

Exploring ngCordova

In this lab, we will explore 6 ngCordova plugins just to get some sense on how they work. It is very easy to implement ngCordova plugins. The 6 that we will be exploring today will be: Dialogs, Device, Network, Vibration, BarcodeScanner, and Camera.

Initial setup

As usual, change directory to your **Desktop** (or any folder in which you want to store your project in), and do `ionic start ngCordovaApp sidemenu`. This time, we will be using the sidemenu starter template.

We will first setup ngCordova. Download the zip file here: <https://github.com/driftyco/ng-cordova/archive/master.zip>. Locate `ng-cordova.min.js` in the **dist** folder and place it into your js folder (`www/js/ng-cordova.min.js`).

Include `ng-cordova.min.js` in your `index.html` file **before** `cordova.js` and after your AngularJS / Ionic file (since ngCordova depends on AngularJS).

```
...
<!-- ionic/angularjs js -->
<script src="lib/ionic/js/ionic.bundle.js"></script>

<!-- cordova script (this will be a 404 during development) -->
<script src="js/ng-cordova.min.js"></script>
<script src="cordova.js"></script>

<!-- your app's js -->
<script src="js/app.js"></script>
<script src="js/controllers.js"></script>
...
```

When you have included the js file, we will need to inject ngCordova as an Angular dependency. Head to `app.js` and include `ngCordova` as a dependency in your angular module.

```
angular.module('starter', ['ionic', 'ngCordova']) // note that starter.controllers is re
moved
```

Next, head to `controllers.js` and change the first line to:

```
angular.module('starter')
```

Before we proceed, let's set up the sidemenus.

