Weather Report using Angular and Bootstrap

Introduction:

We developed an application which gives weather report, containing current and hourly forecast of US states and Cities.

Features:

1. Search by State and City
2. View Details like temperature, weather condition, wind speed, pressure, humidity and hourly forecast.
3. By default, it gives us Kansas City weather report.
4. Display error message when user enter wrong data.

Objectives

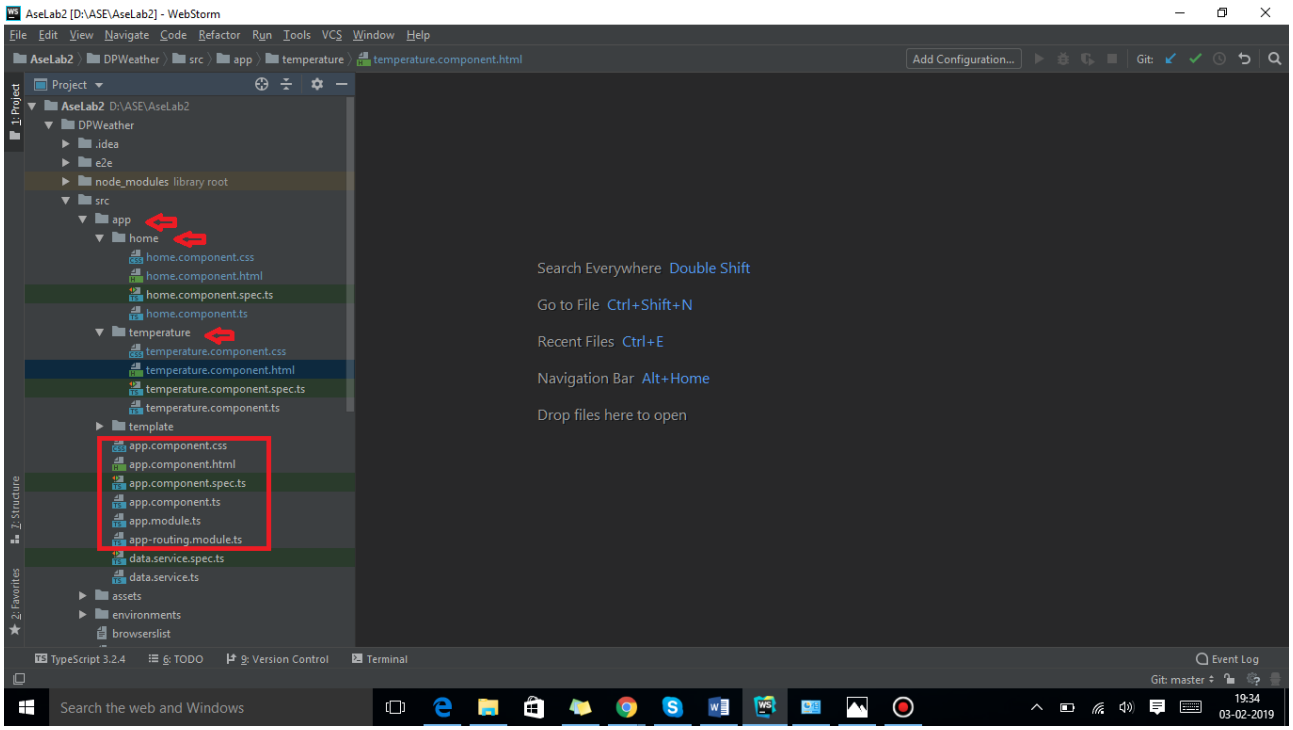
Our objective is to build a weather report using Angular, Bootstrap, WebStorm (IDE)

Prerequisites to build our project:

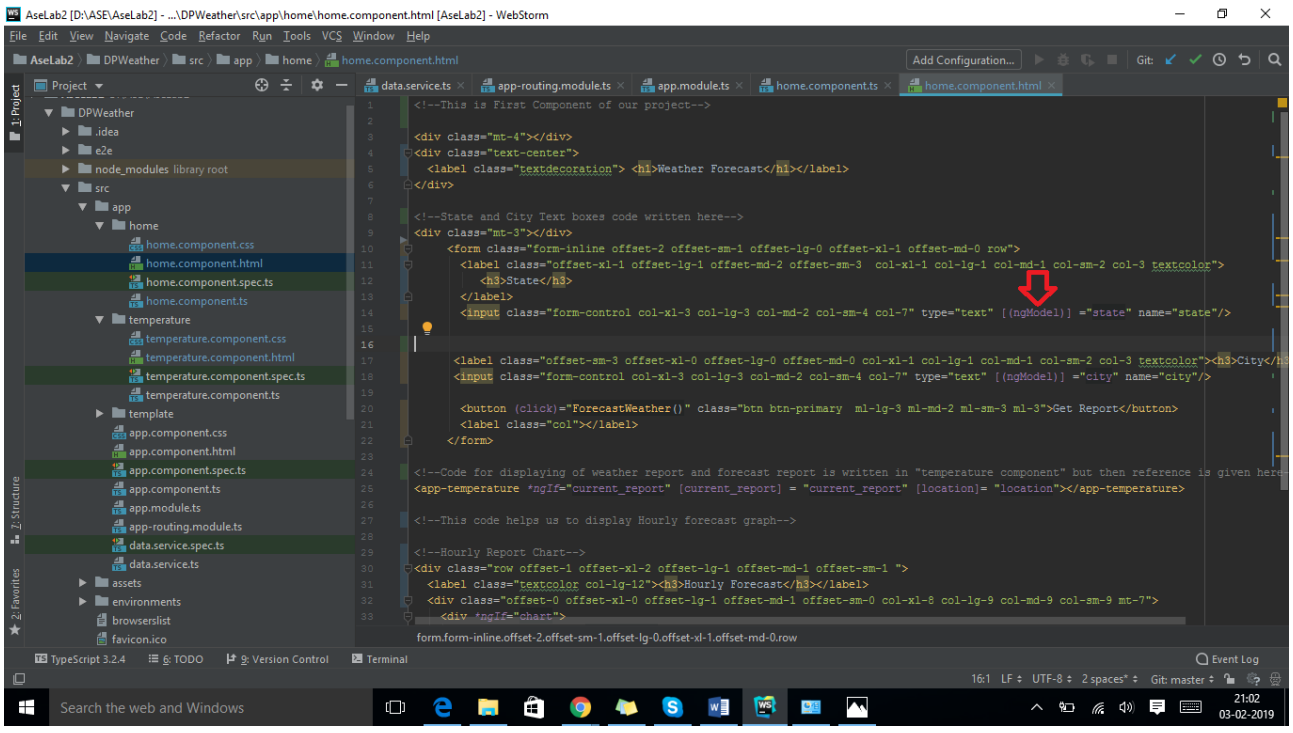
1. Any IDE: We are using WebStorm which assists us to use most popular frameworks.
2. **Node.js**: Should be included into development environment. Required versions: 8x or 10x
3. **npm**: It's a registry used to manage dependencies for node.js
4. Weather forecast APIs in order to fetch service and display weather data
5. Installed Angular CLI globally using npm commands.
6. Installed Bootstrap.
7. Install angular chartjs to the project

