UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

3416-003 Web Development Using Angular 8 and TypeScript

Module 4: Angular Components

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 1: Section 1

What Is This Course About

Installed Bootstrap 4.2.1

- Created directory C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version1
- In dos-prompt: cd C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version1
- Step 1:Type “ng new bootstrap4”
- Click “Yes” for “Would you like to add Angular routing”.
- Click “enter” for “Which stylesheet format would you like to use?”.
- Cd into directory: “cd bootstrap4”

```
C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1>ng new bootstrap4
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
```

```
C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1>cd bootstrap4

C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1\bootstrap4>
```

Installed Bootstrap 4.1.3

- Type following command:
- Step 2: npm install popper.js@1.14.3

```
C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1\bootstrap4>npm install popper.js@1.14.3
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

+ popper.js@1.14.3
added 1 package from 2 contributors and audited 43277 packages in 12.474s
found 0 vulnerabilities
```

Installed Bootstrap 4.1.3

- Type following command:
- Step 3: npm install jquery@3.4.1

```
C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1\bootstrap4>npm install jquery@3.4.1
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

+ jquery@3.4.1
added 1 package from 1 contributor and audited 43278 packages in 11.555s
found 0 vulnerabilities


C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1\bootstrap4>
```

Installed Bootstrap 4.1.3

- Type following command:
- Step 4: npm install bootstrap@4.1.3

```
C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1\bootstrap4>npm install bootstrap@4.1.3
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

+ bootstrap@4.1.3
added 1 package from 2 contributors and audited 43279 packages in 11.523s
found 1 moderate severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details

C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version1\bootstrap4>
```

Installed Bootstrap 4.2.1

- Type following command:
- Step 5: Open C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version1\bootstrap4\angular.json and add the following:
- "node_modules/bootstrap/dist/css/bootstrap.min.css“

```
"styles": [
            "src/styles.css",

"node_modules/bootstrap/dist/css/bootstrap.min.css"

          ]
```

Installed Bootstrap 4.2.1

angular.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "bootstrap4": {
      "root": "",
      "sourceRoot": "src",
      "projectType": "application",
      "prefix": "app",
      "schematics": {},
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "dist/bootstrap4",
            "index": "src/index.html",
            "main": "src/main.ts",
            "polyfills": "src/polyfills.ts",
            "tsConfig": "src/tsconfig.app.json",
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "src/styles.css",
               "node_modules/bootstrap/dist/css/bootstrap.min.css"
            ],
            "scripts": [],
            "es5BrowserSupport": true
          },
          "configurations": {
            "production": {
              "fileReplacements": [
                {
                  "replace": "src/environments/environment.ts",
                  "with": "src/environments/environment.prod.ts"
                }
              ],
              "optimization": true,
              "outputHashing": "all",
              "sourceMap": false,
              "extractCss": true,
              "namedChunks": false,
              "aot": true,
              "extractLicenses": true,
              "vendorChunk": false,
              "buildOptimizer": true,
              "budgets": [
                {
                  "type": "initial",
                  "maximumWarning": "2mb",
                  "maximumError": "5mb"
                }
              ]
            }
          }
        }
      }
```

Add this line

Installed Bootstrap 4.2.1

- Type following command:
- Step 6: Open C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version1 \bootstrap4\angular.json and add the following:

```
"scripts": [
            "node_modules/jquery/dist/jquery.slim.min.js",
            "node_modules/bootstrap/dist/js/bootstrap.min.js",
            "node_modules/popper.js/dist/umd/popper.min.js"


            ]
```

Installed Bootstrap 4.2.1

- Type following command:
- Step 6: Open C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version1\bootstrap4\angular.json and add the following:

```
"scripts": [
            "node_modules/jquery/dist/jquery.slim.min.js",
            "node_modules/bootstrap/dist/js/bootstrap.min.js",
            "node_modules/popper.js/dist/umd/popper.min.js"
            ]
```

Installed Bootstrap 4.2.1

- Type following command:

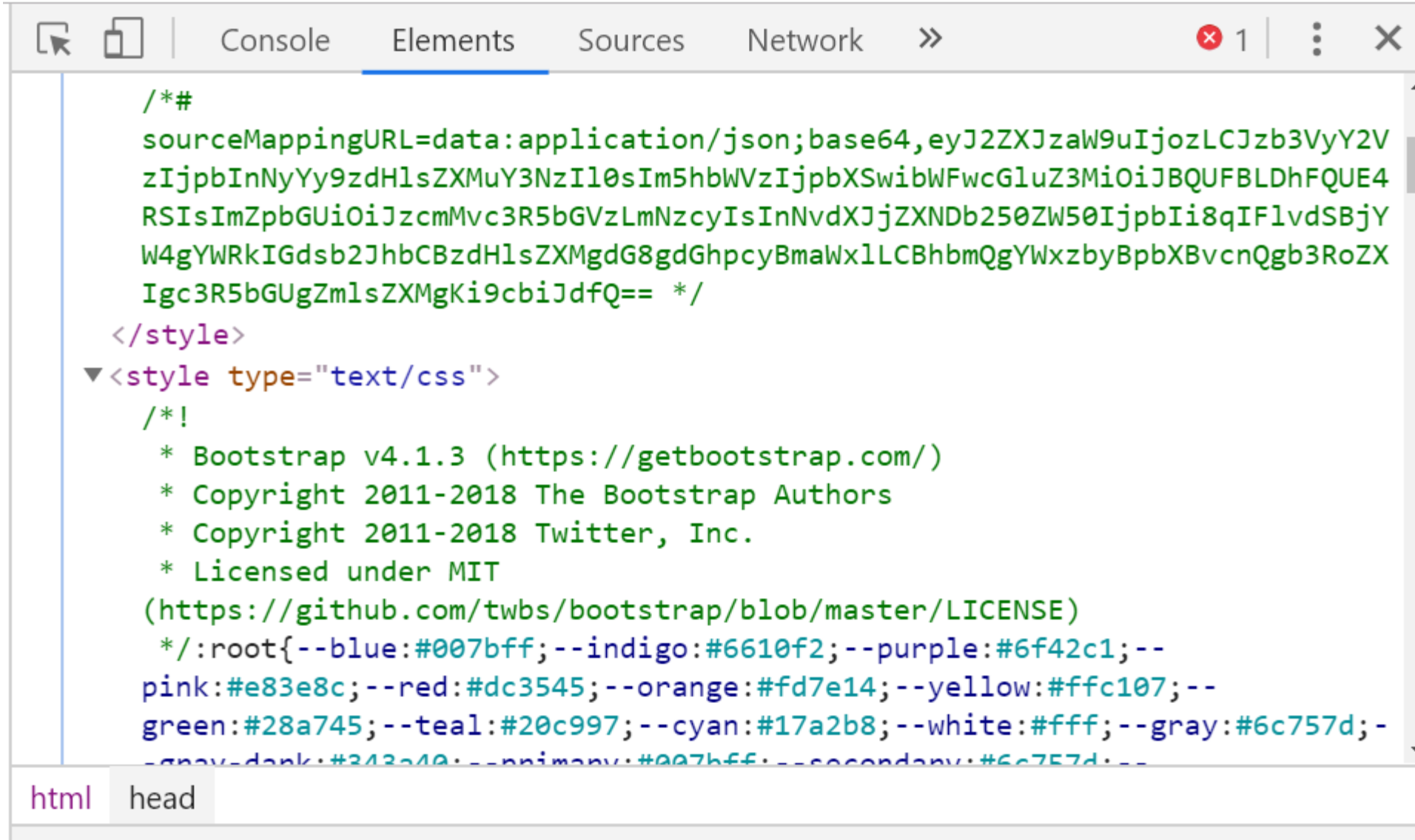
```
"projects": {
  "bootstrap4": {
    "root": "",
    "sourceRoot": "src",
    "projectType": "application",
    "prefix": "app",
    "schematics": {},
    "architect": {
      "build": {
        "builder": "@angular-devkit/build-angular:browser",
        "options": {
          "outputPath": "dist/bootstrap4",
          "index": "src/index.html",
          "main": "src/main.ts",
          "polyfills": "src/polyfills.ts",
          "tsConfig": "src/tsconfig.app.json",
          "assets": [
            "src/favicon.ico",
            "src/assets"
          ],
          "styles": [
            "src/styles.css",
            "node_modules/bootstrap/dist/css/bootstrap.min.css"
          ],
          "scripts": [
          "node_modules/jquery/dist/jquery.slim.min.js",
          "node_modules/bootstrap/dist/js/bootstrap.min.js",
          "node_modules/popper.js/dist/umd/popper.min.js"


