MIRACLE
SOFTWARE SYSTEMS

Connecting your Ionic App to a Public API Service

Open Lab | Digital Summit '18

Miracle Innovation Labs

Miracle Software Systems, Inc.

MIRACLE
SOFTWARE SYSTEMS

Connecting Your Ionic App to a Public API Service

Goal

This document contains a step-by-step process for adding a public weather API to an Ionic 3 Mobile App. Here you will be creating a mobile app to get the weather of a particular city by giving the location name and it's State.

This guide was prepared by **Miracle's Innovation Labs.**

Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the lab (Both PC and MAC are compatible). The following pre-requisites will help you to make the Hands-on Lab experience easier.

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

Technology Involved

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

Lab Steps

Let's get started with the lab!

Step #1 | Create a New Ionic Project

To create/start a new project in Ionic 3 framework, we need to give a below command,
ionic start <application name> <starter template name>

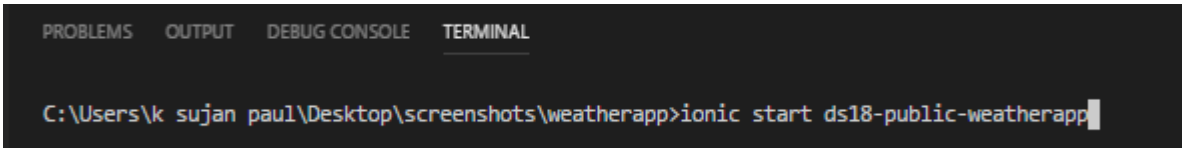
We will be having following starter templates in ionic 3 Framework,

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

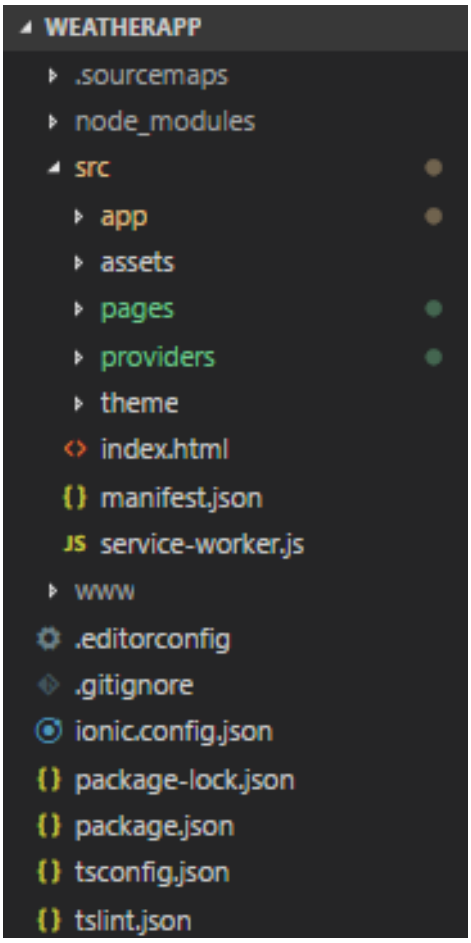
In this lab we are going to use blank template and start implementing the actual application.

