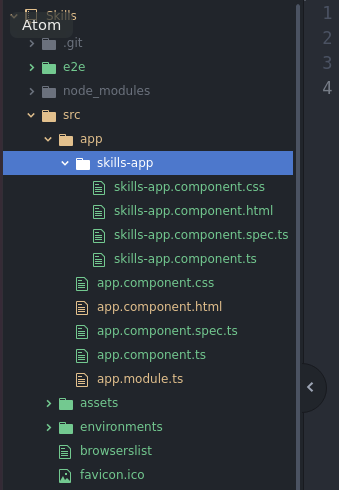
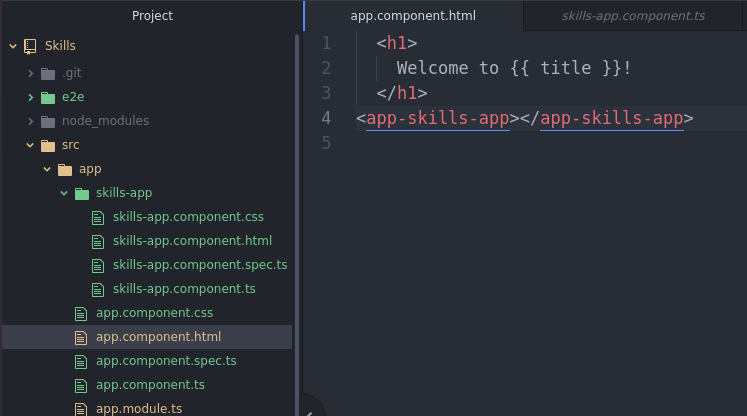
Day04 Introduction to NG 7

part 1 – Create a Custom App

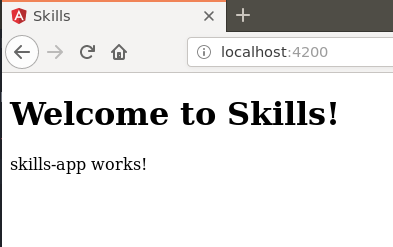
1. From Documents open in terminal window and type the command **ng new Skills**
2. Choose N for routing and CSS for styles, once done CD into the Skills folder and type **ng serve** to run the basic app. You can then open a browser to localhost:4200 to see the basic app. Note, to choose CSS use the arrow keys on the keyboard.
3. After typing ng serve, notice the word compiling, the type script (ts) code is compiled in order to run successfully
4. Open the application in Atom and most of our work will be in the app folder, which acts like the parent folder.
5. View the code of app.component.html, this is the file that feeds the default page that shows up on the browser at 4200. Remove everything except the h1 tags and content. Notice the browser refreshes automatically.
6. In the Skills folder, open in terminal window in order to start a new custom component. Once there, type **ng generate component SkillsApp**
7. Back on atom, you should see a new folder called “skills-app”  
   