          ],
```

Installed Bootstrap 4.2.1

- Type following command:
- Step 7: Open C:\workspace-UT-Angular7-May-2019\session7\example1-bootstrap4-version1\bootstrap4\ng serve -o

Installed Bootstrap 4.2.1

- Right click inspect and choose “Elements” tab.



How an Angular App gets Loaded and Started

- C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version1\bootstrap4 to C:\ workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version2
- Run “ng serve –o”.
- In C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version2\bootstrap4\src\app\app.component.html delete the content and replaced with “I’m in the AppComponent!”

src ▸ app ▸ <> app.component.html ▸ ...

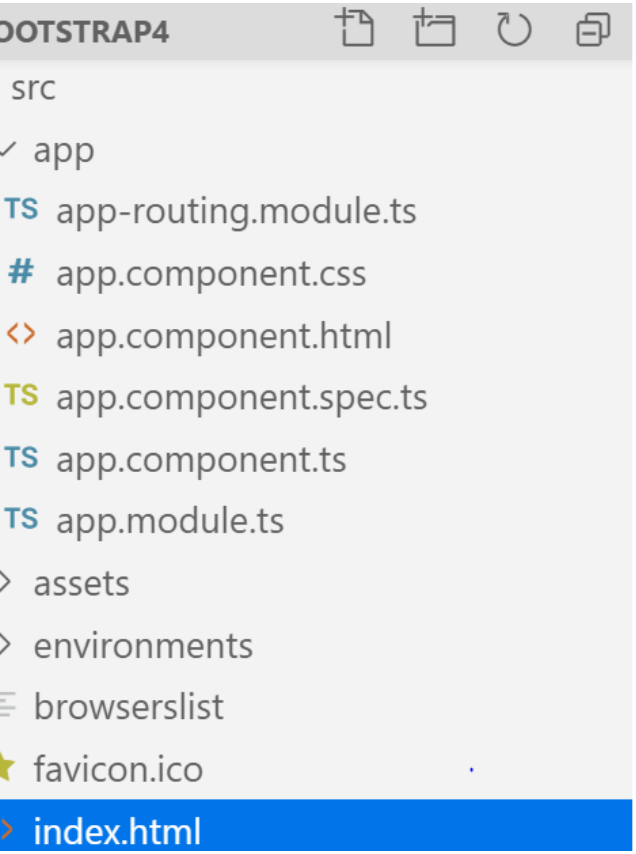
```
1   I’m in the AppComponent!
2
3   <router-outlet></router-outlet>
4
```

How an Angular App gets Loaded and Started

- How does our browser or how does a server hosting our app know that it should render the content of app component html. You could argue, it's the only component we have right now and we will take a closer look at component's soon. But that is not the reason. And actually this is not the file served by the server. Instead the index HTML file is served by the server and remembered that angular is a framework which allows you to create single page application. This is the single page which has served the index HTML file.

How an Angular App gets Loaded and Started

- Now if you have a look at this index.html file we see that this is a normal HTML file.(C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version2\bootstrap4\src\index.html)




```
1   <!doctype html>
2   <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Bootstrap4</title>
6     <base href="/">
7
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <link rel="icon" type="image/x-icon" href="favicon.ico">
10  </head>
11  <body>
12    <app-root></app-root>
13  </body>
14  </html>
15
```

How an Angular App gets Loaded and Started

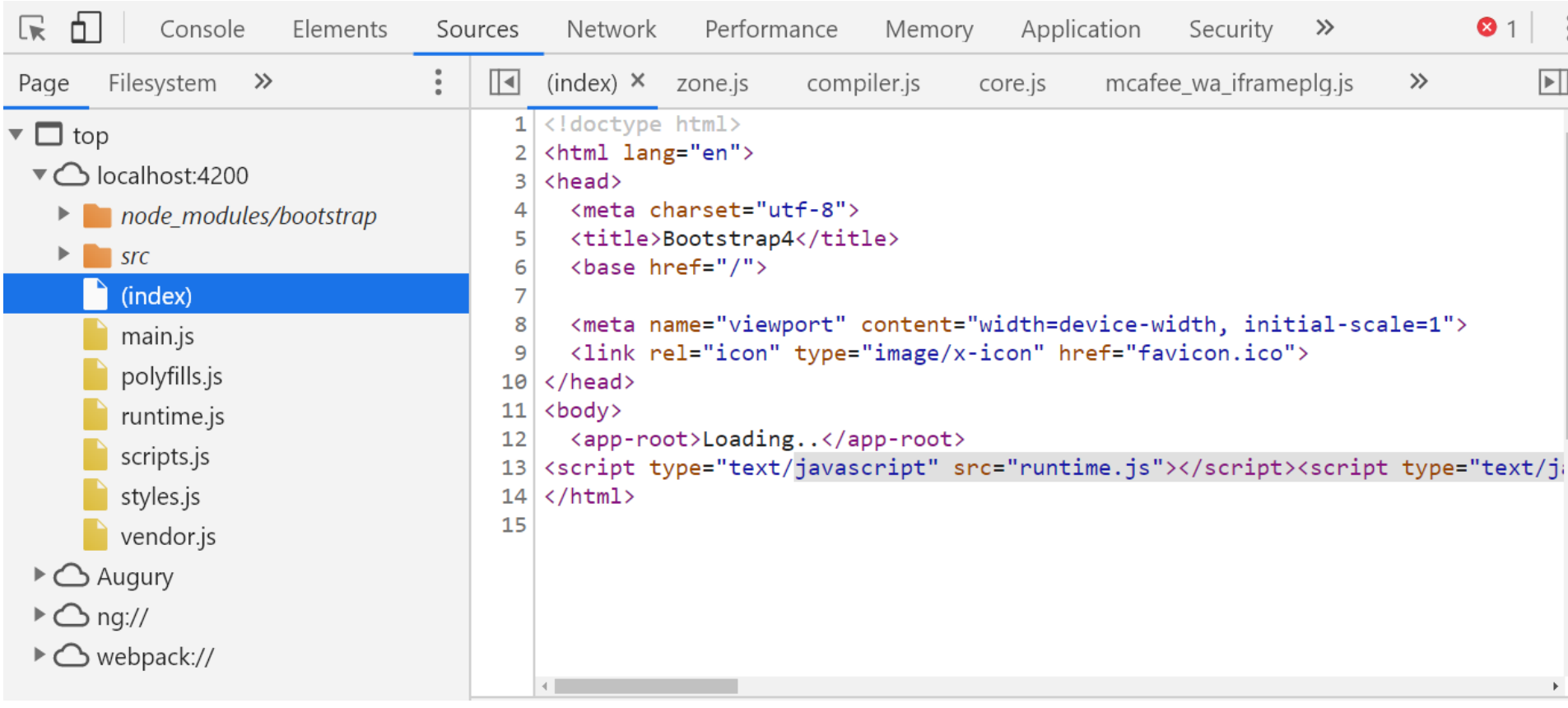
- The body of the index.html file is very interesting. We get this ‘<app-root></app-root>’ . We added:
- <app-root>Loading..</app-root>