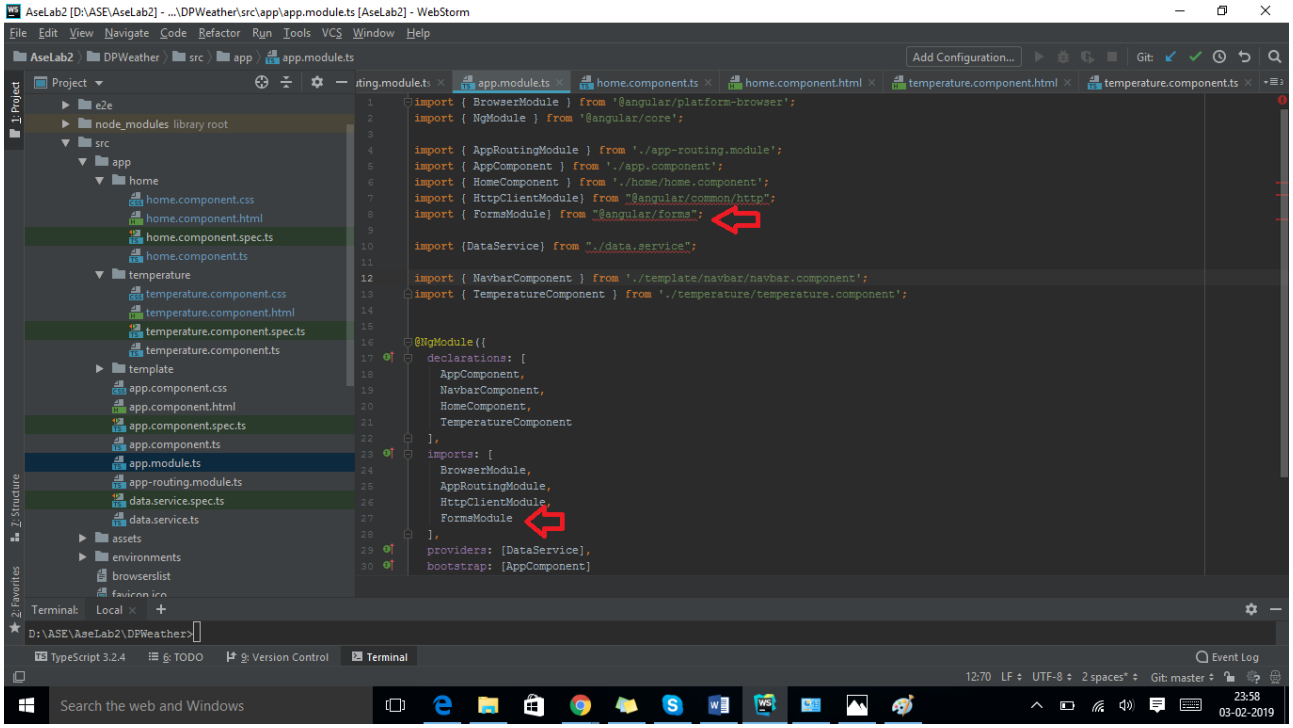
Implementation:

1. Create an angular project using following command in angularcli / webstorm terminal.
   ng new <projectname>
   Once we create a project, by default component called "App" will be created.
   This component acts as a main component of the project
2. Next, create few other components for the project using following command
   ng generate component componentname. The "app.module.ts" has reference for these components.
   a. Home Component main work
      i. A form that takes two inputs State and City
      ii. Displays Hourly Report
   b. Temperature Component:
      i. Displays Weather Report and Forecasts Report