In your `app.js`, replace the whole thing with this. The important part is the routing:

```
// Ionic Starter App

// angular.module is a global place for creating, registering and retrieving Angular modules
// 'starter' is the name of this angular module example (also set in a <body> attribute in index.html)
// the 2nd parameter is an array of 'requires'
// 'starter.controllers' is found in controllers.js
angular.module('starter', ['ionic', 'ngCordova'])

.run(function($ionicPlatform) {
  $ionicPlatform.ready(function() {
    // Hide the accessory bar by default (remove this to show the accessory bar above the keyboard
    // for form inputs)
    if (window.cordova && window.cordova.plugins.Keyboard) {
      cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
      cordova.plugins.Keyboard.disableScroll(true);
    }
    if (window.StatusBar) {
      // org.apache.cordova.statusbar required
      StatusBar.styleDefault();
    }
  });
})

.config(function($stateProvider, $urlRouterProvider) {
  $stateProvider

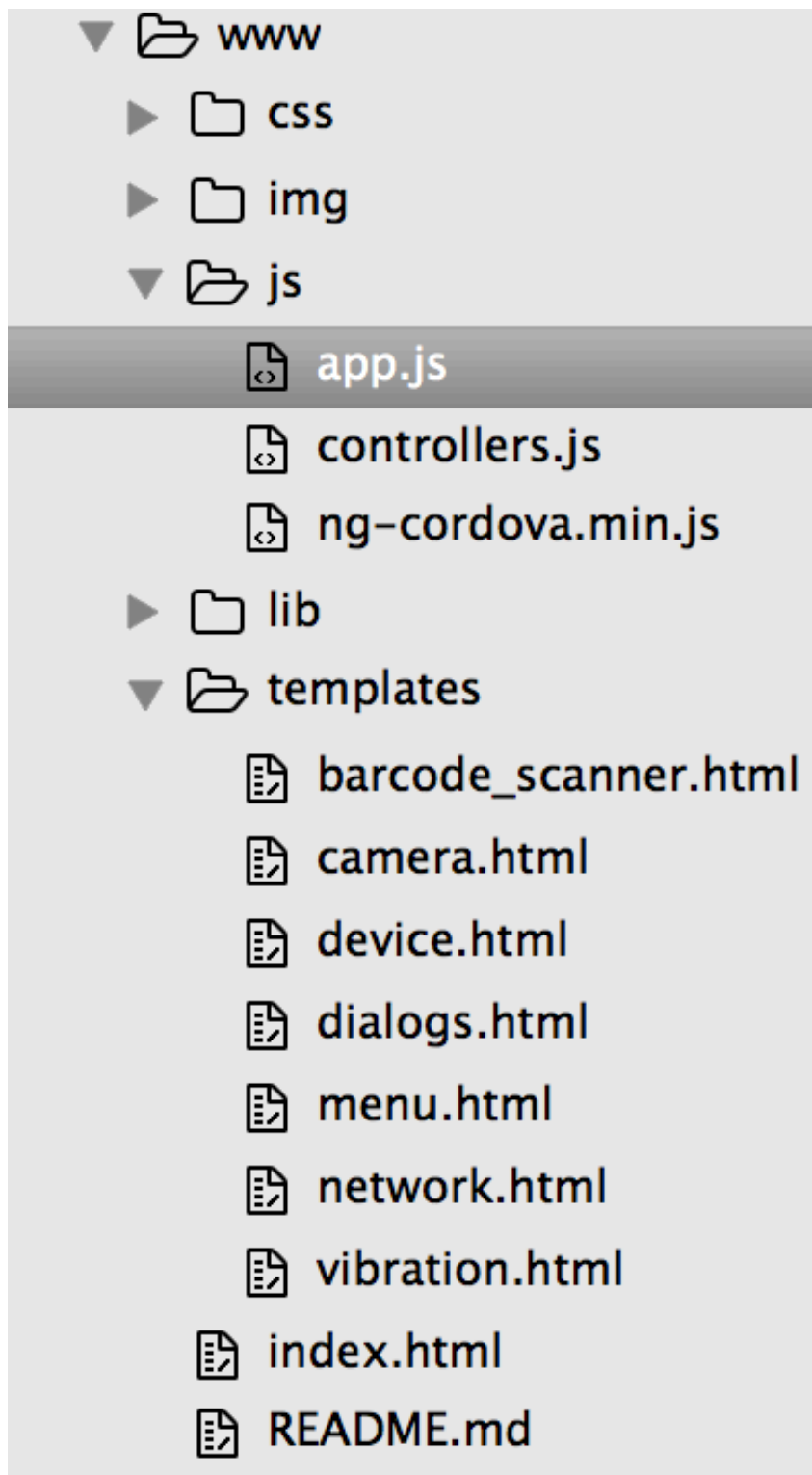
    .state('app', {
      url: '/app',
      abstract: true,
      templateUrl: 'templates/menu.html',
      controller: 'AppCtrl'
    })
    .state('app.dialogs', {
      url: '/dialogs',
      views: {
        'menuContent': {
          templateUrl: 'templates/dialogs.html',
          controller: 'DialogsCtrl'
        }
      }
    })
  })
```

```

    }
  }).state('app.device', {
    url: '/device',
    views: {
      'menuContent': {
        templateUrl: 'templates/device.html',
        controller: 'DeviceCtrl'
      }
    }
  })
  }).state('app.network', {
    url: '/network',
    views: {
      'menuContent': {
        templateUrl: 'templates/network.html',
        controller: 'NetworkCtrl'
      }
    }
  })
  }).state('app.vibration', {
    url: '/vibration',
    views: {
      'menuContent': {
        templateUrl: 'templates/vibration.html',
        controller: 'VibrationCtrl'
      }
    }
  })
  }).state('app.barcode_scanner', {
    url: '/barcode_scanner',
    views: {
      'menuContent': {
        templateUrl: 'templates/barcode_scanner.html',
        controller: 'BarcodeScannerCtrl'
      }
    }
  })
  }).state('app.camera', {
    url: '/camera',
    views: {
      'menuContent': {
        templateUrl: 'templates/camera.html',
        controller: 'CameraCtrl'
      }
    }
  });
  // if none of the above states are matched, use this as the fallback
  $urlRouterProvider.otherwise('/app/dialogs');
});

```

Next, create these few templates and *remove* the rest:



We now have the following new files except for one:

- barcode_scanner.html
- camera.html
- device.html
- dialogs.html
- menu.html (existing file)
- network.html
- vibration.html