```
<body>
  <app-root>Loading..</app-root>
</body>
```

How an Angular App gets Loaded and Started

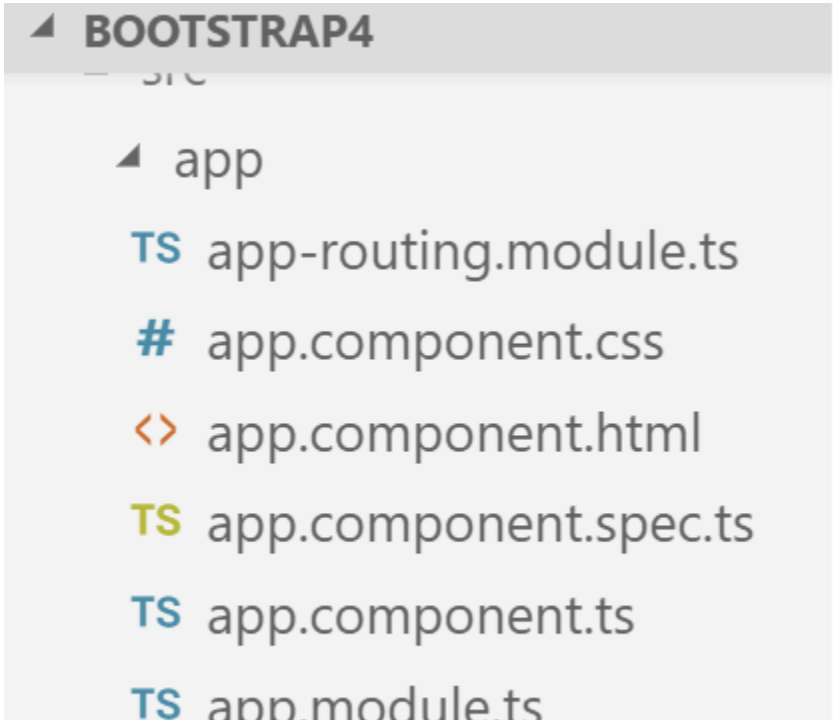
- Don’t see “loading..” in the output in the browser:

I’m in the AppComponent!



How an Angular App gets Loaded and Started

- “<app-root>” here is of course not our default html element. Instead as is one of our own components we will soon dive into how we create our own components but the CLI created one for us. The root component of our application, the component which will tie together our whole application in the end and all the files in the app folder here which have component in their name.

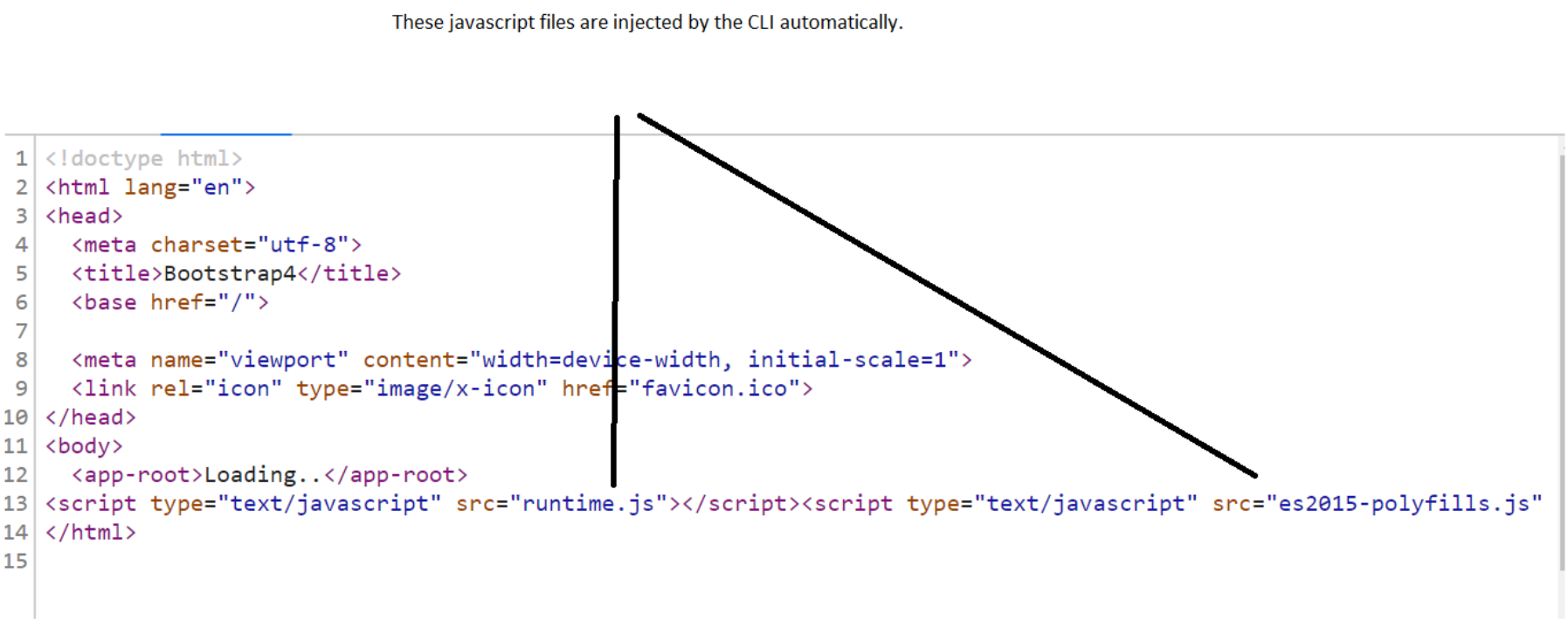


How an Angular App gets Loaded and Started

- The “app.component.ts” file

How an Angular App gets Loaded and Started

- How is Angular trigger, how is it kicked off to actually run over our ‘app.component.html’ file? The answer is in the final index page of the file getting served in the browser an a bunch of javascript files are injected by the CLI.

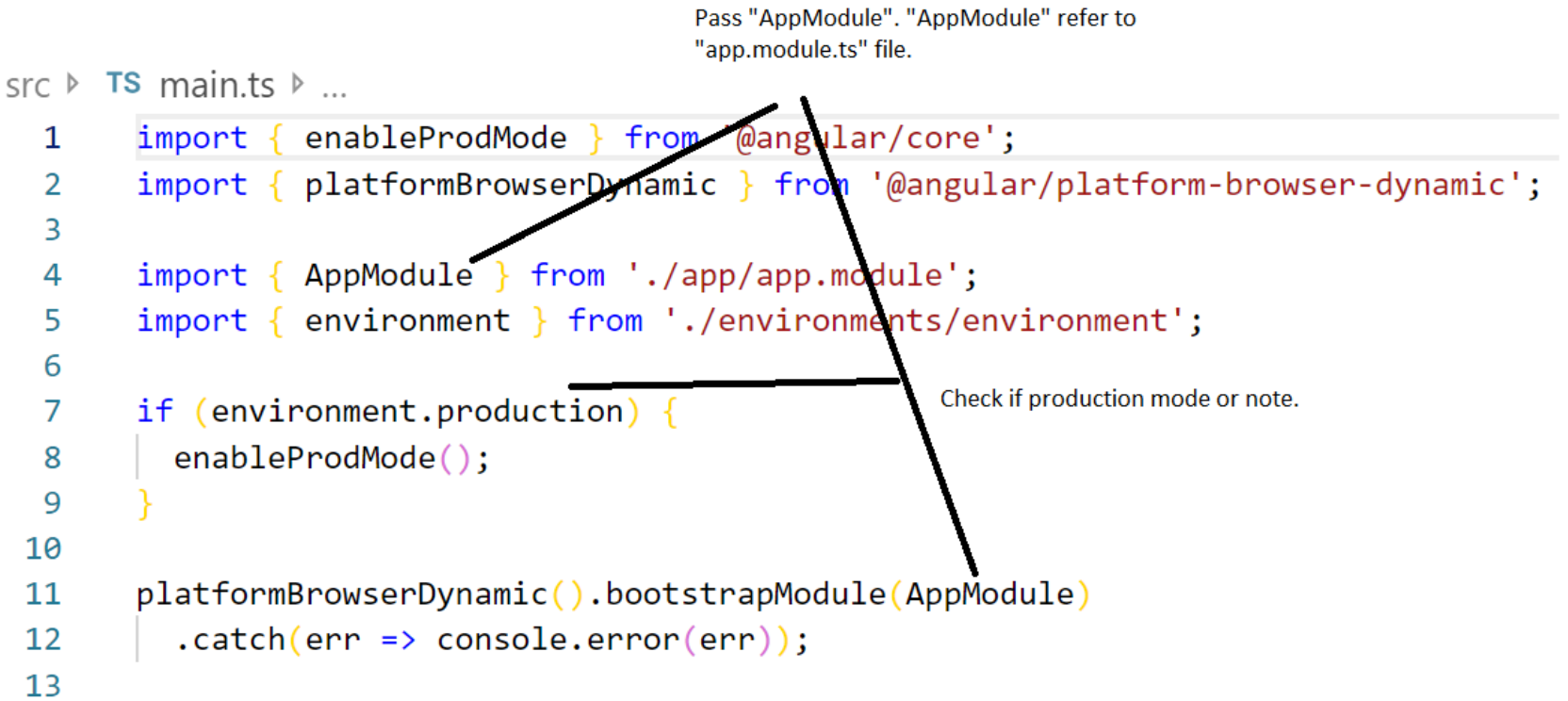


How an Angular App gets Loaded and Started

- So in the final file, script imports here are present and these javascript script files are there for executed and they're actually the first code to be executed.

Main.ts

- Main.ts is the first code get executed.



App.module.ts

- Copy C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version2\bootstrap4 to

C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version3

- The app.module.ts(C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version3\bootstrap4\src\app\app.module.ts) file

App.module.ts

src ▸ app ▸ TS app.module.ts ▸ ...

