MIRACLE
SOFTWARE SYSTEMS

Your First Hybrid Mobile App with Ionic Framework

DS'17 Hands On Mobile Lab

Venkatesh Voona
Lead Researcher(MIL) - Mobile/NextGen
Miracle Software Systems, Inc.

December 9th, 2017

©2017 Miracle Software Systems, Inc.

Your First Hybrid Mobile App with Ionic Framework

Goal

In this lab we will guide you how to create a hybrid mobile application with Ionic 2 Framework. Here we are going to create a sample login screen and then how to navigate from one screen to another screen

Pre-Requisites

The following installations will need to be completed for this lab to be run successfully,

- Node JS and NPM installed
- Cordova
- Ionic Framework
- Text editor (Visual Studio Code is preferable)

Technology Involved

- Ionic 2 Framework
- Cordova
- HTML / CSS / JS
- Typescript
- Angular 2/4

Lab Steps

Let's get started with the lab!

©2017 Miracle Software Systems, Inc.

#1 | Create a new Ionic project

To create/start a new project in Ionic 2 framework, we need to give a below command

ionic start <application name> <starter template name>

We will be having following starter templates in ionic 2 framework.

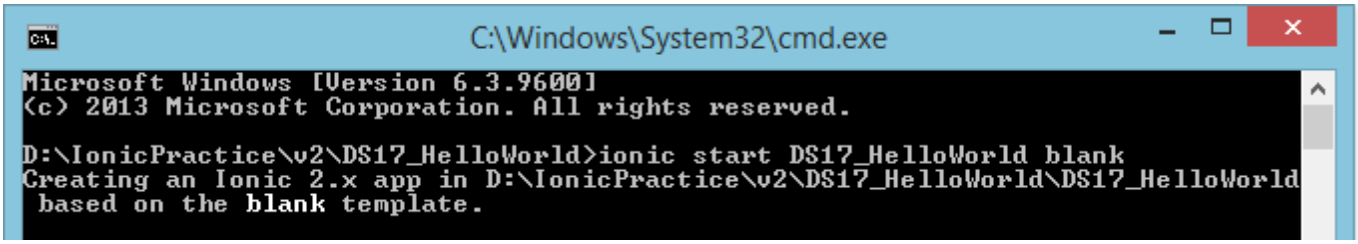
tabs	A starting project with a simple tabbed interface
blank	A blank starter project
sidemenu	A starting project with a side menu with navigation in the content area
super	A starting project complete with pre-built pages, providers and best practices for Ionic development.
conference	A project that demonstrates a realworld application
tutorial	A tutorial based project that goes along with the Ionic documentation
aws	AWS Mobile Hub Starter

In this lab we are going to use blank template and start implementing the actual application.

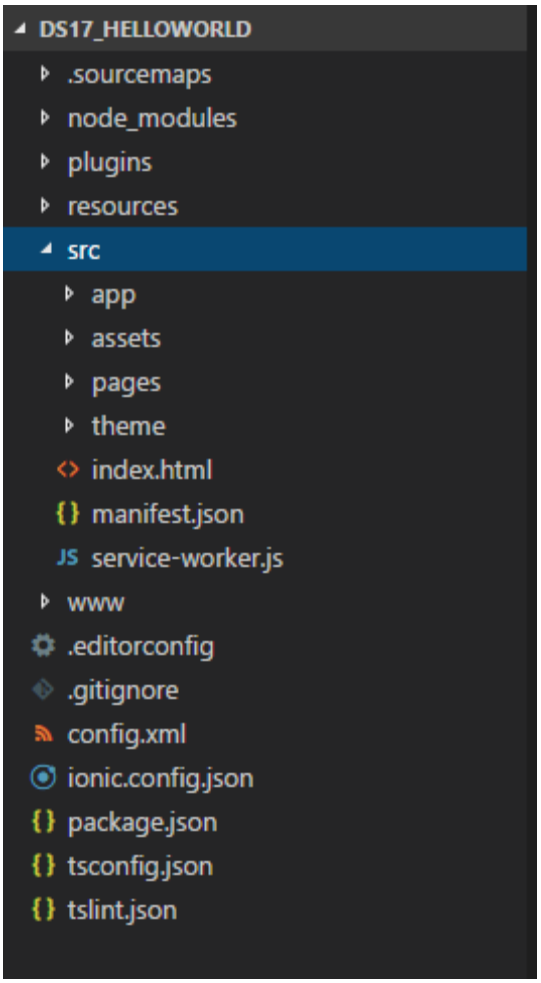
ionic start DS17_HelloWorld blank

Give the above command in the command prompt as shown in the below figure.

©2017 Miracle Software Systems, Inc.