Replace your `menu.html` with the following:

```
<ion-side-menus enable-menu-with-back-views="false">
  <ion-side-menu-content>
    <ion-nav-bar class="bar-stable">
      <ion-nav-back-button>
      </ion-nav-back-button>

      <ion-nav-buttons side="left">
        <button class="button button-icon button-clear ion-navicon" menu-toggle="left">
        </button>
      </ion-nav-buttons>
    </ion-nav-bar>
    <ion-nav-view name="menuContent"></ion-nav-view>
  </ion-side-menu-content>

  <ion-side-menu side="left">
    <ion-header-bar class="bar-stable">
      <h1 class="title">ngCordova Plugins</h1>
    </ion-header-bar>
    <ion-content>
      <ion-list>
        <ion-item menu-close href="#/app/dialogs">
          Dialogs
        </ion-item>
        <ion-item menu-close href="#/app/device">
          Device
        </ion-item>
        <ion-item menu-close href="#/app/network">
          Network
        </ion-item>
        <ion-item menu-close href="#/app/vibration">
          Vibration
        </ion-item>
        <ion-item menu-close href="#/app/barcode_scanner">
          Barcode Scanner
        </ion-item>
        <ion-item menu-close href="#/app/camera">
          Camera
        </ion-item>
      </ion-list>
    </ion-content>
  </ion-side-menu>
</ion-side-menus>
```

The remaining 6 html files will have the following template layout:

```
<ion-view title="TITLE_HERE">
  <ion-content class="has-header padding">
    <h1>Placeholder</h1>
  </ion-content>
</ion-view>
```

Change TITLE_HERE to the respective plugin names, or any names that you want. For example, Barcode Scanner, Camera, Device, Dialogs, Network, and Vibration.

Ionic.io (<http://ionic.io/>)

Next, we would like to introduce you to Ionic.io, a platform as a service which allows you to upload your ionic app online and allow users, testers, friends or family members to view your app on an actual device without needing to download your app from the App Store. You do not need to upload your app (or even purchase the developer's account too). One drawback to this is that everyone will need to download an app called **Ionic View**.

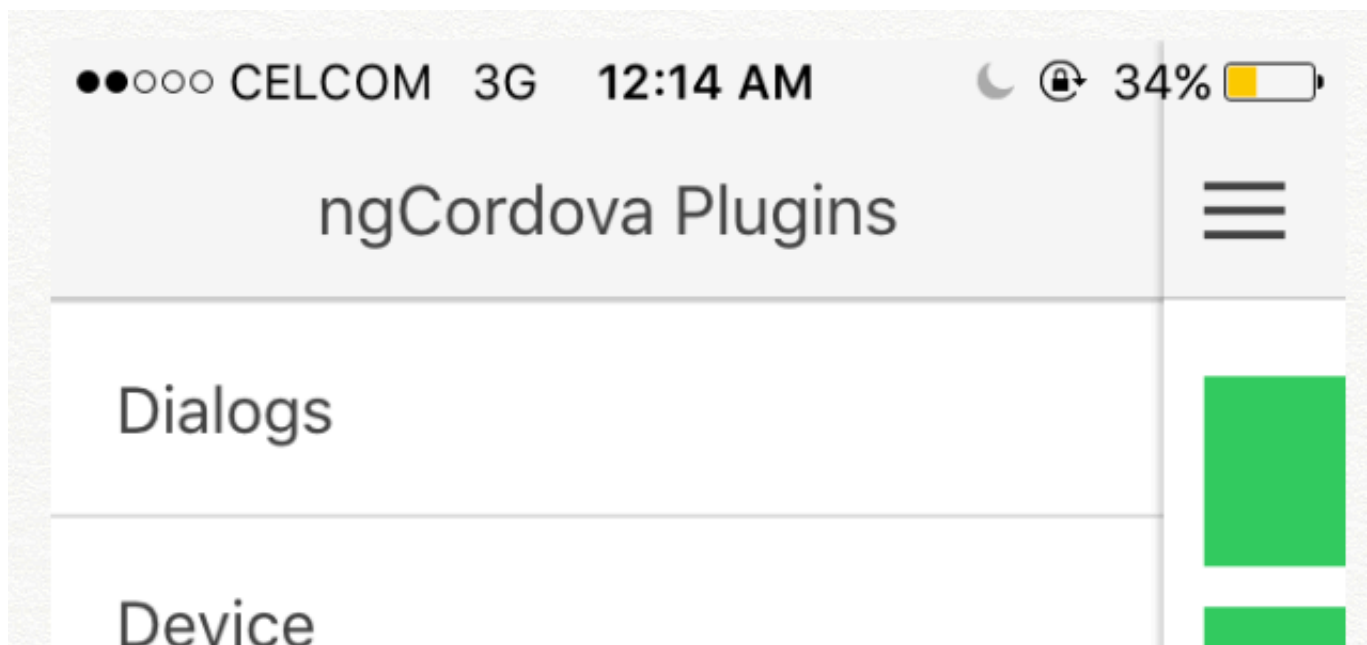
The reason why we need this is that *most* of ngCordova plugins can only be tested on an actual mobile device (not a simulator) as it makes use of the native features.

