



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



Module 4. Angular and Ionic
Hybrid App (Web and Mobile)



<https://www.joshmorony.com/building-a-crud-ionic-2-application-with-firebase-angularfire/>

<https://www.djamware.com/post/585476a280aca7060f443064/ionic-2-firebase-crud-example-part-1>

Task1 . AngularJS with Firebase

Example app: ChatApp single page application with AngularJS framework and a backend Firebase.

Firebase is a backend service with web socket and store information. It is a Real time database, that means that at the same time a mechanism to real time synchronizing the various actor of the application. **AngularFire** is the service offer for AngularJS applications.

Starting the app. With needed libraries via CDN:



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



Index.html

```
<!DOCTYPE html>
<html>
<head>

  <!--Angularare -->
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.0-beta.2/angular.m
in.js"></script>
  <!-- Firebase -->
  <script
src="https://cdn.firebase.com/js/client/2.0.4/firebase.js"></script>
  <!-- AngularFire -->
  <script
src="https://cdn.firebase.com/libs/angularfire/0.9.2/angularfire.min.js"></
script>

  <!--App -->
  <script src="js/app.js"></script>

</head>
<body ng-app="chatApp">
  {{ "AngularJS Loaded" }}

  <div ng-controller="chatController" >
    <p>Nome : <input type="text" ng-model="newmessage.user"></p>
    <p>Messaggio : <input type="text" ng-model="newmessage.text"></p>
    <button ng-click="inzerisci(newmessage)">Invia</button>

    <ul>
      <li ng-repeat="message in messages">
        {{message.user}} dice :{{message.text}}
      </li>
    </ul>
  </div>

</body>
</html>
```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



app.js

```
var app = angular.module('chatApp', ['firebase']);

    app.controller('chatController', ['$scope', 'Message',
function($scope, Message){

    $scope.user="Guest";

    $scope.messages= Message.all;

    $scope.inserisci = function(message){
        Message.create(message);
    };

}]);

app.factory('Message', ['$firebase',
function($firebase) {
    var ref = new
Firebase('https://luminous-heat-165.firebaseio.com');
    var messages = $firebase(ref.child('messages')).$asArray();

    var Message = {
        all: messages,
        create: function (message) {
            return messages.$add(message);
        },
        get: function (messageId) {
            return
$firebase(ref.child('messages').child(messageId)).$asObject();
        },
        delete: function (message) {
            return messages.$remove(message);
        }
    };

    return Message;
}
]);
```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



Task2. AngularJS and IONIC

Ionic is a front-end SDK for building cross-platform mobile apps. Built on top of Angular, Ionic also provides a platform for integrating services like push notifications and analytics.

Previous Knowledge

Basics of JavaScript and AngularJS. Ionic is completely made up of Angular directives and components and uses its popular ui router for SPA.

Set Up



<https://nodejs.org/en/>

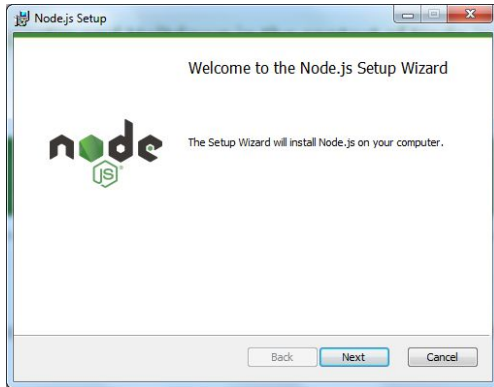
The screenshot shows the Node.js website with a dark navigation bar containing the Node.js logo and links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. Below the navigation bar, there is a paragraph describing Node.js as a JavaScript runtime built on Chrome's V8 JavaScript engine. A light green box highlights the article "Spectre and Meltdown in the context of Node.js." Underneath, there are two buttons for downloading Node.js for Windows (x64): "8.9.4 LTS Recommended For Most Users" and "9.5.0 Current Latest Features". Below these buttons are links for "Other Downloads | Changelog | API Docs". A link to "Or have a look at the LTS schedule." is also present. At the bottom, there is a link to "Sign up for Node.js Everywhere, the official Node.js Weekly Newsletter." The footer contains the Linux Foundation logo and copyright information: "© 2018 Node.js Foundation. All Rights Reserved. Portions of this site originally © 2018 Joyent."



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



```
npm install -g cordova ionic
```

The `-g` flag installs both cordova and ionic globally so we can access it from any where in our machines. It also adds them to our PATH so it won't be a surprise to our CLI. To confirm installation:

```
ioniv -v && cordova -v
```

Create the project

A new project in ionic can be created using the ionic CLI command. It will scaffold the app's file and download it's dependencies. You can supply arguments to the CLI command to alter the type of template generated.

```
ionic start {apname} {template}
```

That is the syntax for a new project in Ionic. The apname is any name of your choice and template is one of the Ionic supported template. It can also be a GitHub URL to a custom template.

There is nothing wrong with tradition - let us create a Todo app to better see what Ionic is made of.