```
1   import { BrowserModule } from '@angular/platform-browser';
2   import { NgModule } from '@angular/core';
3
4   import { AppRoutingModule } from './app-routing.module';
5   import { AppComponent } from './app.component';
6
7   @NgModule({
8     declarations: [
9       AppComponent
10    ],
11    imports: [
12      BrowserModule,
13      AppRoutingModule
14    ],
15    providers: [],
16    bootstrap: [AppComponent]
17  })
18  export class AppModule { }
19
```

Reference our 'AppComponent' in 'app.component.ts'. Angular analyze 'AppComponent' and know there is a selector called 'app-root' and now Angular able to handle 'app-root' in the index.html file. Angular knows to insert the html in 'app.component.html'

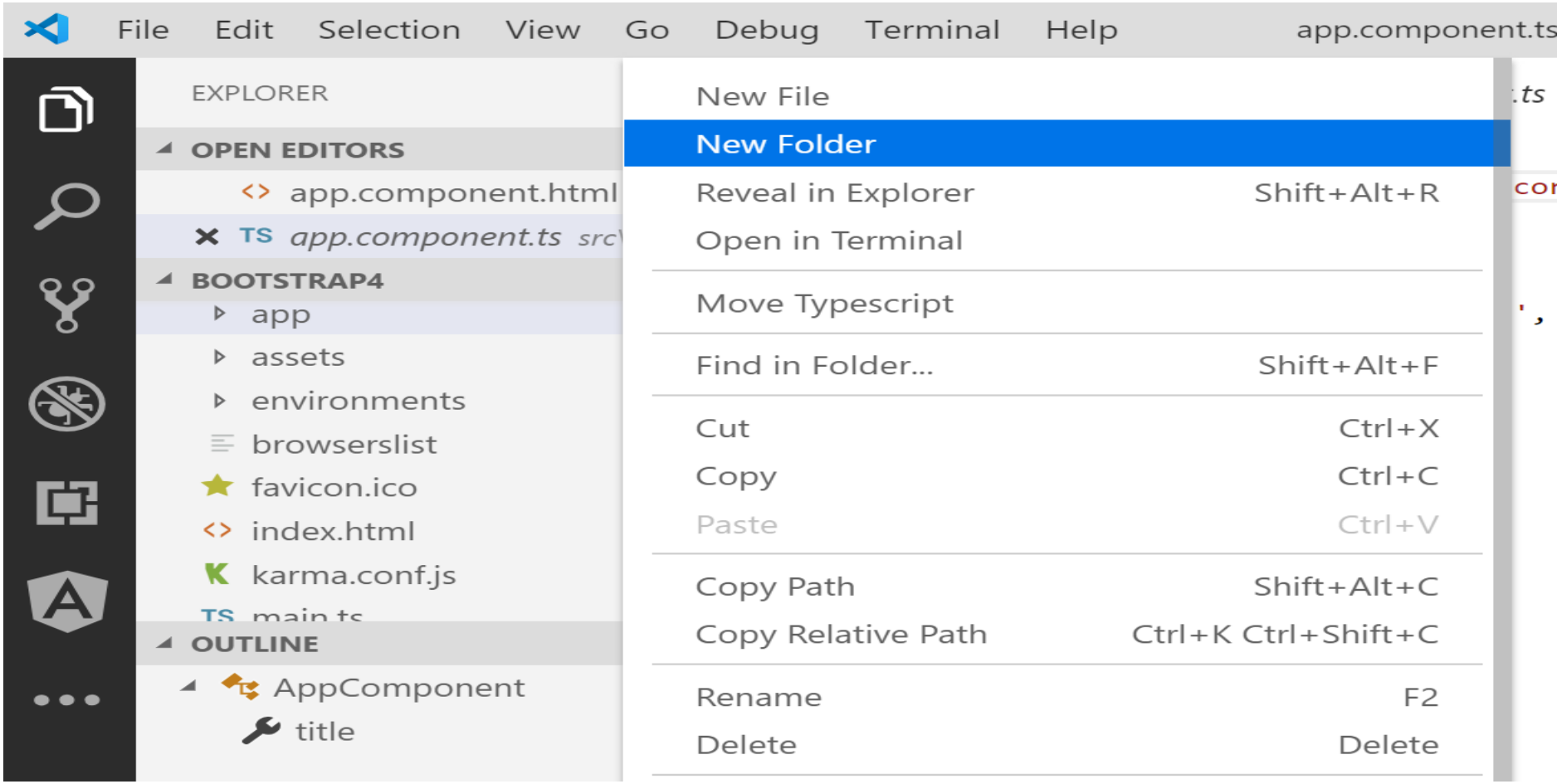
Components are important

- Components are a key feature in Angular. You build a whole application from a couple of components, which you created on your own. We started from “appComponent”, the root component, which holds our entire application in the end. So the root component will be the template where we will add our other components.

Creating a new component

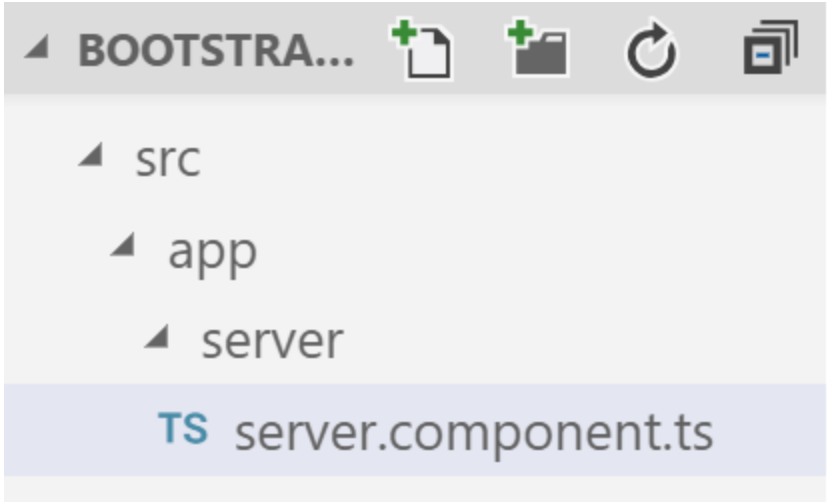
- All new components selectors will be added to the “app.component.html” file.
- Let’s say we want to output some information about a server. We’re building a back-end for our server management application and we want to output some server information.
- Create a new folder inside “app” folder called “server”.
- Right click “app” folder and choose “new folder”.

Creating a new component



Creating a new component

- Right click on “server” folder and choose new file. Name it “server.component.ts”.



Creating a new component

- A component is a typescript class.

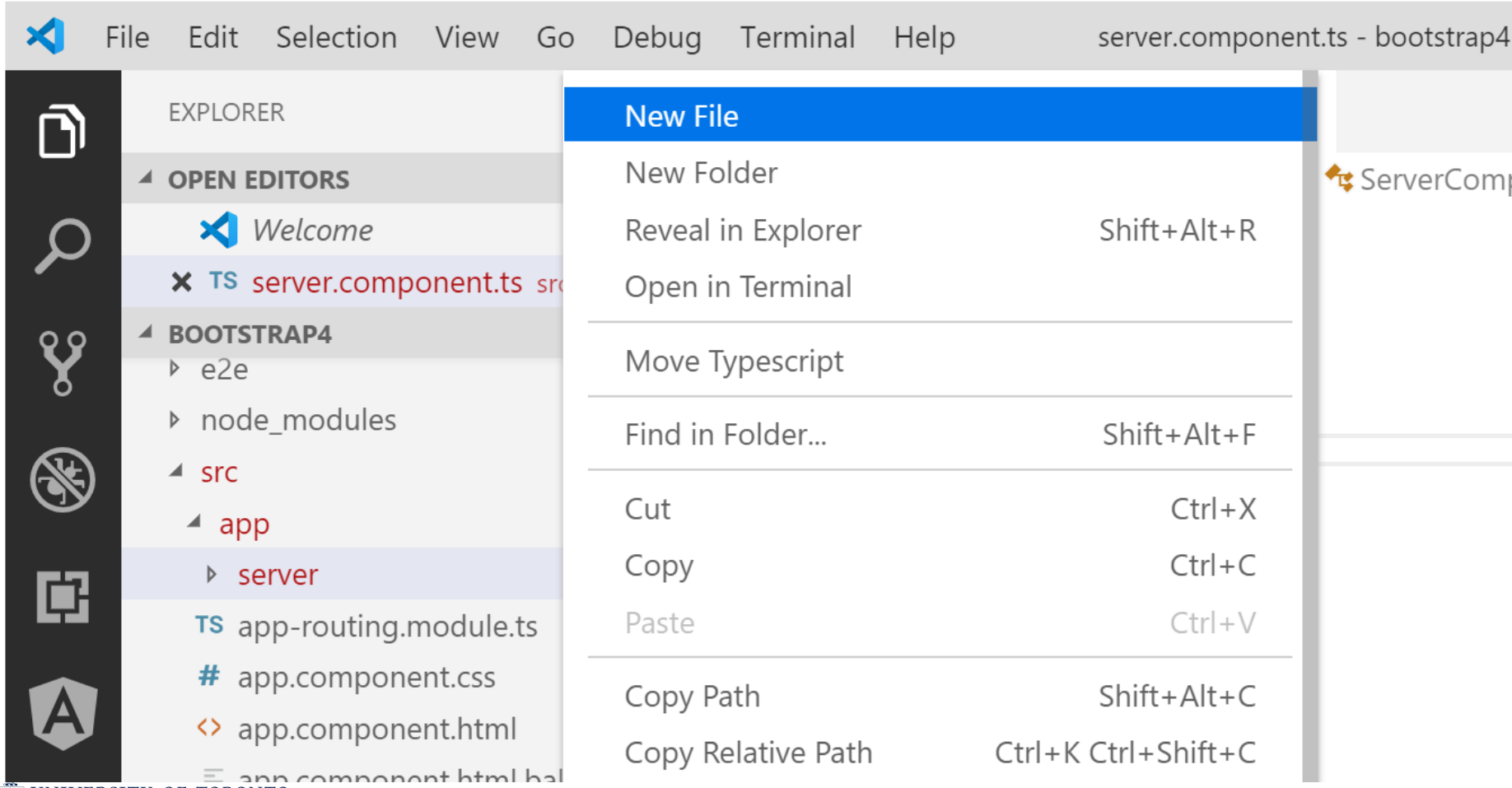
Creating a new component

src ▸ app ▸ server ▸ TS server.component.ts ▸ ...

