Weather Forecast Demo

Technical Specification

Template Version	Author	Version Date
1.0	Hui Li	April 5, 2020

Contents

1 Function Requirement

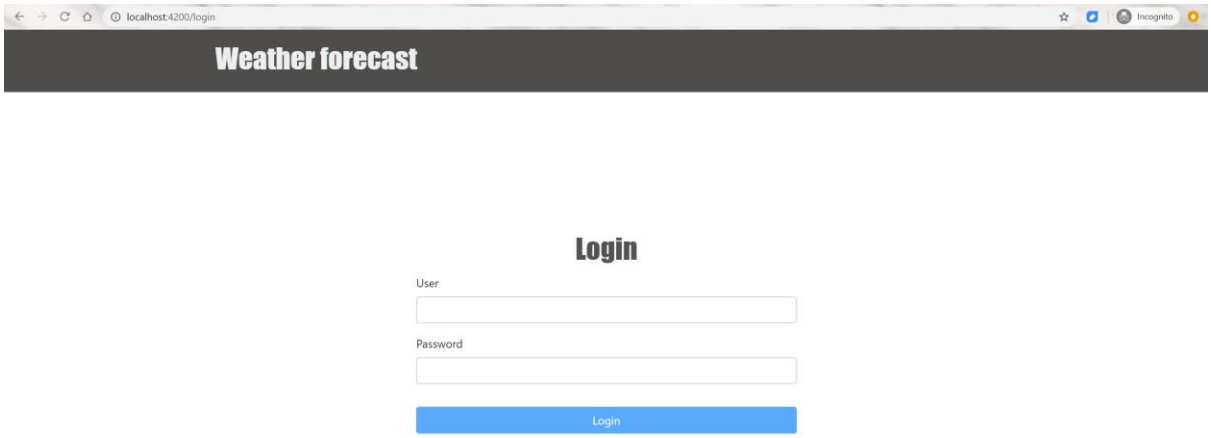
This application is designed as a demo program to show weather forecast, using open weather API: https://openweathermap.org/. The forecast is refreshed per 10 mins. Core functions refer to below:

- Search weather forecast by city name.
- Weather forecast chart in 24 hours.
- Weather forecast in 5 days