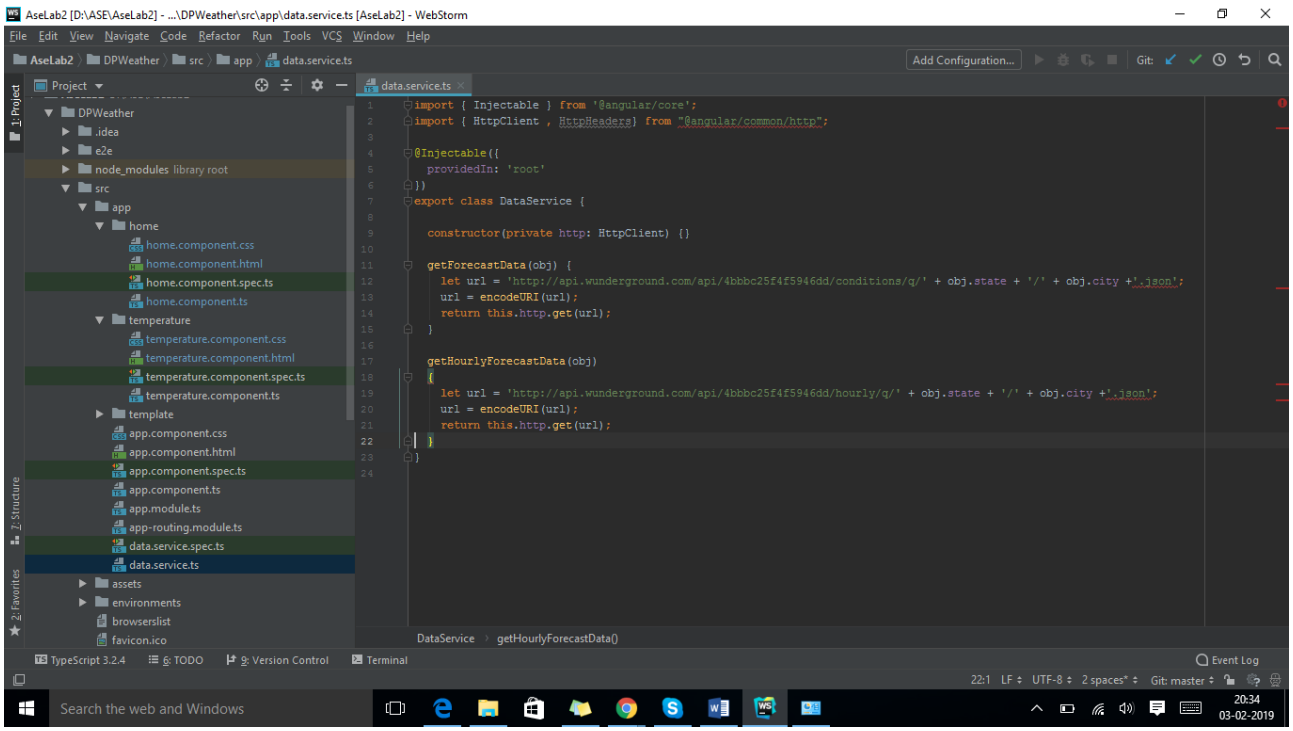
3. After providing state and city, we have to fetch data from services, to achieve that, 2-way data binding needs to be done.
    a. The two data binding in the angular is achieved through [(ngModel)]
    b. Make sure “FormModule” is imported and declared in the “app.module.ts”



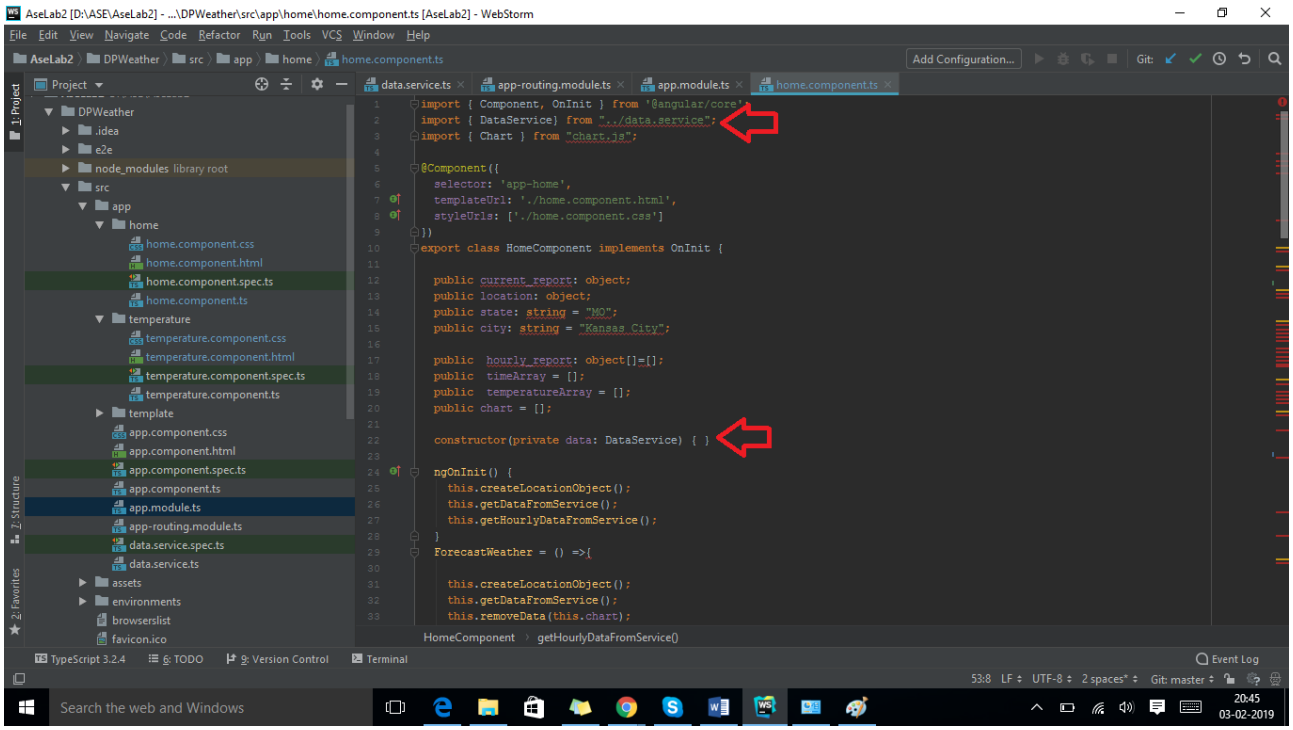


c. As I said before, an angular service is required to fetch the data from service. So, for that we need to create a service and that can be created using following command i.e ng g s “service name”. Here you an see our servicd