```
1   import { Component } from '@angular/core';
2
3   @Component({
4     selector: 'app-server',
5     templateUrl: './server.component.html'
6   })
7   export class ServerComponent {
8
9   }
10
11
12
```

Creating a new component

- Right click on server folder and choose new file.



Creating a new component

- Type “server.component.html” file.

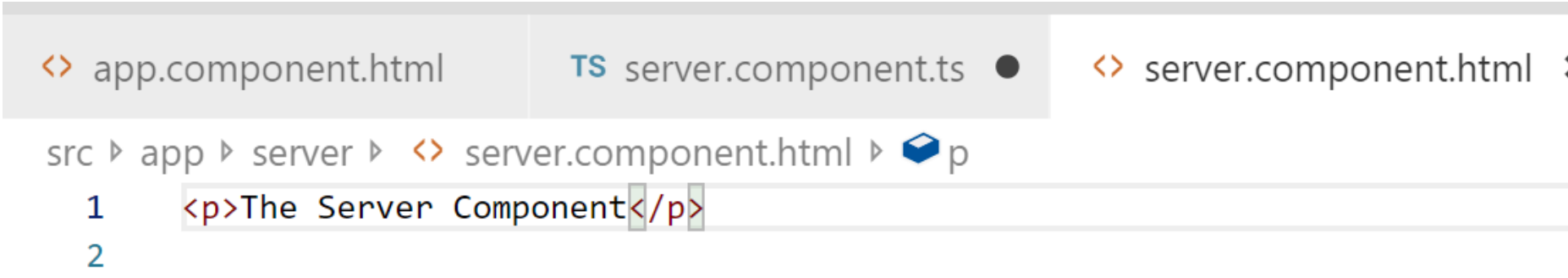
TS app.module.ts | TS server.component.ts ● | <> server.component.html

src > app > server > TS server.component.ts > ServerComponent