Example of a Todo App

```
ionic start todo-app blank
```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



We are using the blank template for simplicity in as much as we can use other types of templates like tabs, side menus, etc. The command will generate a new project for us and also install some dependencies.

```
|— hooks          // custom cordova hooks to execute on specific
commands
|— platforms     // iOS/Android specific builds will reside here
|— plugins       // where your cordova/ionic plugins will be installed
|— resources     // icon and splash screen
|— scss          // scss code, which will output to www/css/
|— www          // application - JS code and libs, CSS, images, etc.
|-----css      //customs styles
|-----img      //app images
|-----js       //angular app and custom JS
|-----lib      //third party libraries
|-----index.html //app master page
|— bower.json    // bower dependencies
|— config.xml   // cordova configuration
|— gulpfile.js  // gulp tasks
|— ionic.project // ionic configuration
|— package.json // node dependencies
```

Configuring Platforms

Cordova gives you the power to write once and run on many platforms (iOS, Android, Blackberry, Windows, etc) but you have to explicitly tell Cordova (with Ionic) which platform should be included in a given project. For our Todo app, we will be trying with two platforms (iOS and Android).

```
cd todo-app
ionic platform add ios
ionic platform add android
```

Previewing the App

We can preview our app in the browser using Ionic. This is not recommended though (about that later) but we can use it to preview (not really test) our basic Todo application.

To see what our blank app looks like, run the below command in the CLI on the todo-app directory:



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



```
ionic serve
```

The command will open up a browser at `http://localhost:8100`. The browser screen is too large to preview a mobile app and for that reason, Ionic added `--lab` as an option to running the app in the browser so as to preview the app better as it would be seen in Android and iOS.

```
ionic serve --lab
```

Developing the project

Ionic is a basically a SPA web app, we need a master `index.html` though we don't need routes as the Todo app is just a one view app.

We should first update our app's `www/index.html` and `www/js/app.js` to make sure the structure looks like the one below.

`www/index.html`

```
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1, maximum-scale=1,
user-scalable=no, width=device-width">
    <title></title>

    <link href="lib/ionic/css/ionic.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
    <!-- ionic/angularjs js -->
    <script src="lib/ionic/js/ionic.bundle.js"></script>

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

    <!-- your app's js -->
    <script src="js/app.js"></script>
  </head>
  <!-- your app's js -->
  <body ng-app="todo-app">

    <ion-pane>
      <ion-header-bar class="bar-dark">
```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



```
<h1 class="title">Scotch-Todo</h1>
</ion-header-bar>
<ion-content>
</ion-content>
</ion-pane>
</body>
</html>
```

[www/js/app.js](#)

```
//setup angular
var app = angular.module('todo-app', ['ionic']);
```

Ionic creates a bundle of Angular and required dependencies (angular-ui-router, angular-animate, angular-sanitize, etc) bundled in one file as ionic.bundle.js.

Those are the files we will work with throughout this tutorial. Before continuing, let us define some tasks:

- Setup Local Storage service to store todo data
- Create a main controller to interact with our view
- Update our index.html to work with the controller

A Local Storage Service

HTML 5 which is ionic's best friend comes with a lot of storage options. Local Storage is one of them and it suits our app more because of all the options provided, it has the ability to persist data even when the browser or app is closed.

Angular on the other hand has a service written to wrap the Local Storage API and is available on GitHub as angular-local-storage. Fortunately, Ionic uses bower to manage dependencies so installing angular-local-storage takes this CLI command:

```
bower install angular-local-storage --save
```

The command will download and add angular-local-storage to our app libraries. Just below where we added the ionic.bundle.js include angular-local-storage as seen below:

```
<script
src="lib/angular-local-storage/dist/angular-local-storage.js"></script>
```




Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



Tell angular it is a dependency by injecting it in the angular app in the www/js/app/js file like this:

```
var app = angular.module('scotch-todo', ['ionic', 'LocalStorageModule']);
```

It is a good practice to add a prefix to our stored entities so as avoid being overwritten in future. Add a config angular block to set the Local Storage prefix

```
app.config(function (localStorageServiceProvider) {  
    localStorageServiceProvider  
        .setPrefix('scotch-todo');  
});
```

Let us leave it at that and continuing to the second item on our task - the controller.

The main Controller

Our controller will be serving as the middle man between our data (Local Storage) and the view (www/index.html). Let us first create a structure of the controller just below our config block in the www/js/app.js:

```
app.controller('main', function ($scope, $ionicModal, localStorageService)  
{ //store the entities name in a variable var taskData = 'task';  
  
    //initialize the tasks scope with empty array  
    $scope.tasks = [];  
  
    //initialize the task scope with empty object  
    $scope.task = {};  
  
    //configure the ionic modal before use  
    $ionicModal.fromTemplateUrl('new-task-modal.html', {  
        scope: $scope,  
        animation: 'slide-in-up'  
    }).then(function (modal) {  
        $scope.newTaskModal = modal;  
    });  
  
    $scope.getTasks = function () {  
        //fetches task from local storage  
        ...  
    }  
    $scope.createTask = function () {  
        //creates a new task
```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



```

    ...
  }
  $scope.removeTask = function () {
    //removes a task
    ...
  }
  $scope.completeTask = function () {
    //updates a task as completed
    ...
  }
  })

```

Notice how we have injected localStorageService in the controller so as to help us interact with Local Storage from the controller. We also injected \$ionicModal service to help us create tasks from a modal.