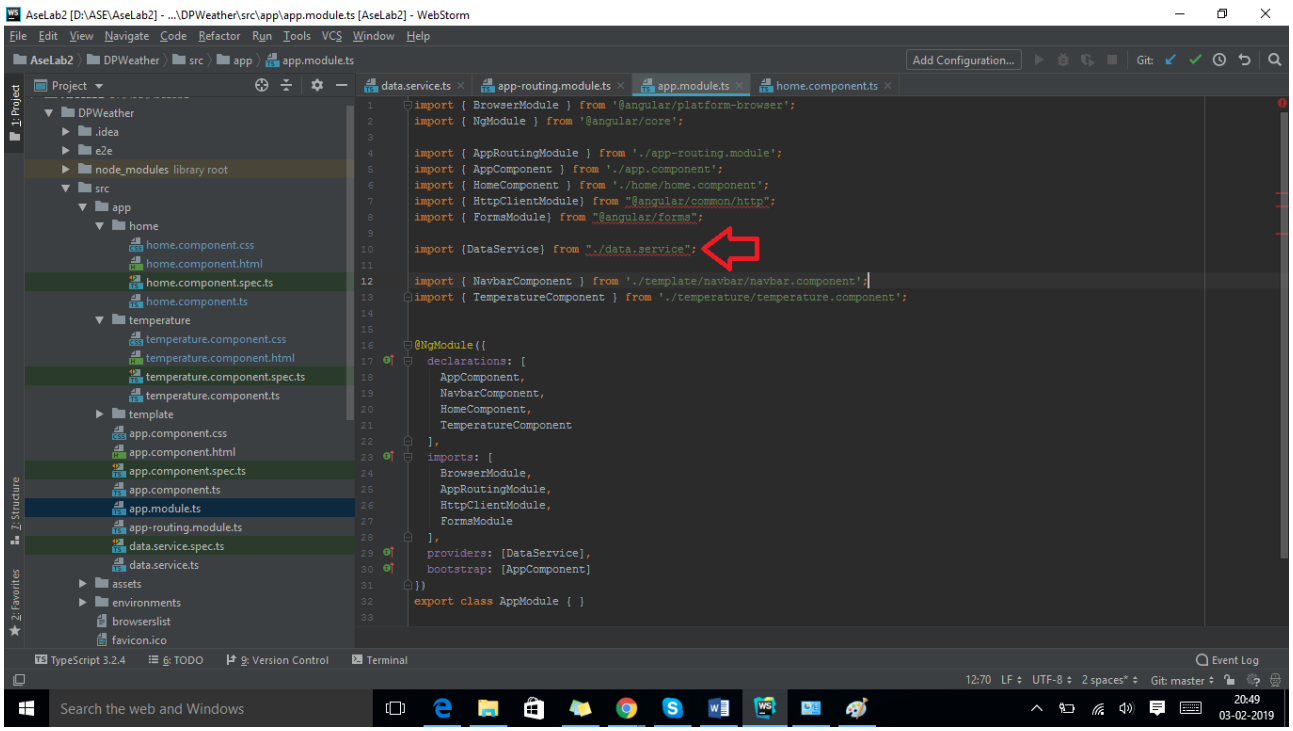
We have created a method here to receive weather services using weather apis



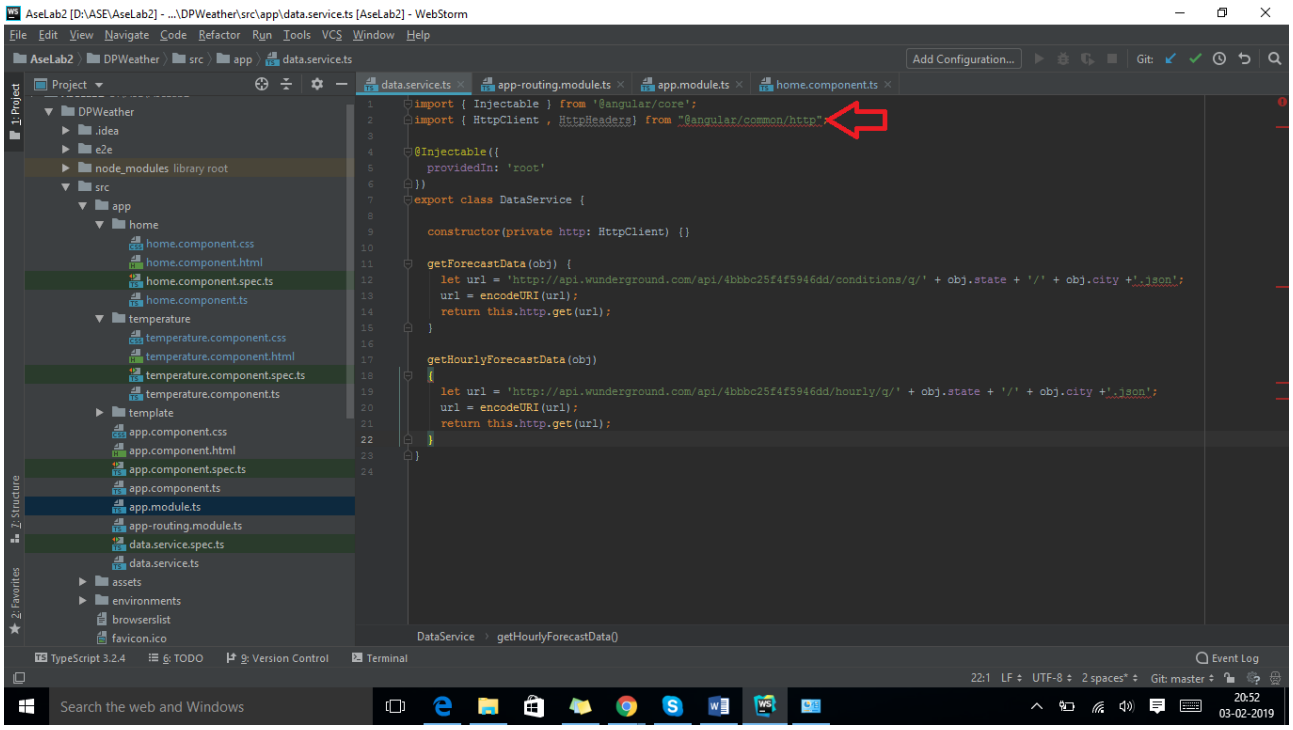
d. In order to access this service method in any other components, import the service in the component and create an instance for the service in the constructor.

