1)course intro

This course is about ionic2, a framework which builds up on angular 2 framework which of course

allows you to build awesome web applications, to then take this angle 2 knowledge to build awesome

native applications applications running natively on iOS or Android or Windows Phone devices.

3)what is ionic 2/3

Well the goal with ionic is to create apps for Android or Windows or ios.And I'm speaking of native apps so apps running on those devices installed on those devices.

We might not know languages neede to build these but you know html, js and css. now would be nice if you could kind of take that knowledge ,take those web apps and transform them into native apps for iOS Android

and Windows.

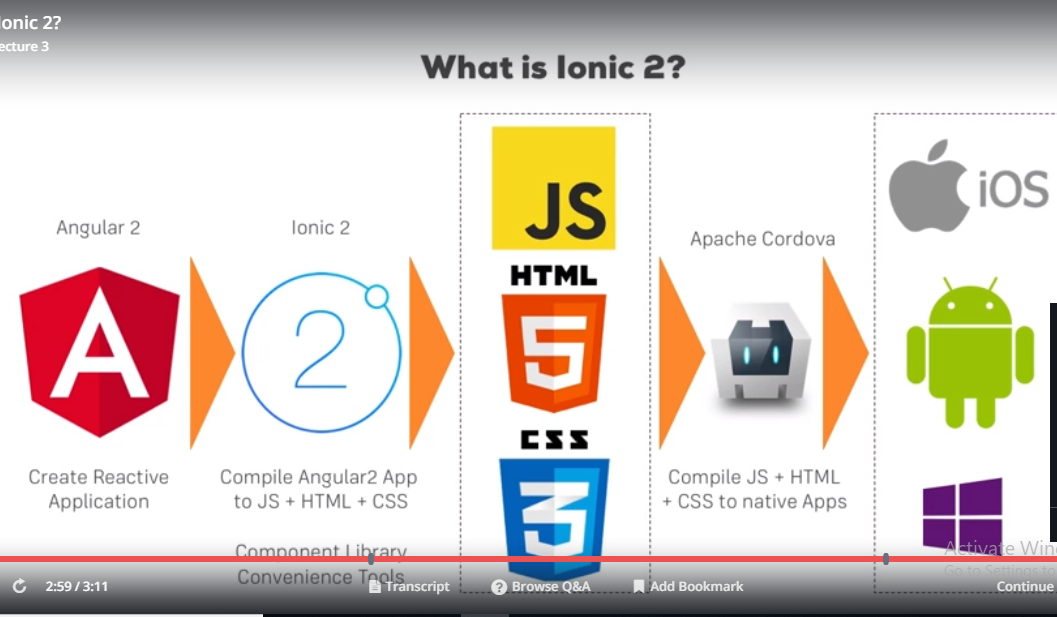
Now it turns out there is a helpful tool a framework which does just that and it is not ionic 2 its

Apache Cordoba framework by Apache.



And what does corodova does is it compiles js, html, css , so web apps into native apps. Now of course those apps have to follow certain structures and need to be setup in certain way. But if that is the case the apache corodova is able to easily create those native apps for you. So where ionic 2 come into picture. No chances are you dnt want to create your web app with vanilla js, html and css but you do know angular(specifically angular 4 here). Which is awesome js frameowrk. Now you have already have way of compiling angular 2 code you could say, so typescript to js ,html and css code creating such web applications.

Still ionic 2 is introduced as extra step between angular 2 and the web app cordova ios android windows part here.



What ionic does is it gives you some tooling which allows you to create projects following that structure that apache cordova needs in the need. It will also compile your angular ts code into native or into a vanilla js,,html or css code, following the structure that cordova needs and it also gives you a rich component library with beatifully styled components , you will need for your mobile apps like nice modals , alerts, sliders and tabs so on and lot of convenient tools , helper methods and much more. This is what ionic 2 is in the end, a package , a tool set , a command line interface in the end that you will install, which allows you to easily create pre-configured projects , a work flow which will compile your angular code and which will then also trigger this apache codova compulation so that you never have to work with apache cordova directly and this rich library of conveneince tools and these components you can use.

6)creating your first app

In the next lecture, we're going to create our first Ionic app.

Here's an important information: The command we need to run did change slightly.

When you see me type ionic start firstapp --v2 , you should NOT use --v2 . Just run ionic start firstapp --type=ionic-angular

You'll then be asked a couple of questions.

1) Use Cordova? => Pick "YES" (y)

2) Use Ionic Pro? => No, not required => "NO" (n)

7)Creating your First Ionic 2 project and application

install node. This is because we need npm to manage dependencies. We will need nodejs itself because the ionic framework or ionic toolset also ships with little development server which allows us to preview our app in browser, for this it needs nodejs to spin up a development server.