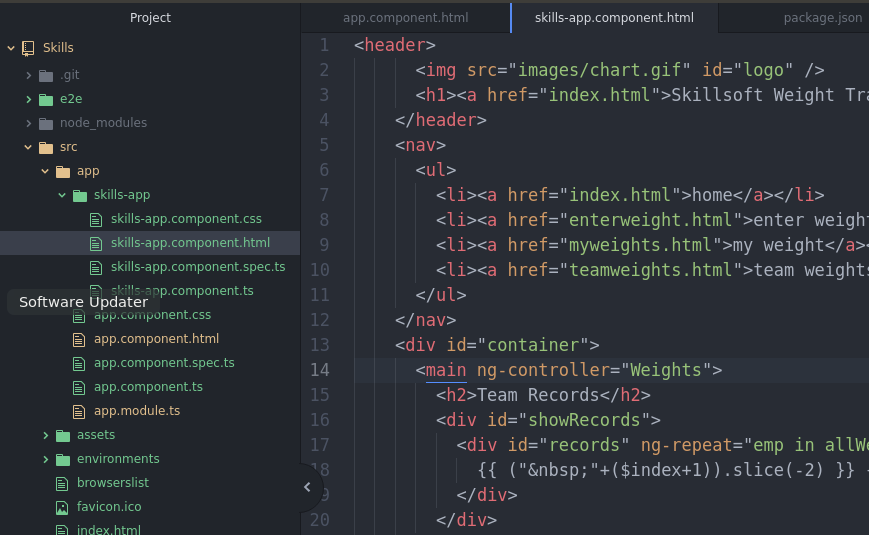
part 2 – Configuring New App

1. Open skills-app.component.html file and notice that a line of code was already inserted automatically
2. Go back to the parent folder and look at the file app.component.ts, there is a selector property with a value of app-root. Now open index.html under the original app folder and notice that between the body tags, this name appears between angle brackets.
3. Now go back to our component and inside of skills-app.component.ts file there is a selector with the name app-skills-app, lets insert this name in the app.component.html file with angle brackets.



1. Once you save, the browser will reload the new page



1. We will use the teamweights.html file as an example for the next step. Open that page in an editor and copy everything between the body tags except the script tags. Paste all this code in the skills-app.component.html file replacing the original text that was there.  
   
2. Copy the image from the original images folder into the assets folder of the new NG7 app. Change the image link in skills-app.component.html file accordingly
3. Copy the original css file and replace the NG7 css file in the src folder
4. Finally for this part, remove the Welcome to Skills text from the parent app, app.component.html

part 3 – API Calls

1. We would need the http module, so in the parent app.module.ts file import this module

|  |
| --- |
| **import { BrowserModule } from '@angular/platform-browser';**  **import { NgModule } from '@angular/core';**  **import { HttpClientModule } from '@angular/common/http';**  **import { AppComponent } from './app.component';**  **import { SkillsAppComponent } from './skills-app/skills-app.component';** |

1. Import the module at the imports section:

|  |
| --- |
| **AppComponent,**  **SkillsAppComponent**  **],**  **imports: [**  **BrowserModule,**  **HttpClientModule**  **],**  **providers: [],**  **bootstrap: [AppComponent]** |

(remember to insert a comma at the line above)

1. Back in the child component, go into the skills-app.component.ts file and import the HttpClient like we did in the parent ts file

|  |
| --- |
| **import { Component, OnInit } from '@angular/core';**  **import { HttpClient } from '@angular/common/http';**  **@Component({** |

1. Now in the constructor, pass in the HttpClient

|  |
| --- |
| **export class SkillsAppComponent implements OnInit {**  **constructor(private http: HttpClient) { }**  **ngOnInit() {** |

1. Within that class, create a new property call it allweights:

|  |
| --- |
| **export class SkillsAppComponent implements OnInit {**  **allWeights: any;**  **constructor(private http: HttpClient) { }** |

1. Once the app starts, the method ngOnInit() will fire, in here we can use our http object to make the call

|  |
| --- |
| **allWeights: any;**  **constructor(private http: HttpClient) { }**  **ngOnInit() {**  **this.http.get('http://localhost:8000/getallrecords');**  **}** |

1. Assign the observable returned from the get() to allWeights

|  |
| --- |
| **ngOnInit() {**  **this.allWeights = this.http.get('http://localhost:8000/getallrecords');**  **}** |

1. Now back on the html file, remove the old AngularJS code and the ng-repeat code
2. Insert this code instead

|  |
| --- |
| **<h2>Team Records</h2>**  **<div id="showRecords">**  **<div id="records" \*ngFor="let emp of allWeights | async">**  **</div>**  **</div>**  **</main>** |

We extract each document and pipe it to an async method just in case the content changes

1. Now we have access to the individual fields

|  |
| --- |
| **<h2>Team Records</h2>**  **<div id="showRecords">**  **<div id="records" \*ngFor="let emp of allWeights | async">**  **{{emp.empName | json}} weighed {{emp.empWeight}} on [date].**  **</div>**  **</div** |