```
1   import { Component } from '@angular/core';
2
3   @Component({
4     selector: 'app-server',
5     templateUrl: './server.component.html'
6
7   })
8   export class ServerComponent {
9
10  }
11
```

Creating a new component

- Type the following in “server.component.html”.



App.module.ts

- What is “app.module.ts”?
- Angular uses components to build web pages and uses modules to bungle different pieces. For example, components of your apps to packages. A module is basically a bungle of functionality of your app. It basically gives Angular information which features does my app have?
- Just creating the server.component.ts is not enough. Angular doesn’t know it exists. In order to let Angular know the server.component.ts exist, we have to register it in the app.module.ts, specifically in the “@NgModule”.

App.module.ts

- Original code:

src ▸ app ▸ TS app.module.ts ▸ ...

```
1   import { BrowserModule } from '@angular/platform-browser';
2   import { NgModule } from '@angular/core';
3
4   import { AppRoutingModule } from './app-routing.module';
5   import { AppComponent } from './app.component';
6
7   @NgModule({
8     declarations: [
9       AppComponent
10    ],
11    imports: [
12      BrowserModule,
13      AppRoutingModule
14    ],
15    providers: [],
16    bootstrap: [AppComponent]
17  })
18  export class AppModule { }
19
```

App.module.ts

- new code:

src ▸ app ▸ TS app.module.ts ▸ AppModule

```
1   import { BrowserModule } from '@angular/platform-browser';
2   import { NgModule } from '@angular/core';
3
4   import { AppRoutingModule } from './app-routing.module';
5   import { AppComponent } from './app.component';
6   import { ServerComponent } from './server/server.component';
7
8   @NgModule({
9     declarations: [
10      AppComponent,
11      ServerComponent
12    ],
13    imports: [
14      BrowserModule,
15      AppRoutingModule
16    ],
17    providers: [],
18    bootstrap: [AppComponent]
19  })
20  export class AppModule { }
```

Add two new lines

Simply add other modules into this module.
IMporting some modules into Angular

App.module.ts

- Run following:
- C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version3\bootstrap4>npm install @angular/http@latest or npm install @angular/http@7.2.15

```
C:\workspace-UT-Angular7-Sept-2019\Session4\example1- bootstrap4-version3\bootstrap4>npm install @angular/http@7.2.15
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

+ @angular/http@7.2.15
added 1 package from 1 contributor and audited 43281 packages in 44.04s
found 1 moderate severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details