1. Register an account on ionic.io
2. In your terminal, type `ionic login` and type in your account credentials.
3. Now do `ionic upload`. You have uploaded your app onto the cloud.

Isn't it simple?

Now, download Ionic View on your mobile (from the App Store, or Google Play Store) and login with the same account info. You will be able to view your app through Ionic View. You can also make changes to your code, then do `ionic upload` again, and you would be able to see the changes on your phone.

However, this is not a livereload function. You need to redownload your app (or update your app) **through** Ionic View in order for the changes to be reflected.



Network

Vibration

Barcode Scanner

Camera

Plugin 1: Dialogs

Reference: <http://ngcordova.com/docs/plugins/dialogs/>

Type this to install the plugin:

```
cordova plugin add cordova-plugin-dialogs
```

In `dialogs.html` , do this:

```
<ion-view view-title="Dialogs">
  <ion-content class="has-header padding">
    <button class="button button-full button-balanced" ng-click="alert()">Alert</button>
    <button class="button button-full button-balanced" ng-click="confirm()">Confirm</button>
    <button class="button button-full button-balanced" ng-click="prompt()">Prompt</button>
    <button class="button button-full button-balanced" ng-click="beep()">Beep</button>
  </ion-content>
</ion-view>
```

Finally, in `DialogsCtrl` in `controllers.js` , add these:

```
...
.controller('DialogsCtrl', function($cordovaDialogs, $scope) {
  // REF: http://ngcordova.com/docs/plugins/dialogs/

  // alert(message, title, buttonName)
  $scope.alert = function () {
    $cordovaDialogs.alert('Wow!', 'Alert Title', 'Dismiss');
  };

  // confirm(message, title, buttonArray)
  $scope.confirm = function () {
    $cordovaDialogs.confirm('Are you sure?', 'Confirmation', ['Yes', 'No'])
      .then(function(buttonIndex) {