We also need to tell the index.html which controller to use and run getTasks() when the app starts by updating the body tag as shown:

```

<body ng-app="scotch-todo" ng-controller="main" ng-init="getTasks()">
  ...
</body>

```

We are making reasonable progress. Let us now implement those empty methods in the controller. We will start with the getTasks() which basically checks for tasks entity in the Local Storage and if it exists, fetch it.

```

$scope.getTasks = function () {
  //fetches task from local storage
  if (localStorageService.get(taskData)) {
    $scope.tasks = localStorageService.get(taskData);
  } else {
    $scope.tasks = [];
  }
}

```

Furthermore we can create a new task by pushing a task object to the tasks array and updating the Local Storage with the tasks.

```

$scope.createTask = function () {
  //creates a new task

```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



```
$scope.tasks.push($scope.task);  
localStorageService.set(taskData, $scope.tasks);  
$scope.task = {};  
//close new task modal  
$scope.newTaskModal.hide();  
}
```

To remove task we can now remove a task object from the task array and update the Local Storage.

```
$scope.removeTask = function (index) {  
    //removes a task  
    $scope.tasks.splice(index, 1);  
    localStorageService.set(taskData, $scope.tasks);  
}
```

Then to complete a task we find task from the array using its index, update it's completed value and update Local Storage.

```
$scope.completeTask = function (index) {  
    //updates a task as completed  
    if (index !== -1) {  
        $scope.tasks[index].completed = true;  
    }  
  
    localStorageService.set(taskData, $scope.tasks);  
}
```

That is it! We can boast of a controller now. Let us not get too excited though because our view is waiting patiently to get a treat as we did to our controller.

The view

For a cleaner display of tasks, let us use ionic's card component for display of tasks. So right inside the <ion-content> tag (which is actually a directive) add the following block:

```
<div class="list card" ng-repeat="task in tasks track by $index">  
    <div class="item item-divider">  
        <span ng-bind="task.title"></span>  
    </div>  
    <div class="item item-body">
```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



```

        <strong>Title: <span ng-bind="task.title"></span></strong>
        <p>
            <span ng-bind="task.content"></span>
        </p>
    </div>

    <div class="item tabs tabs-secondary tabs-icon-left">
        <span class="item item-checkbox">
            <label class="checkbox">
                <input type="checkbox" ng-model="task.completed"
ng-click="completeTask($index)">
            </label>
        </span>
        <a class="tab-item assertive"
ng-click="removeTask($index)">
            <i class="icon ion-android-close"></i>
        </a>
    </div>

</div>

```

With that we have a list of tasks presented with cards in our view. Now to create a new task, we will use the modal which we already configured in our controller. We are tracking the array with a unique \$index as it has none specified explicitly in the data source. Just before the body closing tag:

```

<script id="new-task-modal.html" type="text/ng-template">
    <ion-modal-view>
        <ion-header-bar class="bar-dark">
            <h1 class="title">Create a new Task</h1>
            <button class="button button-icon"
ng-click="closeTaskModal()">
                <i class="icon ion-android-close"></i>
            </button>
        </ion-header-bar>
        <ion-content>
            <form ng-submit="createTask()">
                <div class="list list-inset">
                    <label class="item item-input">
                        <input ng-model="task.title" type="text"
placeholder="Task title">
                    </label>
                </div>
            </form>
        </ion-content>
    </ion-modal-view>
</script>

```



Erasmus+



FPIC
FORMACIÓN PROFESIONAL
IES CAMPANILLAS



EU-Geschäftsstelle
Wirtschaft + Berufsbildung



```

        <label class="item item-input">
            <textarea ng-model="task.content" rows="5"
placeholder="Task content"></textarea>
        </label>
        <ul class="list">
            <li class="item item-toggle">
                Completed?
                <label class="toggle toggle-balanced">
                    <input type="checkbox"
ng-model="task.completed">
                    <div class="track">
                        <div class="handle"></div>
                    </div>
                </label>
            </li>
        </ul>
        <button type="submit" class="button button-block
button-positive">Create Task</button>
    </div>
</form>
</ion-content>
</ion-modal-view>
</script>

```

Finally on the view, add a button to the ion-header-bar directive which will be used to open the modal.

```

<ion-header-bar class="bar-dark">
    <h1 class="title">Scotch-Todo</h1>
    <!-- New Task button-->
    <button class="button button-icon" ng-click="openTaskModal()">
        <i class="icon ion-compose"></i>
    </button>
</ion-header-bar>

```