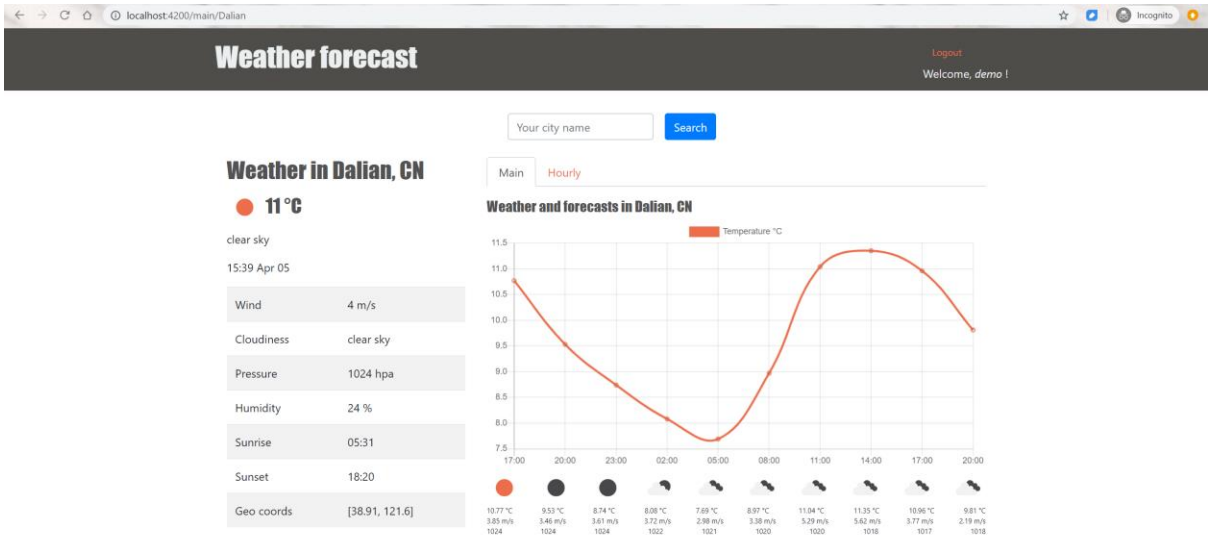
1.1 Execution Screenshot

user: demo password: 123456

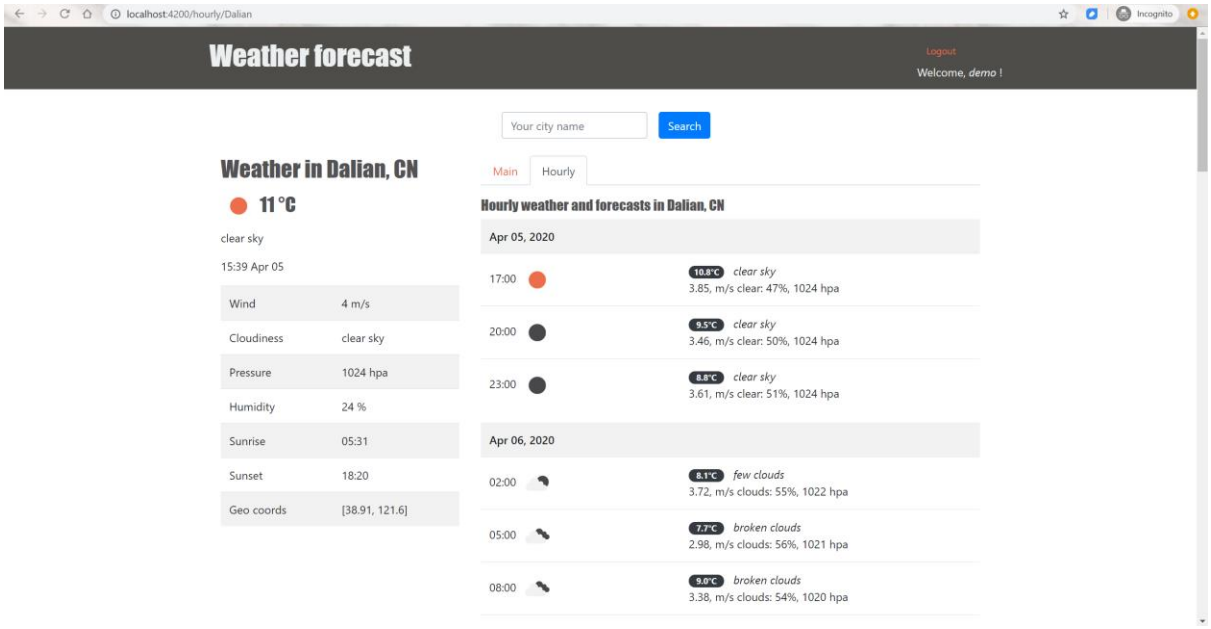
1.1.1 Login



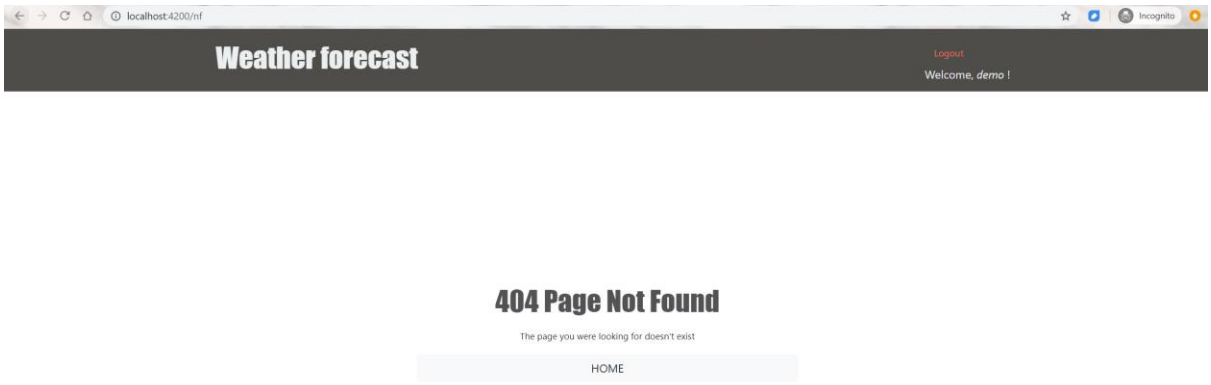
1.1.2 Weather Forecast Chart



1.1.3 Hourly Forecast List



1.1.4 404 Page



2 Program Design

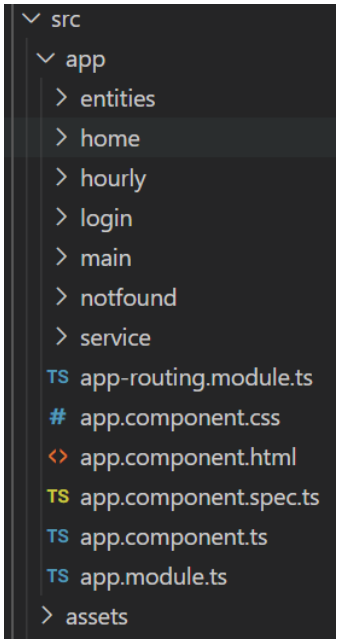
This web application is developed with Angular 9 and Bootstrap 4.4.1.

2.1 Git repository

The git repository path is *https://github.com/AmyLi1025/weather_demo.git*.

To run through this application, clone it to local IDE and execute *npm install* command to install all dependencies. Execute *ng serve --open* to run application in browser.

2.2 Program Source Framework



- entities: this folder is for instances used for data structure.
- home: component for home page.
- hourly: component for hourly tab.
- login: component for login page.
- main: component for main tab.
- notfound: component for 404 page.
- service: this folder is for services used in program.