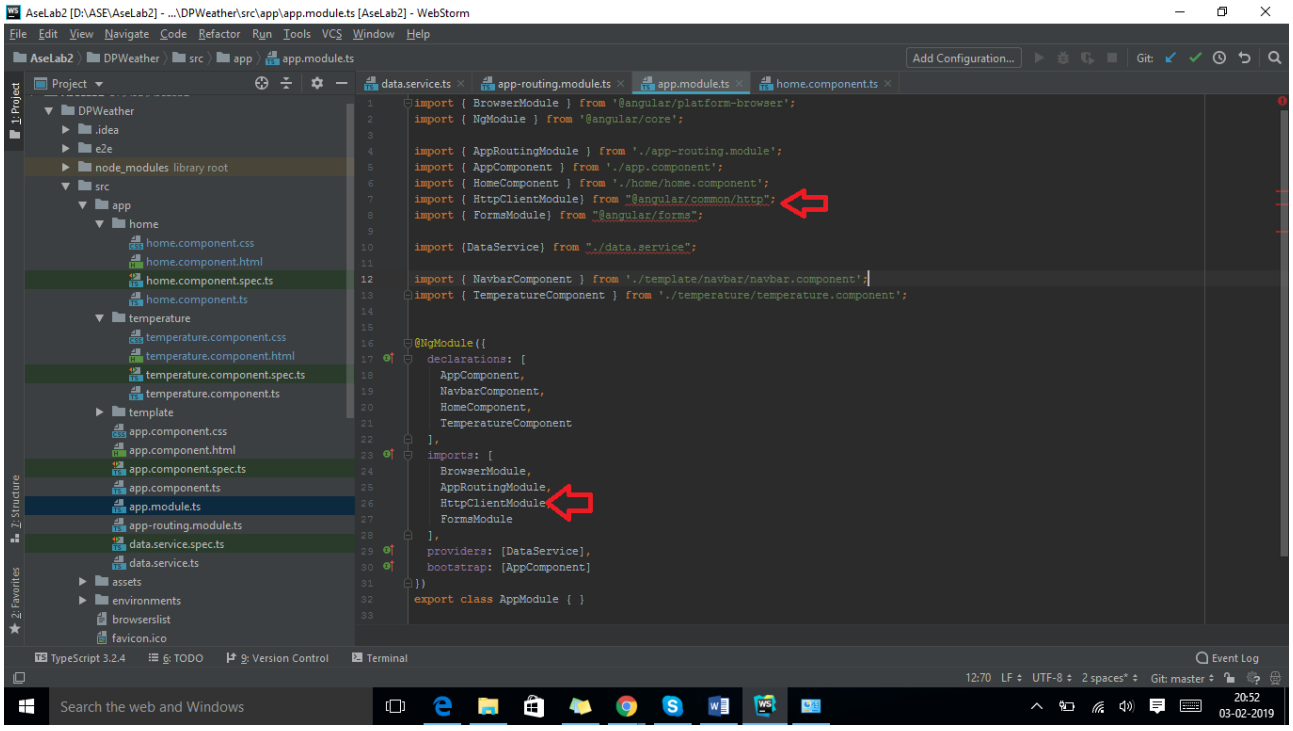


i. Also import the “DataService” into app.Module.ts and add the service to providers array.

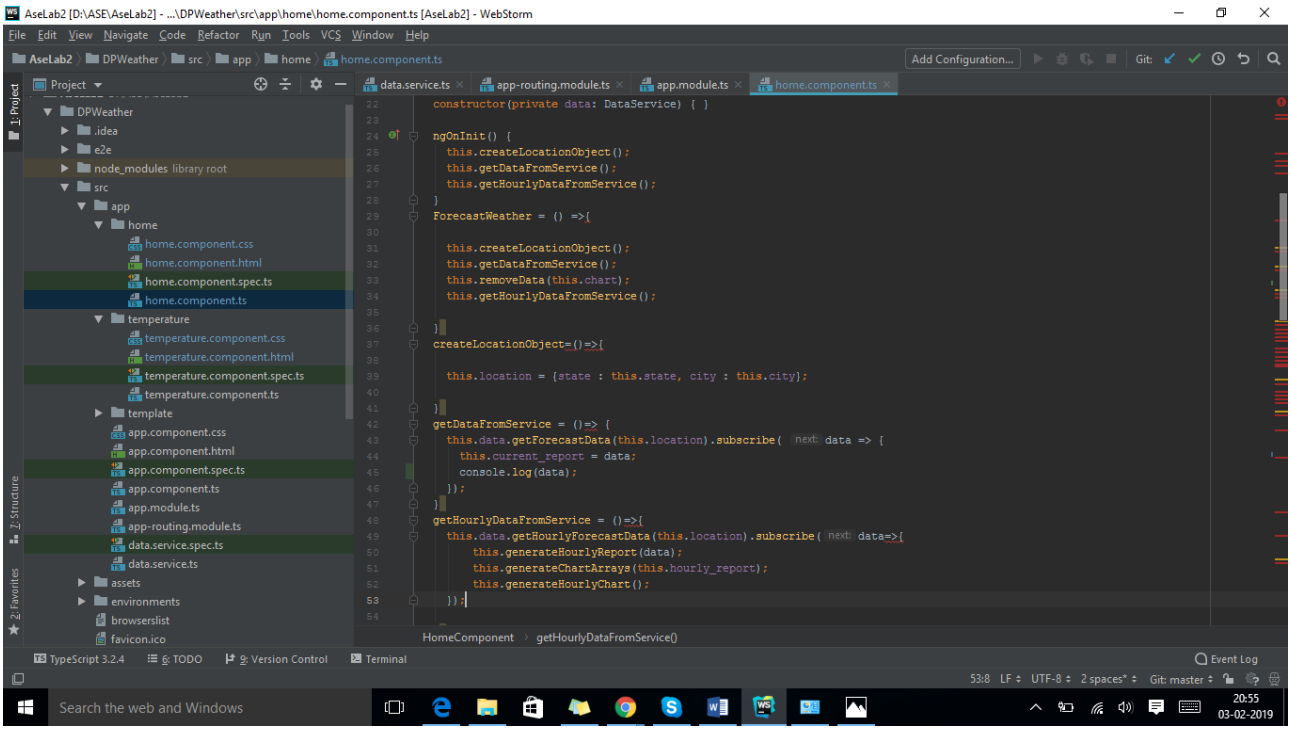


ii. Moreover as we are getting service through http we should also import “HttpClient” package into “data.service.ts” and “HttpClientModule” under app.Module.ts and include HttpClientModule in the imports.