E.g.: ionic start DS18_public-weatherapp blank

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



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



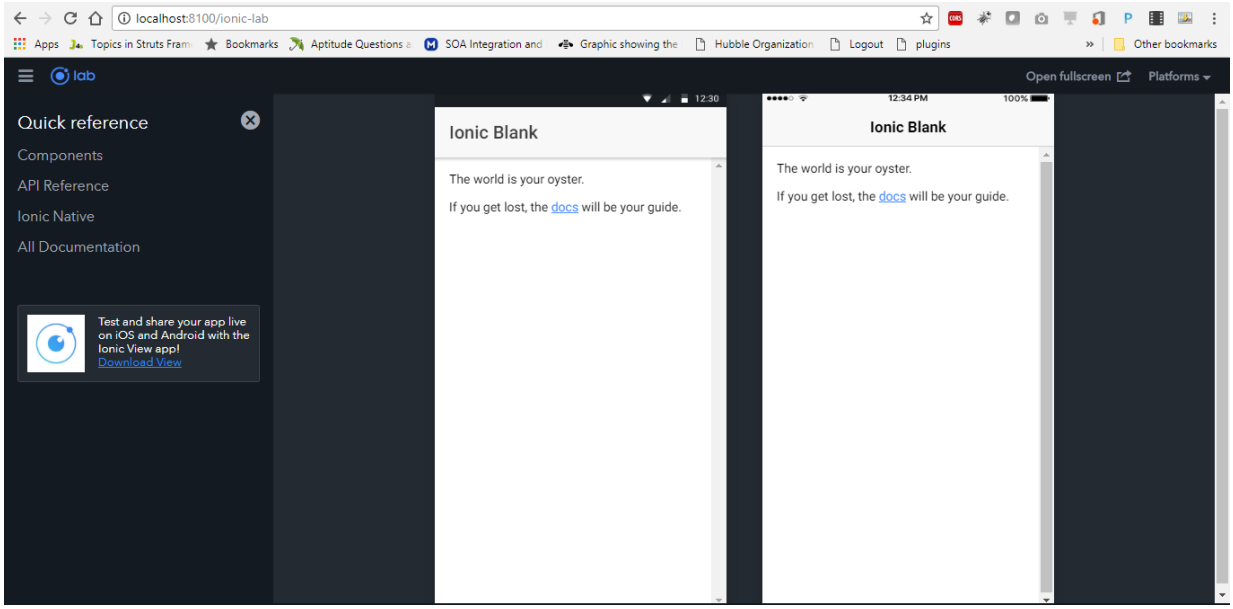
Step #2 | Running the Application

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

Run on Browser - ionic serve
Run on Mobile simulator - ionic serve -l

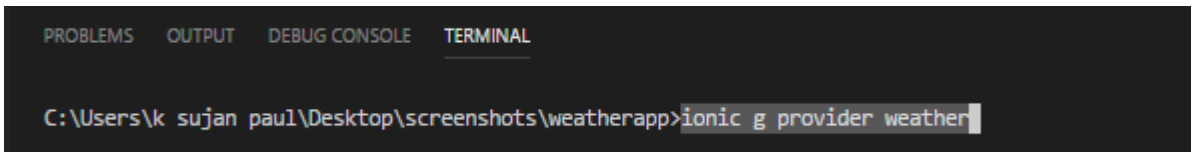
If you run the application in the browser, you will be able to see the output as shown in the below figure,

Run the application on a simulator, and the output will be as shown below. Here if you observe you can notice that the output is in two different platforms one is in Android and another one is in iOS.



Step #3 | Create a Weather Service

In order to create a service provider in your Ionic application, provide the command **ionic g provider <Service Name>** in the command prompt as shown below.



Right after the service provider creation, you need to add this service in the providers array in **app.module.ts**.

Provider Array -

```
providers: [
  StatusBar,
  SplashScreen,
  Weatherprovider,
  {provide: ErrorHandler, useClass: IonicErrorHandler}
]
```

Step #4 | Service Implementation to give a Specified Location

Replace the code snippet for implementing the weather report service (provider) in the Ionic application.

Code Snippet : WeatherProvider.ts [Please refer to the cheat sheet]

Step #5 | Create the Location Input Page and Weather Report Page

To create a new page in your Ionic application, provide the following command, **ionic g page <page name>**

After creating the new page, you have to update the **app.module.ts** with this new page details like importing page, adding it in the **declarations** and **entryComponents** arrays in **app.module.ts** file.

Code Snippet : app.module.ts [Please refer to the cheat sheet]

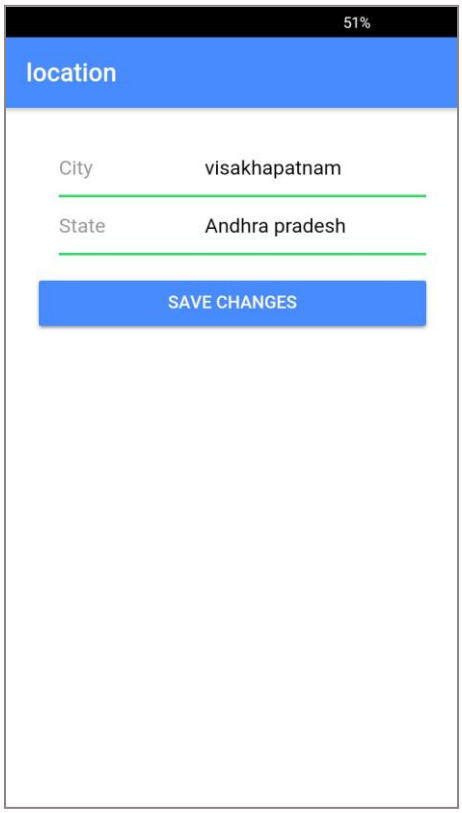
Step #6 | Designing the Screens

In this lab you need to design 2 screens, one is location screen and the second one is weather report screen. Below are the code snippets for each screen.

Code Snippet : Location.html [Please refer to the cheat sheet]

The outputs of the above screens are shown in the figure. No need to run the execution command again and again, if you made any changes then the application will automatically builds.

Screenshots



Step #7 | Accessing Service from Component

In this lab, you are going to use home screen as a sign-in and register screen as a sign-up. Below are the code snippets for accessing service from the component.

Code Snippet : location.ts [Please refer to the cheat sheet]

Code Snippet : weather.ts [Please refer to the cheat sheet]

Step #8 | Getting the Weather Report Analysis

In order to know the weather details for a particular location, you need to provide the City and State name. You will be able to see the results as shown below,

Hurrah!! With this lab you were able to create a mobile app to get the weather of particular city by giving the location name and its state.

For any questions regarding the lab please feel free to reach out to innovation@miraclesoft.com. **We hope you enjoyed creating the mobile app with us.**