- assets: all images used in program is stored in this folder.

2.3 Involved API

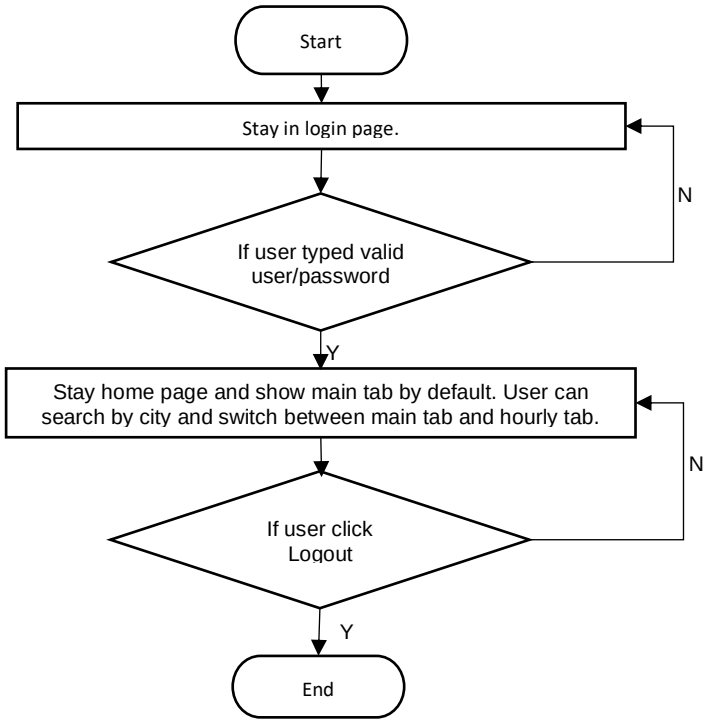
API key: 7ec083dd373bfc29ad9f3b553c1d11a4

- http://api.openweathermap.org/data/2.5/weather?q=Dalian&appid=7ec083dd373bfc29ad9f3b553c1d11a4&units=metric

- http://api.openweathermap.org/data/2.5/forecast?q=Dalian&appid=7ec083dd373bfc29ad9f3b553c1d11a4&cnt=40&units=metric

- http://api.openweathermap.org/data/2.5/forecast?q=Dalian&appid=7ec083dd373bfc29ad9f3b553c1d11a4&cnt=10&units=metric

2.4 Technical Flow Diagram



2.5 Detailed Design

2.5.1 Entity Instances

- *Forecast*: data structure of API api.openweathermap.org/data/2.5/forecast.
- *ForecastDetail*: data structure of list in API api.openweathermap.org/data/2.5/weather.
- *ForecastChart*: data structure for chart.
- *ForecastUI*: data structure for hourly tab.
- *ForecastUIDetail*: data structure of table in hourly tab.
- *Weather*: data structure of API api.openweathermap.org/data/2.5/weather.