Then we need to install ionic and cordova. This step needs to be excuted once and you will need to do that to get the ionic cli , the toolset which then allows you to create ionic projects. Run this command-

**npm install ionic cordova -g**

here we are installing 2 packages ionic and cordova. Now lets create ionic project. Run this-

ionic start firstapp

this will give ionic first app, we want to use ionic 2 so run this-

ionic start firstapp –v2

but in last lecture it was mentioned . So we run this command-

**ionic start firstapp --type=ionic-angular**

now you will be asked whether about some options. You should use tabs and Cordova.

Or you can use this command-

**ionic start firstapp blank**

Then you will options to install ionic free sdk and connect your app, select no.

Now navigate into project and run-

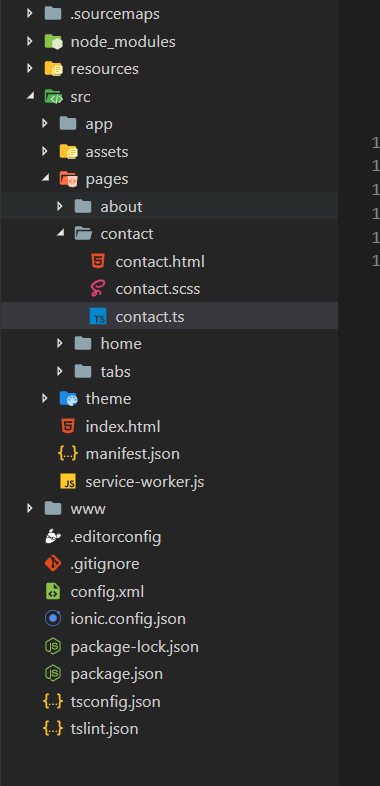
**ionic serve**

this command will compile your application to js,css and html not to a native app yet. I will let you know later you how to do this. Then it will spin up little development serve, running at port shown in cmd.

8)Changing your first app

We selected tab template while creating project, so our app has 3 tabs at bottom which loads different pages. Now our process is running in cmd, when ever we will change something in ide, code will recompiled and browser will be reloaded.

Lets open our project in ide, there we have lot of files and folders. I will come back what these individual files do. For now we will concentrate on core folder we work with, the source folder. This is where we create our angular2 application and where we therefore work 99 percent of the time. In this folder you might not instantly recognize angular application, but that will change if you open app folder. Here you will see app.component.ts, app.module.ts and main.ts(file which bootstraps the app.module) and in then end this is normal behavior you know with root component being setup and so on. Now the content you see in these files like in app.components.ts and especially in app.module.ts, might differ from your default angular 2 app. And I will come back to especially this file, app.module and explain how this works and how ionic 2 app gets started, in next section. Folder structure-



Here I want to do something different, I want to change the app. In app.component.html we cannot see the content we see on browser. So it is not the place where it gets changed. Ionic works with pages and pages are the pages you see when you click on each tab. Here we have 3 pages that we can see when we click on each tab. So got to pages folder. There we see 4 folders. 3 folder corrosponds to 3 pages. We also got separate page for tabs. So you can think like tabs at bottom has their own page, which ha sthen huge empty space which gets populated with fitting page for tab we clicked on. So place here we can change something is pages folder.there you will see that apage in the end is just a angular 2 component. Each page has .ts file which has @component decorator. And there fore it also has a template.

Contact.ts-

import { Component } from '@angular/core';

import { NavController } from 'ionic-angular';

@Component({

selector: 'page-contact',

templateUrl: 'contact.html'

})

export class ContactPage {

constructor(public navCtrl: NavController) {

}

}

Each page has .ts file which has @component decorator. And therefore it also has a template. So template is where we can change something. Lets add a h1 tag to about page then.

About.html-

<ion-header>

<ion-navbar>

<ion-title>

About

</ion-title>

</ion-navbar>

</ion-header>

<ion-content padding>

<h1>This is about page</h1>

</ion-content>

9)Running your app on real device

In this section and also in the next few sections, we'll run the app in the browser (ionic serve). This simply is the best way of developing the app - it's easy to implement changes (and see them) as well as to debug the app.

Later in the course (in **section 5**) you will of course also **learn how to ship your app** to an **emulator** or a **real device**. If you can't wait, you can of course skip ahead but I strongly recommend sticking to the browser for now.

12)Section source codes and links

**The source code of this section can be found attached to this lecture!**

If you got problems running it, you might be using a newer version of the Ionic CLI. Try npm run ionic:serve  after npm install in the project folder in such cases!

Got **issues with the installation** of Ionic 2 or the creation of your project? Have a look at this article from the official docs: <https://ionicframework.com/docs/setup/installation/>

Do you want to learn more about the **CLI** and available **commands**? The following article helps: <https://ionicframework.com/docs/cli/>

EaEa