        // no button = 0, 'Yes' = 1, 'No' = 2
        $cordovaDialogs.alert('You selected button ' + buttonIndex, 'Response');
      });
  };

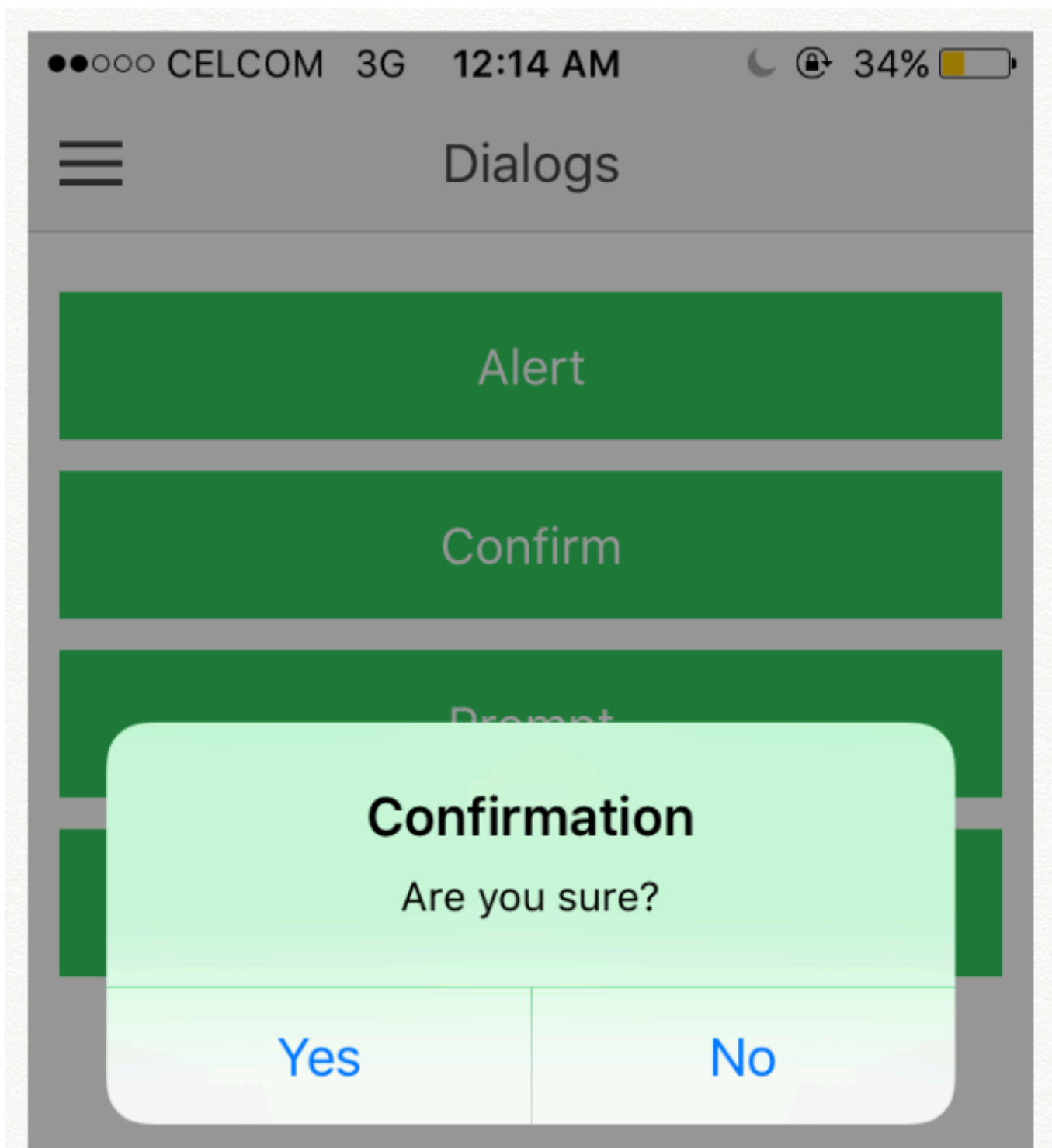
  // prompt(message, title, buttonArray, defaultText)
  $scope.prompt = function () {
    $cordovaDialogs.prompt('Please Login', 'Login', ['Login', 'Cancel'], 'Change me')
      .then(function(result) {
        var input = result.input1;
        // no button = 0, 'OK' = 1, 'Cancel' = 2
        var btnIndex = result.buttonIndex;

        $cordovaDialogs.alert('You selected button number ' + btnIndex + ' and entered ' +
```



```
+ input, 'Response');  
    });  
};  
  
// beep(repetitions)  
$scope.beep = function () {  
    // beep 1 time  
    $cordovaDialogs.beep(1);  
};  
})  
...
```

Do `ionic upload` and view your app through `Ionic View`. Play around with the buttons.



Plugin 2: Device

Reference: <http://ngcordova.com/docs/plugins/device/>

Type this to install the plugin:

```
ccordova plugin add cordova-plugin-device
```

In `device.html` , do this:

```
<ion-view title="Device Info">
  <ion-content class="has-header padding">
    <button class="button button-full button-balanced" ng-click="deviceModel()">Device Model</button>
    <button class="button button-full button-balanced" ng-click="devicePlatform()">Device Platform</button>
    <button class="button button-full button-balanced" ng-click="deviceUUID()">Device UUID</button>
    <button class="button button-full button-balanced" ng-click="deviceVersion()">Device Version</button>
    <button class="button button-full button-balanced" ng-click="cordovaVersion()">Cordova Version</button>
    <button class="button button-full button-balanced" ng-click="device()">Device Info</button>
  </ion-content>
</ion-view>
```

```
<h2>{{ results }}</h2>
</ion-content>
</ion-view>
```