2.5.2 Routing Module

Routing is configured in /src/app-routing.module.ts. For path '' is redirected to 'main/Dalian', and for not pattered path is routed to NotFoundComponent.

```
const routes: Routes = [{
 path: '',
 component: HomeComponent,
 children: [
  {
   path: '',
   redirectTo: 'main/Dalian',
   pathMatch: 'full'
  }, {
   path: 'main/:city',
   component: MainComponent
  }, {
   path: 'hourly/:city',
   component: HourlyComponent
  }
]
```

```
}, {
  path: 'main/:city',
  component: MainComponent
}, {
  path: 'hourly/:city',
  component: HourlyComponent
}, {
  path: 'login',
  component: LoginComponent
}, {
  path: '**',
  component: NotfoundComponent
}];
```

2.5.3 Login Page

Login page related codes are stored in /src/app/login folder. With template file login.component.html and controller file login.component.ts.

The attributes of class LoginComponent:

- *bError*: indicator for validity of user info.
- *bLoggedIn*: indicator for user logon status.
- *oFormGroup*: form group for login form.
- *USER_MAX_LENGTH*: readonly attribute with default value 30.

The main function in controller refer to below:

- *ngOnInit*: initialize login status bLoggedIn and form group oFormGroup.
- *onLogin*: event hander for submit action, call function getValidity to check if the typed user and password are valid. If not, set bError as true and show error messages in page. If valid, set cookie to store user by injected service CookieService, and emit user name by oLoginEmmiter of injected service WeatherService.
- *getValidity*: simulate the server service to check user validity.

2.5.4 Home Page

Home page related codes are stored in /src/app/home folder. With template file home.component.html and controller file home.component.ts.

The attributes of class HomeComponent:

- *oWeather*: weather information.
- *sSearchKey*: search text typed by user.
- *bNotFound*: indicator for search validity.

The main function in controller refer to below:

- *ngOnInit*: get user cookie to check if already logged. If no, navigate to login page. Otherwise call function getWeather to get weather details, and subscribe oTimerEmitter of injected service WeatherService.
- *onSearch*: call function refresh to reload weather data.
- *getWeather*: call function getWeatherByCity of service WeatherService.
- *refresh*: call function getWeather to get weather details.

2.5.5 Main Tab

Main tab related codes are stored in /src/app/main folder. With template file main.component.html and controller file main.component.ts.

The attributes of class MainComponent:

- *oChartData*: data for chart display.
- *sCity*: current city.

- *sCountry*: country information.

The main function in controller refer to below:

- *ngOnInit*: get city from route parameter, and call function getForeCast to get weather details. Subscribe oCityChangeEmitter of injected service WeatherService.
- *ngAfterViewInit*: set main tab as active status.
- *getForeCast*: call function getForecast of injected service WeatherService to retrieve weather forecast details. After data retrieved successfully, call function convertForcastData to convert weather data from open service API to local structure, then call function showLineChart to show chart.
- *showLineChart*: create line chart by Chart.js.

2.5.6 Hourly Tab

Hourly tab related codes are stored in /src/app/hourly folder. With template file hourly.component.html and controller file hourly.component.ts.

The attributes of class HourlyComponent:

- *aForeCastData*: array for weather forecast data.
- *sCity*: current city.
- *sCountry*: country information.

The main function in controller refer to below:

- *ngOnInit*: get city from route parameter, and call function getForeCast to get weather details. Subscribe oCityChangeEmitter of injected service WeatherService.
- *ngAfterViewInit*: set hourly tab as active status.
- *getForeCast*: call function getForecast of injected service WeatherService to retrieve weather forecast details. After data retrieved successfully, call function convertForcastData to convert weather data from open service API to local structure.

2.5.7 Not Found Page

Hourly tab related codes are stored in /src/app/notfound folder. With template file notfound.component.html and controller file notfound.component.ts.

The main function in controller refer to below:

- *navHome*: navigate to home page.

3 Unit Test Results

Scenario #	Step	Expected Result	Actual Test Pass/Fail
1	Run http://localhost:4200/	Navigate to login page	Pass
2	Type in user: 'demo', password: 'test'	Show error messages for invalid user	Pass
3	Type in user: 'demo', password: '123456'	Enter into home page, show 'Main' tab by default, and show Dalian weather and forecast chart	Pass
4	Select 'Hourly' tab	Switch to 'Hourly' tab and show hourly forecast list	Pass

5	Type in 'beijing' in search input	Refresh weather data of beijing	Pass
6	Per 10 mins, check if data refresh	Weather data refresh.	Pass
7	Click Logout	User logout and navigate to login page.	Pass