Once you give the above command, the ionic application will be created with the following folder structure as shown in the figure.



#2 | Running the application

Before going to give run command, we need to route to the current application folder and then give the following commands.

©2017 Miracle Software Systems, Inc.

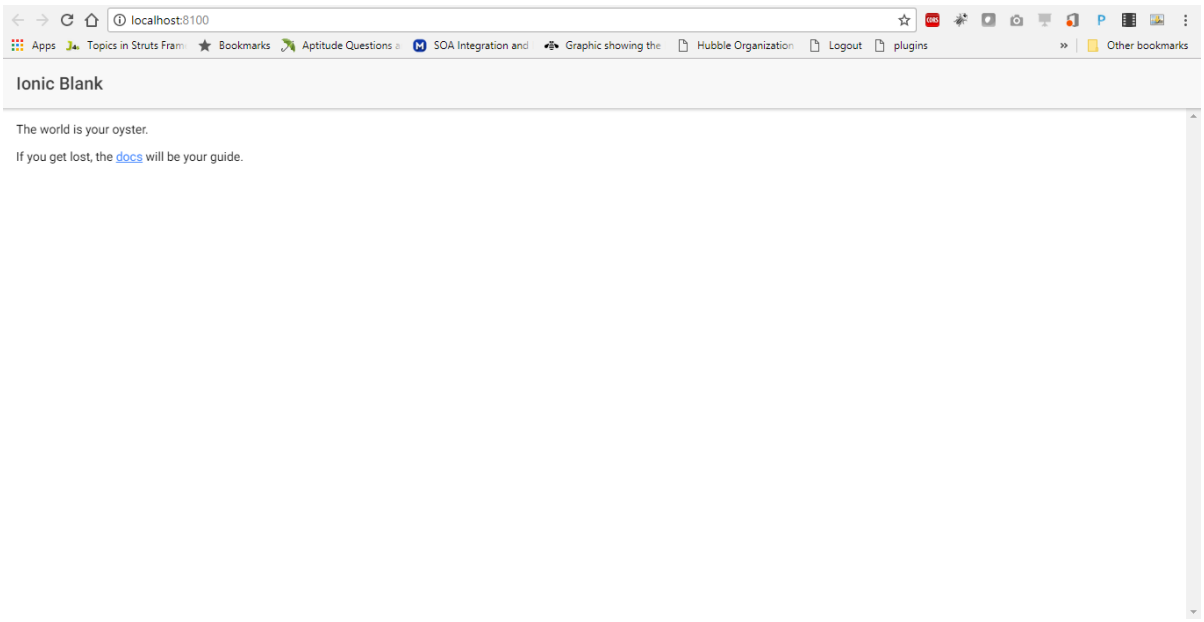
Run on Browser,

ionic serve

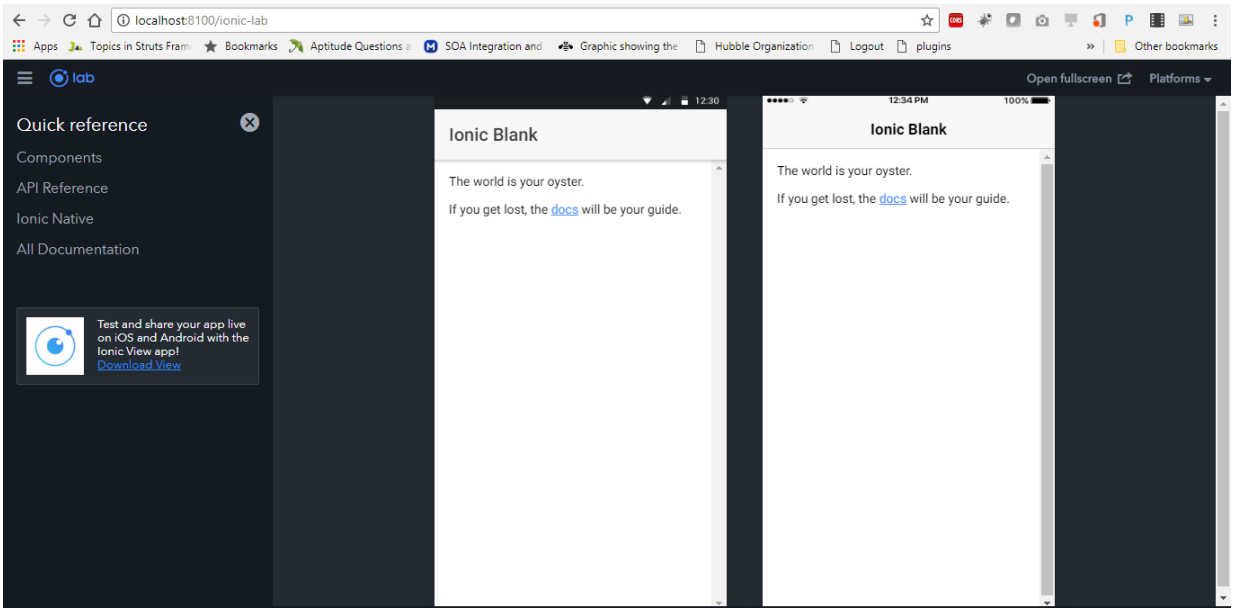
Run on Mobile simulator,

ionic serve --lab

If you run the application on browser, you can see the output as shown in the below figure



If you run the application on mobile browser simulator, you can see the output as shown in the below figure. Here if you observe you can see the output in two different platforms one is Android and another one is IOS.



#3 | Customizing Application

As you have observed above folder structure, you will be having src folder where we need to customize our application based on our requirements.

Here, in our case we are going to implement an sample login form. Go to src/pages/ and open home.html. Copy and paste the below code snippets to design sample login form.

Code Snippet : home.html

```
<ion-header>
  <ion-navbar color="headerColor">
    <ion-title>
      DS'17 Hello Word App
    </ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <!-- DS'17 Logo     -->
```

```
  <div text-center>
    <img src="assets/imgs/ds17logo.png" width="150"
height="150">
  </div>

  <!-- Login Form  -->
  <ion-item>
    <ion-label floating>
      <ion-icon name="person"></ion-icon> Username</ion-label>
    <ion-input type="text" name="username"></ion-input>
  </ion-item>

  <ion-item class="login-signup">
    <ion-label floating>
      <ion-icon name="lock"></ion-icon> Password
    </ion-label>
    <ion-input type="password" name="password"></ion-input>
  </ion-item>

  <div text-center padding>
    <button type="submit" ion-button class="lButton"
(click)="nextPage()">Login</button>
  </div>

</ion-content>
```