Finally, in `DeviceCtrl` in `controllers.js`, add these:

```
...
.controller('DeviceCtrl', function($cordovaDevice, $scope) {
  // REF: http://ngcordova.com/docs/plugins/device/

  $scope.device = function () {
    $scope.results = $cordovaDevice.getDevice();
  };

  $scope.cordovaVersion = function () {
    $scope.results = $cordovaDevice.getCordova();
  };

  $scope.deviceModel = function() {
    $scope.results = $cordovaDevice.getModel();
  };

  $scope.devicePlatform = function () {
    $scope.results = $cordovaDevice.getPlatform();
  };

  $scope.deviceUUID = function(){
    $scope.results = $cordovaDevice.getUUID();
  };

  $scope.deviceVersion = function () {
    $scope.results = $cordovaDevice.getVersion();
  };
})
...
```

Do `ionic upload` and view your app through `Ionic View`. Play around with the buttons.



Device Info

Device Model

Device Platform

Device UUID

Device Version

Cordova Version

Device Info

iOS

Plugin 3: Network

Reference: <http://ngcordova.com/docs/plugins/network/>

Type this to install the plugin:

```
cordova plugin add cordova-plugin-network-information
```

In `network.html` , do this:

```
<ion-view title="Network">
  <ion-content class="has-header padding">
    <h3>Type: {{ type }}</h3>
    <h3>Is Online: {{ isOnline }}</h3>
    <h3>Is Offline: {{ isOffline }}</h3>
  </ion-content>
</ion-view>
```

Finally, in `NetworkCtrl` in `controllers.js` , add these:

```
...
.controller('NetworkCtrl', function($cordovaNetwork, $scope, $rootScope, $cordovaDialogs) {
  // REF: http://ngcordova.com/docs/plugins/network/

  $scope.type = $cordovaNetwork.getNetwork();

  $scope.isOnline = $cordovaNetwork.isOnline();

  $scope.isOffline = $cordovaNetwork.isOffline();

  // listen for Online event
  $rootScope.$on('$cordovaNetwork:online', function(event, networkState){
    var onlineState = networkState;

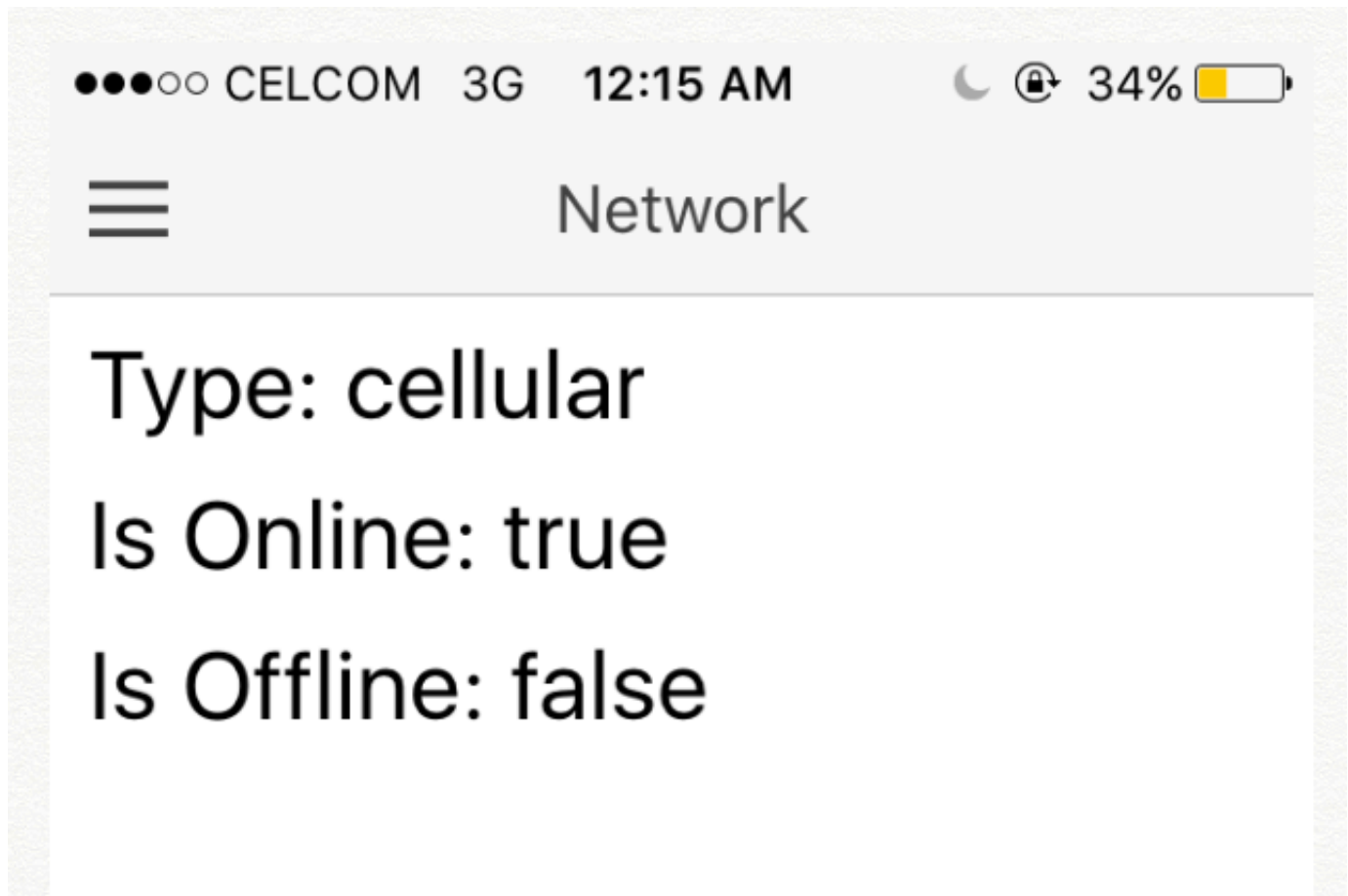
    $cordovaDialogs.alert('Online event is fired. Status: ' + onlineState, 'Fired!');
  });

  // listen for Offline event
  $rootScope.$on('$cordovaNetwork:offline', function(event, networkState){
    var offlineState = networkState;

    $cordovaDialogs.alert('Offline event is fired. Status: ' + onlineState, 'Fired!');
  });
});
```

```
})  
...
```