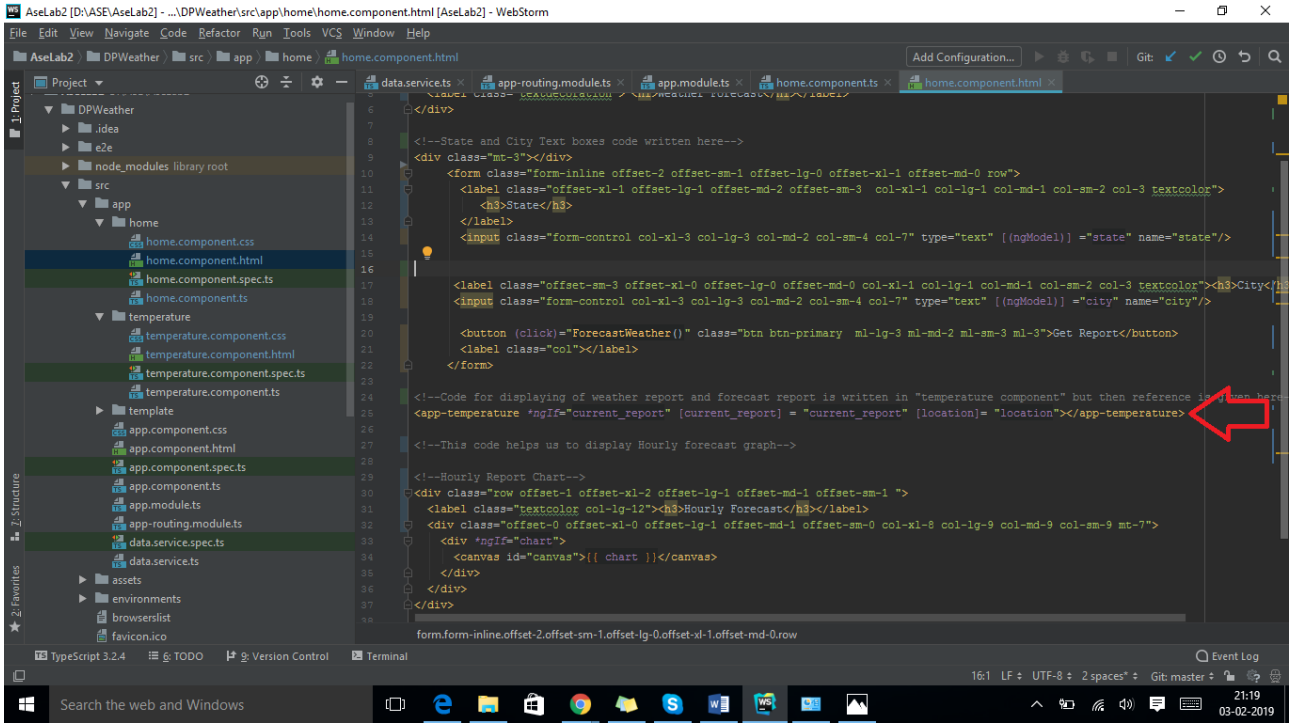


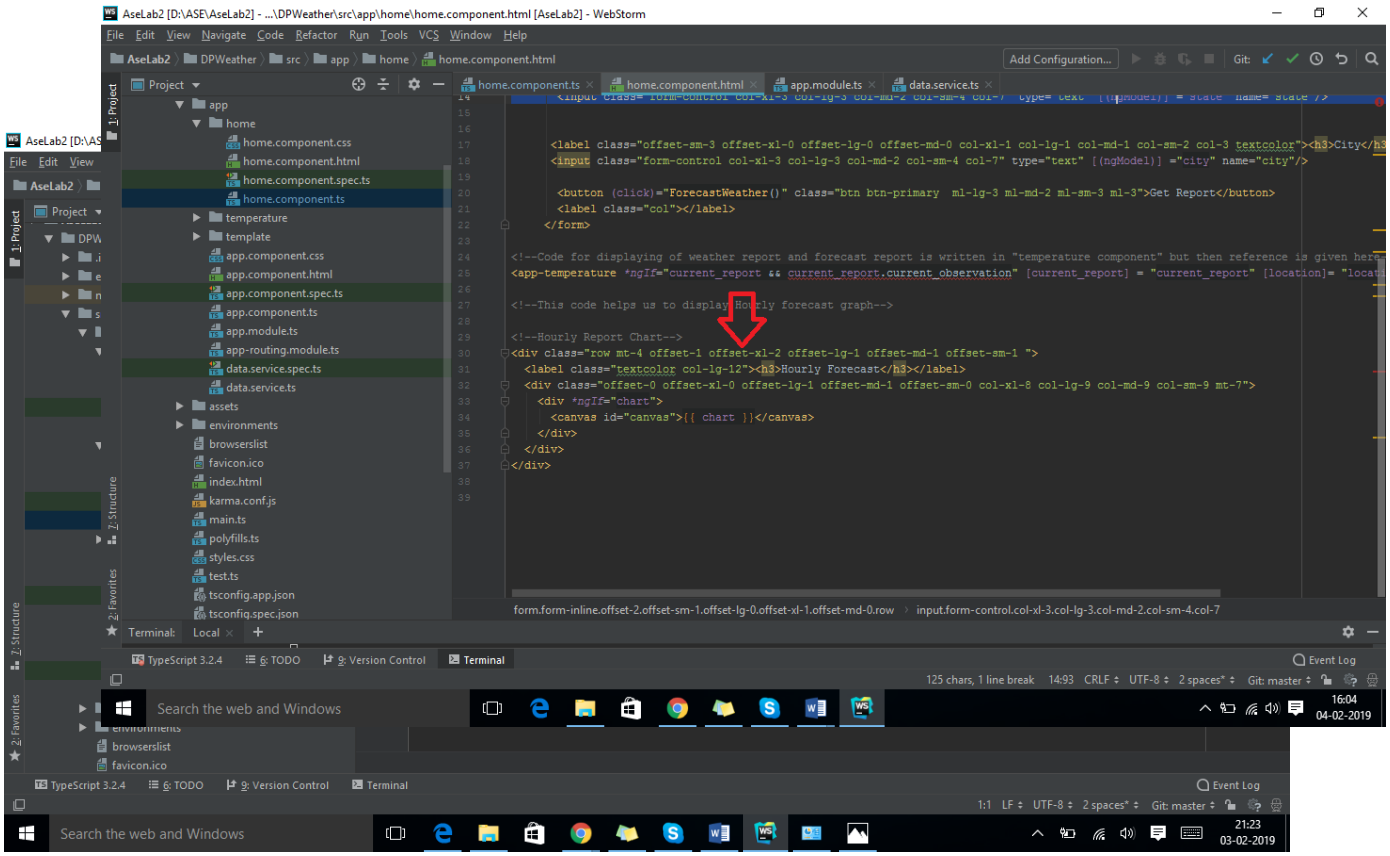
e. Once we hit “Get Report” button, we fire a request to “forecastWeather()” method which inturn calls “getDataFromService” method to get data from Data Service.