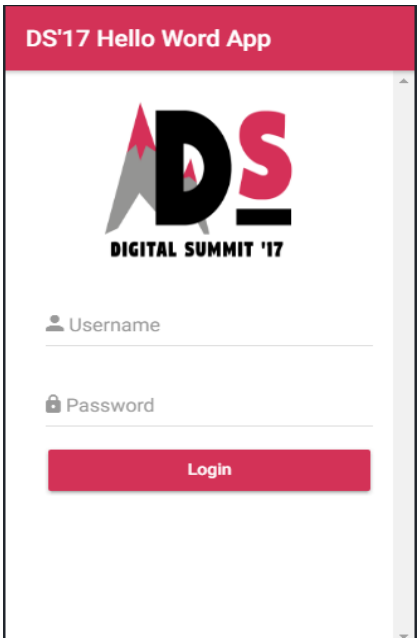
Code Snippet : home.scss

```
.lButton{
    width:105%;
    text-transform: none;
    background-color: #d33257
}
.fPwd{
    color: #00aae7;
    margin-top: -7px;
}
```

Code Snippet : home.ts

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { SecondPage } from '../second-page/second-page';

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {
  constructor(public navCtrl: NavController) {

  }
  nextPage() {
        this.navCtrl.push(SecondPage);
  }
}
```

If you add above code snippets into your app, you can see the following screen.



©2017 Miracle Software Systems, Inc.

#4 | Navigate from one screen to another screen

To navigate from one screen to another screen, NavController module needs to be imported. Below is the statement for importing module from ionic- angular.

```
import { NavController } from 'ionic-angular';
```

Before going to navigate we need to create a new page by using the following command,

```
ionic generate [<type>] [<name>]
```

```
ionic generate page secondPage
```

By giving the above command, the following folder will create automatically.

Right after new page creation, you need to configure newly created page in the app.module.ts. Add the **SecondPage** in the declarations and entryComponents arrays in app.module.ts file as shown in the below code.

Code Snippet - app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
import { SplashScreen } from '@ionic-native/splash-screen';
import { StatusBar } from '@ionic-native/status-bar';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { SecondPage } from '../pages/second-page/second-page';

@NgModule({
 declarations: [
  MyApp,
  HomePage,
  SecondPage
 ],
 imports: [
```

```
    BrowserModule,
    IonicModule.forRoot(MyApp)
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage,
    SecondPage
  ],
  providers: [
    StatusBar,
    SplashScreen,
    {provide: ErrorHandler, useClass: IonicErrorHandler}
  ]
})
export class AppModule {}
```