Do `ionic upload` and view your app through `Ionic View`.



Plugin 4: Vibration

Reference: <http://ngcordova.com/docs/plugins/vibration/>

Type this to install the plugin:

```
cordova plugin add cordova-plugin-vibration
```

In `vibration.html`, do this:

```
<ion-view title="Vibration">  
  <ion-content class="has-header padding">  
    <button class="button button-full button-balanced" ng-click="vibrate()">Vibrate</but  
ton>  
  </ion-content>  
</ion-view>
```

Finally, in `VibrationCtrl` in `controllers.js`, add these:

```
...  
.controller('VibrationCtrl', function($cordovaVibration, $scope) {  
  // REF: http://ngcordova.com/docs/plugins/vibration/  
  
  $scope.vibrate = function () {  
    $cordovaVibration.vibrate(500);  
  };  
})  
...
```

Do `ionic upload` and view your app through `Ionic View`. Play around with the vibration button. Your phone will vibrate when you press the button.

Plugin 5: Barcode Scanner

Reference: <http://ngcordova.com/docs/plugins/barcodeScanner/>

Type this to install the plugin:

```
cordova plugin add https://github.com/phonegap/phonegap-plugin-barcodescanner.git
```

In `barcode_scanner.html`, do this:

```
<ion-view title="Barcode Scanner">  
  <ion-content class="has-header padding">  
    <button class="button button-full button-balanced" ng-click="scan()">Scan</button>  
    <button class="button button-full button-stable" ng-click="reset()">Reset</button>  
  
    <div ng-show="results">  
      <h2>Results: {{ results }}</h2>  
  
      <button class="button button-full button-energized" ng-click="lookup()">Lookup</button>  
    </div>  
  </ion-content>  
</ion-view>
```