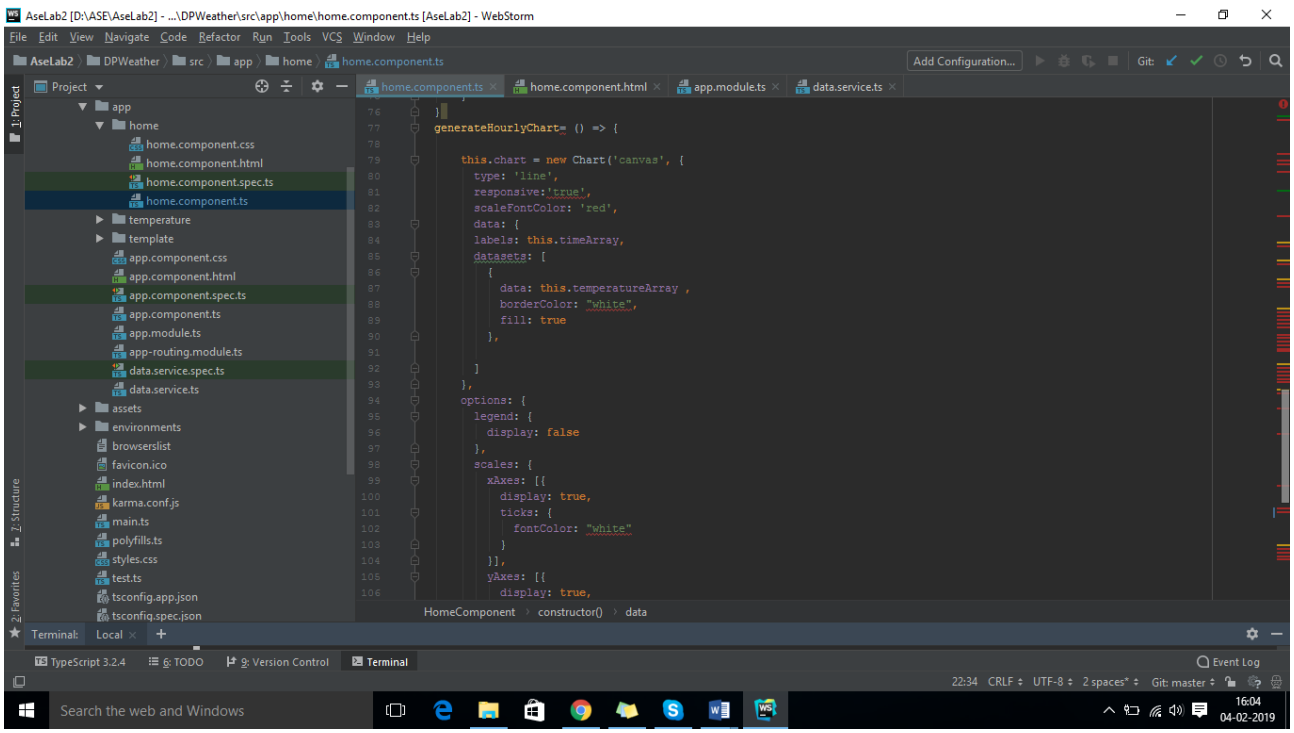
f. As I said before we have 2 components , now in order to passed data from one component (Home Component) to other component (Temperature Component) we should do Components interaction.

g. For that first we need to do property binding to the child component selector, in this way we are sending data from parent component to child component.





