**<app.component.css>**

p {

font-family: Lato;

}

display {

width: 50%;

margin-left: auto;

margin-right: auto;

}

table {

width: 50%;

margin-left: auto;

margin-right: auto;

}

table, th, td {

border: 1px solid black;

border-collapse: collapse;

align-items: center;

text-align: center;

}

th, td {

padding: 5px;

text-align: left;

}

**<app.component.html>**

<table><tr><td>

<hello name="{{ name }}"></hello>

<p>

Ckick the button!

</p>

State code: <input type="text" [(ngModel)]="state"><br>

City name: <input type="text" [(ngModel)]="city"><br><br>

<button (click)="show()">Show</button>

</td></tr></table>

<div \*ngIf="result">

<table>

<tr>

<td>{{result.current\_observation.display\_location.city}}, {{result.current\_observation.display\_location.state}} <img src="{{result.current\_observation.icon\_url}}"></td>

<td>Wind: </td>

<td colspan="4">{{result.current\_observation.wind\_string}}</td>

</tr>

<tr>

<td>Pressure: </td>

<td>{{result.current\_observation.pressure\_in}}</td>

<td>Humidity: </td>

<td>{{result.current\_observation.relative\_humidity}}</td>

<td>Temperature: </td>

<td colspan="1">{{result.current\_observation.temp\_f}} F : {{result.current\_observation.temp\_c}} C</td>

</tr>

<tr>

<td>Time</td>

<td>{{result2.hourly\_forecast[0].FCTTIME.civil}}</td>

<td>{{result2.hourly\_forecast[1].FCTTIME.civil}}</td>

<td>{{result2.hourly\_forecast[2].FCTTIME.civil}}</td>

<td>{{result2.hourly\_forecast[3].FCTTIME.civil}}</td>

<td>{{result2.hourly\_forecast[4].FCTTIME.civil}}</td>

</tr>

<tr>

<td>condition</td>

<td>{{result2.hourly\_forecast[0].condition}}</td>

<td>{{result2.hourly\_forecast[1].condition}}</td>

<td>{{result2.hourly\_forecast[2].condition}}</td>

<td>{{result2.hourly\_forecast[3].condition}}</td>

<td>{{result2.hourly\_forecast[4].condition}}</td>

</tr>

<tr>

<td>Feelslike</td>

<td>{{result2.hourly\_forecast[0].feelslike.english}}</td>

<td>{{result2.hourly\_forecast[1].feelslike.english}}</td>

<td>{{result2.hourly\_forecast[2].feelslike.english}}</td>

<td>{{result2.hourly\_forecast[3].feelslike.english}}</td>

<td>{{result2.hourly\_forecast[4].feelslike.english}}</td>

</tr>

<tr>

<td>Temp</td>

<td>{{result2.hourly\_forecast[0].temp.english}}</td>

<td>{{result2.hourly\_forecast[1].temp.english}}</td>

<td>{{result2.hourly\_forecast[2].temp.english}}</td>

<td>{{result2.hourly\_forecast[3].temp.english}}</td>

<td>{{result2.hourly\_forecast[4].temp.english}}</td>

</tr>

</table>

</div>

**<app.component.spec.ts>**

import { TestBed, async } from '@angular/core/testing';

import { AppComponent } from './app.component';

describe('AppComponent', () => {

beforeEach(async(() => {

TestBed.configureTestingModule({

declarations: [

AppComponent

],

}).compileComponents();

}));

it('should create the app', () => {

const fixture = TestBed.createComponent(AppComponent);

const app = fixture.debugElement.componentInstance;

expect(app).toBeTruthy();

});

it(`should have as title 'angular-team\_1\_1'`, () => {

const fixture = TestBed.createComponent(AppComponent);

const app = fixture.debugElement.componentInstance;

expect(app.title).toEqual('angular-gzohhk');

});

it('should render title in a h1 tag', () => {

const fixture = TestBed.createComponent(AppComponent);

fixture.detectChanges();

const compiled = fixture.debugElement.nativeElement;

expect(compiled.querySelector('h1').textContent).toContain('Welcome to angular-gzohhk!');

});

});

**<app.component.ts>**

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

import {HttpClient} from '@angular/common/http';

@Component({

selector: 'app-my-app',

templateUrl: './app.component.html',

styleUrls: [ './app.component.css' ]

})

export class AppComponent {

name = 'to Lab2';

state = 'MO';

city = 'Kansas City';

result: any;

result2: any;

link1 = 'https://api.wunderground.com/api/4bbbc25f4f5946dd/conditions/q/MO/kansas%20city.json';

link2 = 'https://api.wunderground.com/api/36b799dc821d5836/hourly/q/MO/Kansas\_City.json';

constructor(private http: HttpClient) {

}

show() {

this.http.get('https://api.wunderground.com/api/4bbbc25f4f5946dd/conditions/q/' + this.state + '/' + this.city + '.json')

.subscribe((result) => {

this.result = result;

console.log(this.result);

});

this.http.get('https://api.wunderground.com/api/36b799dc821d5836/hourly/q/' + this.state + '/' + this.city + '.json')

.subscribe((result2) => {

this.result2 = result2;

console.log(this.result2);

});

}

}

**<app.module.ts>**

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

import { BrowserModule } from '@angular/platform-browser';

import { FormsModule } from '@angular/forms';

import {HttpClientModule} from '@angular/common/http';

import { AppComponent } from './app.component';

import { HelloComponent } from './hello.component';

@NgModule({

imports: [ BrowserModule, FormsModule, HttpClientModule ],

declarations: [ AppComponent, HelloComponent ],

bootstrap: [ AppComponent ]

})

export class AppModule { }

**<hello.component.ts>**

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

@Component({

selector: 'hello',

template: `<h1>Hello {{name}}!</h1>`,

styles: [`h1 { font-family: Lato; }`]

})

export class HelloComponent {

@Input() name: string;

}