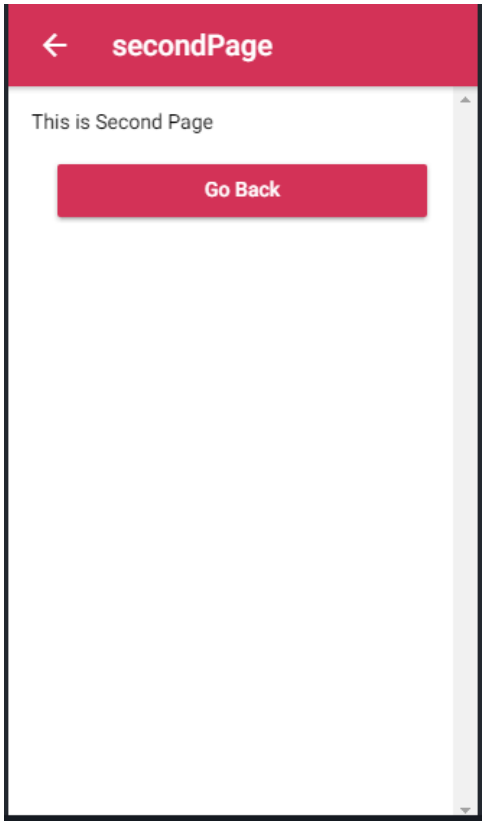
Now we are going to design a sample screen for second page, below is the code snippet for second screen

Code Snippet : secondpage.html

```
<ion-header>

<ion-navbar color="headerColor">
<ion-title>secondPage</ion-title>
</ion-navbar>
</ion-header>
<ion-content padding>

This is Second Page
<div text-center padding>
        <button    type="submit"  ion-button  class="lButton"  (click)  =
"goBack()">Go Back</button>
 </div>

</ion-content>
```

Copy and paste the above code in secondpage.html, you will get the following screen,

©2017 Miracle Software Systems, Inc.

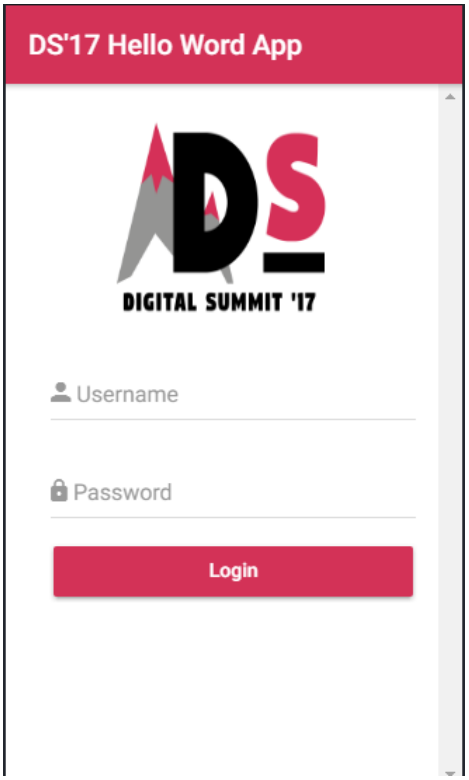


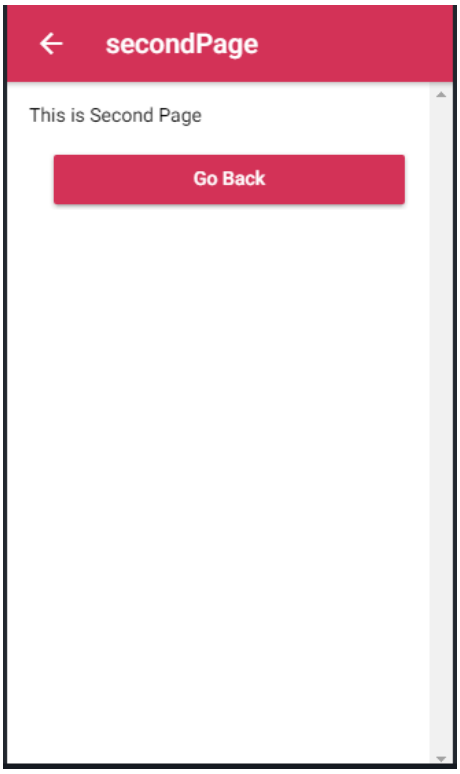
Finally, we need to give the click action to submit button in home screen and add the following code in home.ts file .

```
nextPage() {
      this.navCtrl.push(SecondPage);
}
```

By adding the above code, we can achieve the navigation from one screen to another screen. If you click on login button in the screen one, then your app will navigate to second screen as shown in the below screens.

©2017 Miracle Software Systems, Inc.





©2017 Miracle Software Systems, Inc.