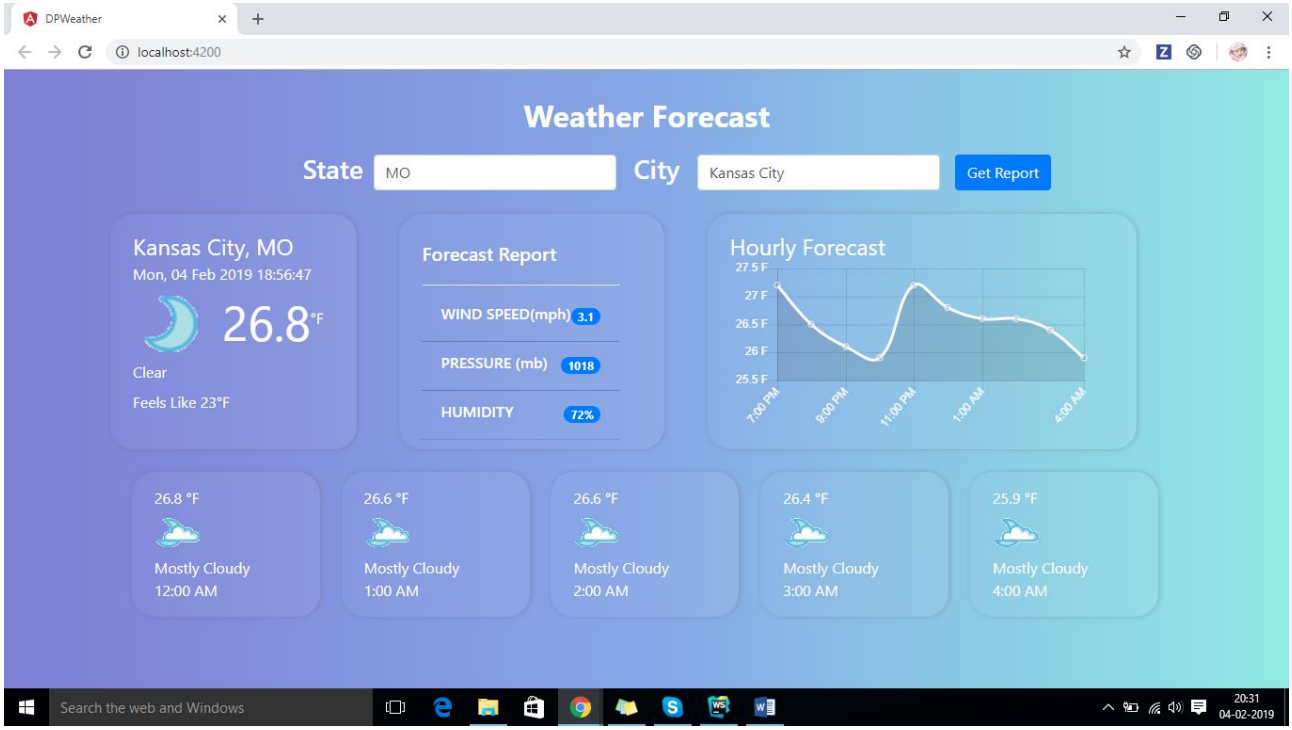
4. Now we should allow child component to receive data from parent component, that can be achieved by declaring properties using input decorator and importing “Input” from angular/core in child “.ts” file.
5. Moreover, As you can see we have also implemented chart to display hourly data ..for this we included angular

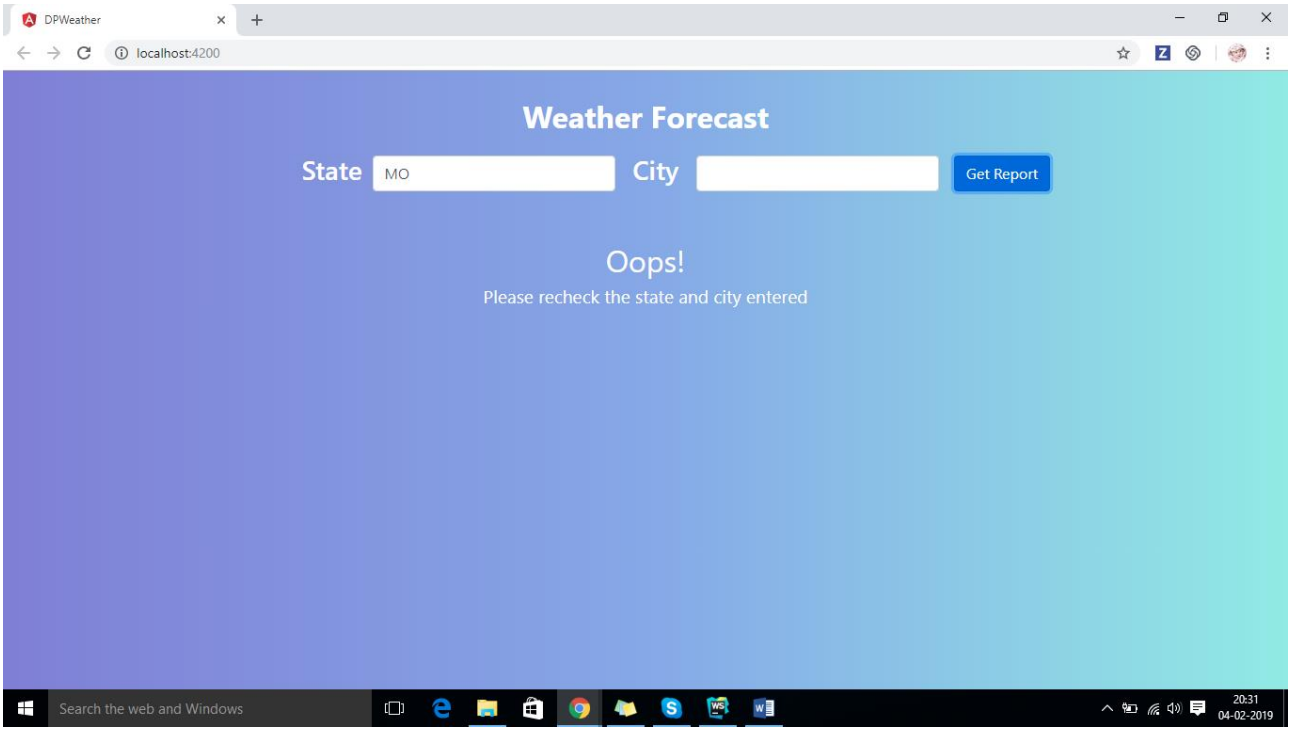


Later, the necessary business logics are implemented into each component to make sure the project satisfies all the requirements.

Design:

By default we are displaying Kansas City report.





Issues / Limitations:

Chart represented in the UI has the problem of loading previous data and current data which was fixed by nullifying the hourly report array.

Team Members:

1. Chakra Pavan Kumar Kota (16283878), ckdcb@mail.umkc.edu

2. Dharani Muli (16286308), dm6f9@mail.umkc.edu

Conclusion:

This lab Assignment motivated us to kick start working on technologies like Angular, Bootstrap, WebStorm, npm, typescript and Github and now we understood that it is our responsibility to start going deep and learn more about this technologies and we are on the way ☺ .