Finally, in `BarcodeScannerCtrl` in `controllers.js`, add these:

```

...
.controller('BarcodeScannerCtrl', function($cordovaBarcodeScanner, $scope, $cordovaDialogs) {
    // REF: http://ngcordova.com/docs/plugins/barcodeScanner/

    $scope.scan = function () {
        $cordovaBarcodeScanner.scan().then(function (imageData) {
            // Success! Barcode data is here
            $scope.results = imageData;
        }, function (err) {
            // An error occurred. Show a message to the user
            $cordovaDialogs.alert('Error scanning: ' + err, 'Error!');
        });
    };

    $scope.lookup = function () {
        window.open("http://www.upcindex.com/" + $scope.results.text, "_system");
    };

    $scope.reset = function () {
        $scope.results = null;
    };
})
...

```

Do `ionic upload` and view your app through `Ionic View`. Play around with the buttons and scan a random barcode. (If you do not have one, try searching for one on Google and use your phone to scan the code on your computer screen.)



Barcode Scanner

Scan

Reset

Results:

```
{"cancelled":0,"text":"9556113002727","format":"EAN_13"}
```

Lookup

Plugin 6: Camera

Type this to install the plugin:

```
cordova plugin add cordova-plugin-camera
```

In `camera.html` , do this:

```
<ion-view title="Camera">
  <ion-content class="has-header padding">
    <div class="row">
      <div class="col">
        <button class="button button-balanced" ng-click="takePicture()">Take Picture</button>
      </div>
      <div class="col">
        <button class="button button-balanced" ng-click="selectPicture()">Select Picture</button>
      </div>
    </div>

    <div class="card">
      <div class="item item-image">
        
      </div>
    </div>
  </ion-content>
</ion-view>
```

Finally, in `CameraCtrl` in `controllers.js` , add these:

```
...
.controller('CameraCtrl', function($cordovaCamera, $scope, $cordovaDialogs) {
  // REF: http://ngcordova.com/docs/plugins/camera/

  $scope.takePicture = function () {
    var options = {
      quality : 100,
      destinationType : Camera.DestinationType.DATA_URL,
      sourceType : Camera.PictureSourceType.CAMERA,
      allowEdit : true,
      encodingType: Camera.EncodingType.JPEG,
      targetWidth: 300,
      targetHeight: 300,
      popoverOptions: CameraPopoverOptions,
      saveToPhotoAlbum: false,
```

```

        correctOrientation: true
    };

    $cordovaCamera.getPicture(options).then(function (imageData) {
        // Success! Image data is here
        $scope.imgSrc = "data:image/jpeg;base64," + imageData;
    }, function (err) {
        $cordovaDialogs.alert('An error occurred: ' + err, 'Error!');
    });
};

$scope.selectPicture = function () {
    var options = {
        destinationType: Camera.DestinationType.FILE_URI,
        sourceType: Camera.PictureSourceType.PHOTOLIBRARY
    };

    $cordovaCamera.getPicture(options).then(function (imageUri) {
        // Success! Image data is here
        $scope.imgSrc = imageUri;
    }, function (err) {
        $cordovaDialogs.alert('An error occurred: ' + err, 'Error!');
    });
};
});
...

```

Do `ionic upload` and view your app through `Ionic View`. Play around with the buttons.



Camera

Take Picture

Select Picture

```
on-view title="Barcode Scanner">
<ion-content class="has-header padding">
  <button class="button button-full butt
    click="scan()">Scan</button>
  <button class="button button-full butt
    "reset()">Reset</button>

  <div ng-show="results">
    <h2>Results: {{ results }}</h2>

    <button class="button button-full bu
      click="lookup()">Lookup</button>
  </div>
</ion-content>
</ion-view>
```

Too simple?

Pick some plugins and then implement them in your current app.

A list of plugins is available here: <http://ngcordova.com/docs/plugins/>

1. Add a new template
2. Modify `app.js` to add a new route
3. Modify `menu.html` to add a new link
4. Install the plugin
5. Add a new controller and the respective functions for the view
6. `ionic upload`
7. Test on your phone
8. Repeat for another plugin.

There are more than 50 plugins for you to test! Do as much as possible.