the instructions in 4.5

6.0 (Additional) Task 3: Develop “Where is my Car?” app



Latitude: 37.4215
Longitude: -122.1527



This part of the lab is focused on working with Cordova APIs.

Please download the basic skeleton for this app from github.

As part of the lab, you will fill in several missing parts of the app, which involve understanding how to work with some basic Javascript. You will make use of the tools you used in the previous task to test and debug as you develop.

This app will use Cordova's geolocation APIs to access your phone's GPS to remember where you parked your car.

6.1 Open "Where is my car" project

- The project already has the UI necessary for the app. There are two buttons as you open the app.
- There are two pages "homePage" and "mapPage"
- There are two buttons
- "Remember My Location" or "curLoc" saves the current location. We make use of HTML5's localStorage to save the location persistently. This enables us to save the location even when we close and reopen the app afresh.

- “Locate my Car” or “lastLoc” will open a map, showing the position of the car using the location we saved using “curLoc” (localStorage.lat & localStorage.long) and the user’s current location. The user’s current location updates as the user moves around.
- We use the Google Maps API to embed the map and add the markers.

6.2 Fill in the “TODO” items.

There are several blanks in the project corresponding to cordova geolocation APIs. Refer to Cordova’s documentation for the details of the API.

Look for TODO and follow through in the order suggested. FILL_IN is the parts where you will have to fill in the required APIs or inputs as necessary. To begin with the app does not load

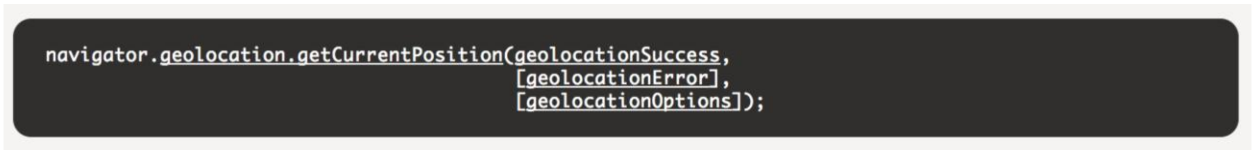
TODO 1: deviceready is an essential to all Cordova applications. It signals that all the Cordova APIs are loaded and the app is ready to use. Refer to the document for further details. Fill in the right event in place of FILL_IN. You will be able to see the home page with two buttons now.

TODO 2: Fill in the Cordova API for getting the current location. Refer to the document for exact details.

TODO 3: We need an API that updates as the user location changes. While similar to previous API call, this requires an API that watches for user location changes. Imagine moving around with your phone trying to locate your car, this API should watch for user changes and be invoked every time it changes.

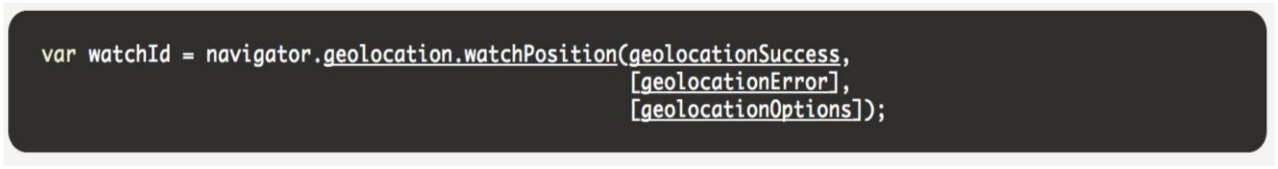
Note: There are two different Cordova APIs we primarily use.

- getCurrentLocation – gets the current location



```
navigator.geolocation.getCurrentPosition(geolocationSuccess,
                                         [geolocationError],
                                         [geolocationOptions]);
```

- watchPosition – watches for changes to user location and is called

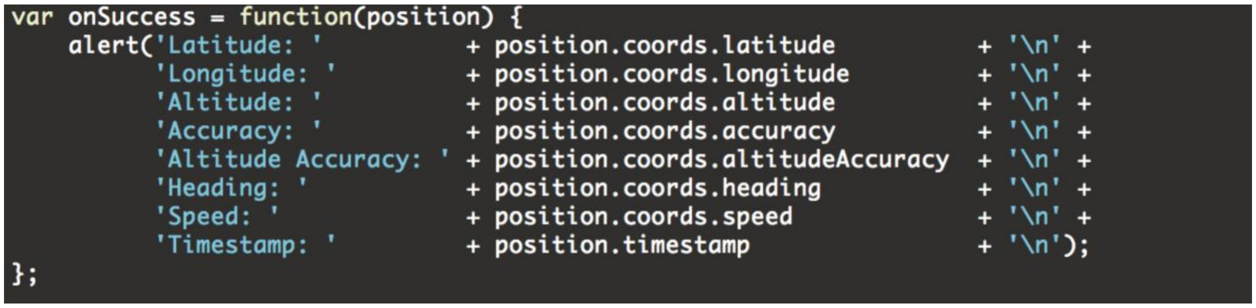


```
var watchId = navigator.geolocation.watchPosition(geolocationSuccess,
                                                  [geolocationError],
                                                  [geolocationOptions]);
```

TODO 4: Uncomment the addPin function call and fill in the latitude and longitude values with the car’s latitude and longitude values. This change adds a car marker on the map corresponding to the car location.

TODO 5: Uncomment the addPin function call and fill in the latitude and longitude values with the user’s current location’s latitude and longitude values. This change adds a user marker on the map corresponding to the car location.

Note: This is called within the success handler of the Cordova API call. So, you will need to find the latitude and longitude based on the "position" variable.



```
var onSuccess = function(position) {
    alert('Latitude: '          + position.coords.latitude          + '\n' +
          'Longitude: '         + position.coords.longitude         + '\n' +
          'Altitude: '          + position.coords.altitude          + '\n' +
          'Accuracy: '          + position.coords.accuracy          + '\n' +
          'Altitude Accuracy: ' + position.coords.altitudeAccuracy  + '\n' +
          'Heading: '           + position.coords.heading           + '\n' +
          'Speed: '             + position.coords.speed             + '\n' +
          'Timestamp: '         + position.timestamp                + '\n');
};
```

6.3 Preview in emulator

Verify all the code changes with emulator. You can emulate geolocation APIs in the XDK in the emulator.

Follow instructions in 4.3

Note: You can also add alert or console.log messages to print out debug messages. See examples in the project

6.4 Preview on Device

Follow the instructions in 4.4

6.5 Build a package

Follow the instructions in 4.5

Note: You will need to set the right permission in the build UI. Open the Details tab in the Build page to set permissions for geolocation access.

7.0 Important links

Where to download XDK New: http://xdk-software.intel.com/download.html

Tutorials & Documentation: http://software.intel.com/en-us/html5/tools

How to use XDK New: http://www.html5dev-software.intel.com/en-us/xdkdocs/index.htm

http://www.html5dev-software.intel.com/en-us/xdkdocs/intel-xdk-overview.html

App Designer: http://www.html5dev-software.intel.com/en-us/xdkdocs/app_designer/index.htm

Device Emulator: http://www.html5dev-software.intel.com/en-us/xdkdocs/dev_emulator/index.htm

Intel XDK APIs: http://www.html5dev-software.intel.com/documentation/jsAPI/index.html

(Replace AppMobi with intel.xdk)

Cordova APIs: http://cordova.apache.org/docs/en/2.9.0/