C:\workspace-UT-Angular7-Sept-2019\Session4\example1- bootstrap4-version3\bootstrap4>
```

Creating a new component

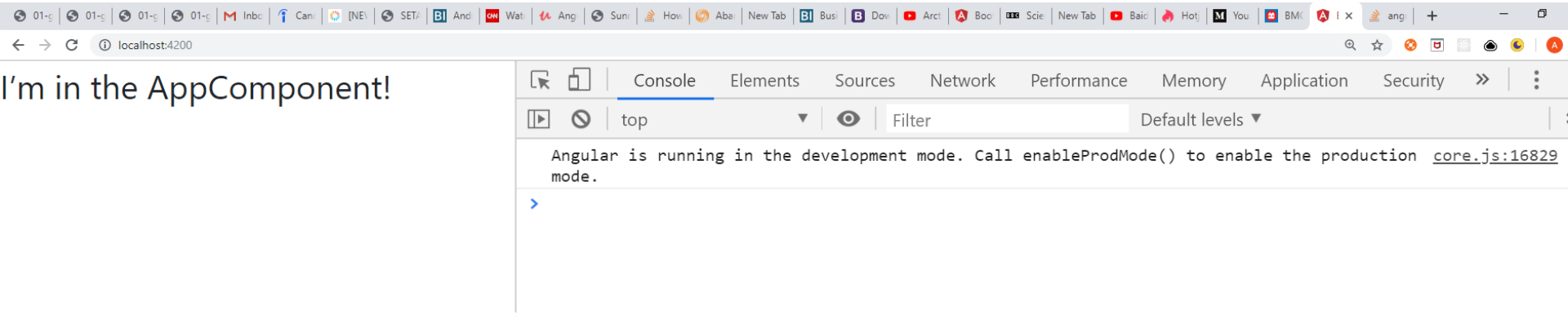
src ▸ app ▸ TS app.module.ts ▸ ...

```
1   import { BrowserModule } from '@angular/platform-browser';
2   import { NgModule } from '@angular/core';
3
4   import { FormsModule } from '@angular/forms';
5   import { HttpModule } from '@angular/http';
6
7   import { AppRoutingModule } from './app-routing.module';
8   import { ServerComponent } from './server/server.component';
9   import { AppComponent } from './app.component';
10
11
12
13  @NgModule({
14    declarations: [
15      AppComponent,
16      ServerComponent
17    ],
18    imports: [
19      BrowserModule,
20      FormsModule,
21      HttpModule,
22      AppRoutingModule
23    ],
24    providers: [],
25    bootstrap: [AppComponent]
26  })
27  export class AppModule { }
28
```

Add new lines of code

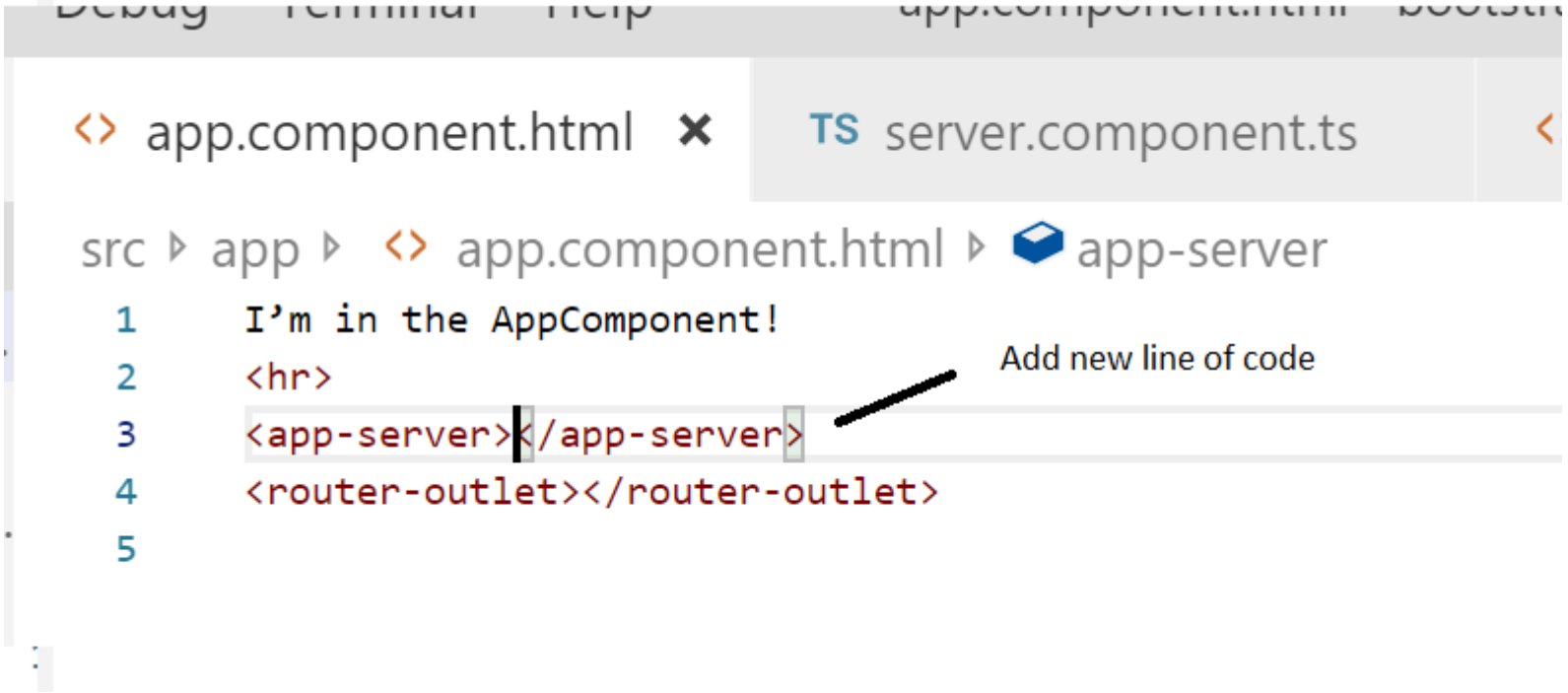
Creating a new component

- Make sure still working:



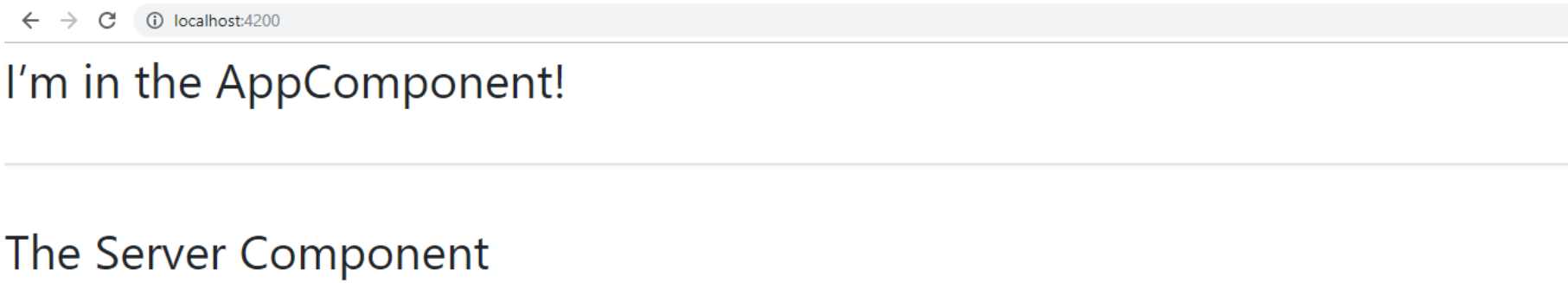
Using Custom Components

- Add new line of code in app.component.html:



Using Custom Components

- Look at output in google chrome.

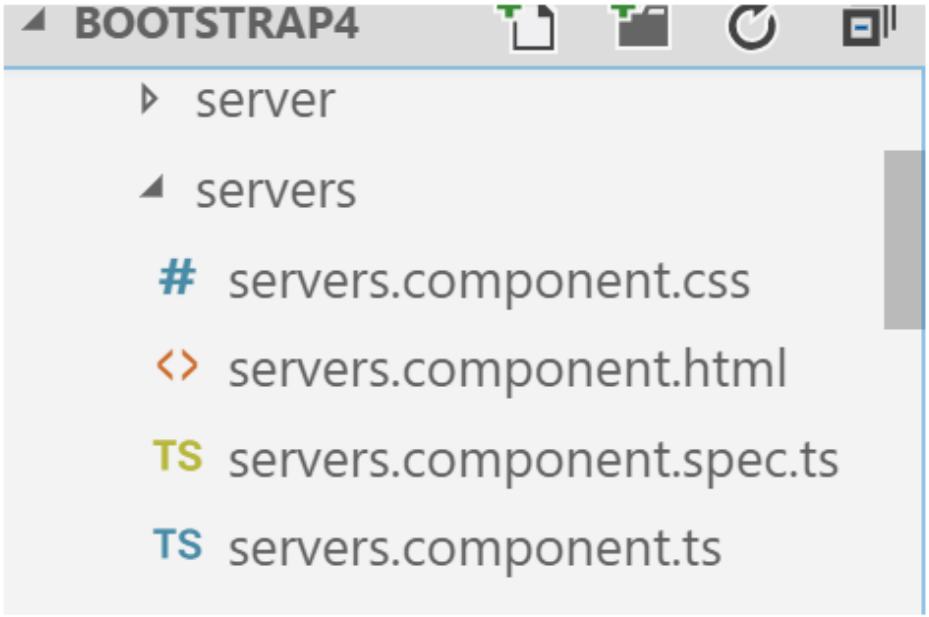


Generate Custom Components

- Copy C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version3\bootstrap4 to C:\ workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version4
- Stop current CLI and change to directory C:\workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version4\bootstrap4
- In dos-prompt type:ng generate component servers or ng g c servers

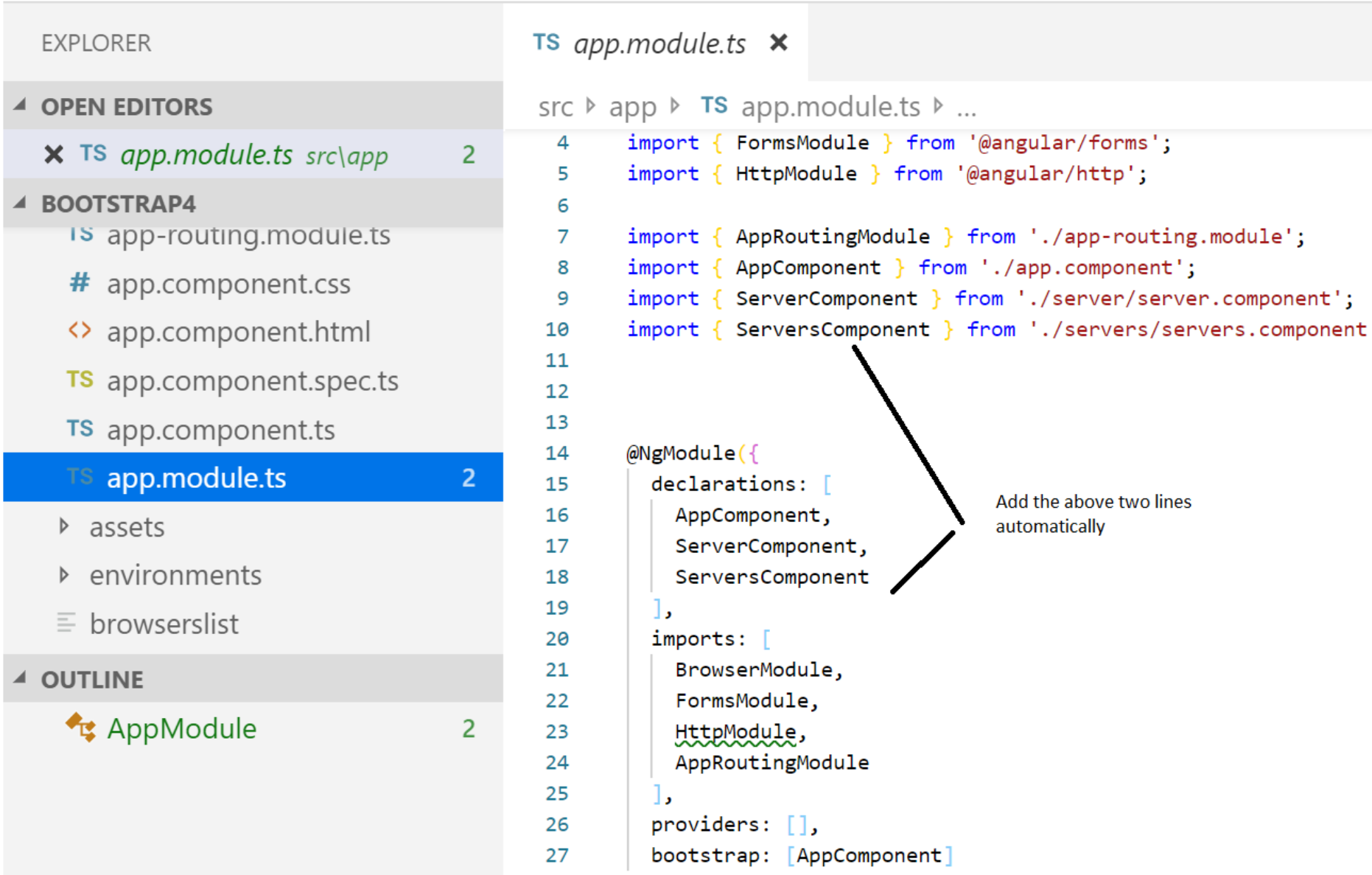
```
C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version4\bootstrap4>ng g c servers
CREATE src/app/servers/servers.component.html (26 bytes)
CREATE src/app/servers/servers.component.spec.ts (635 bytes)
CREATE src/app/servers/servers.component.ts (273 bytes)
CREATE src/app/servers/servers.component.css (0 bytes)
UPDATE src/app/app.module.ts (685 bytes)

C:\workspace-UT-Angular7-Sept-2019\Session4\example1-bootstrap4-version4\bootstrap4>
```

Generate Custom Components

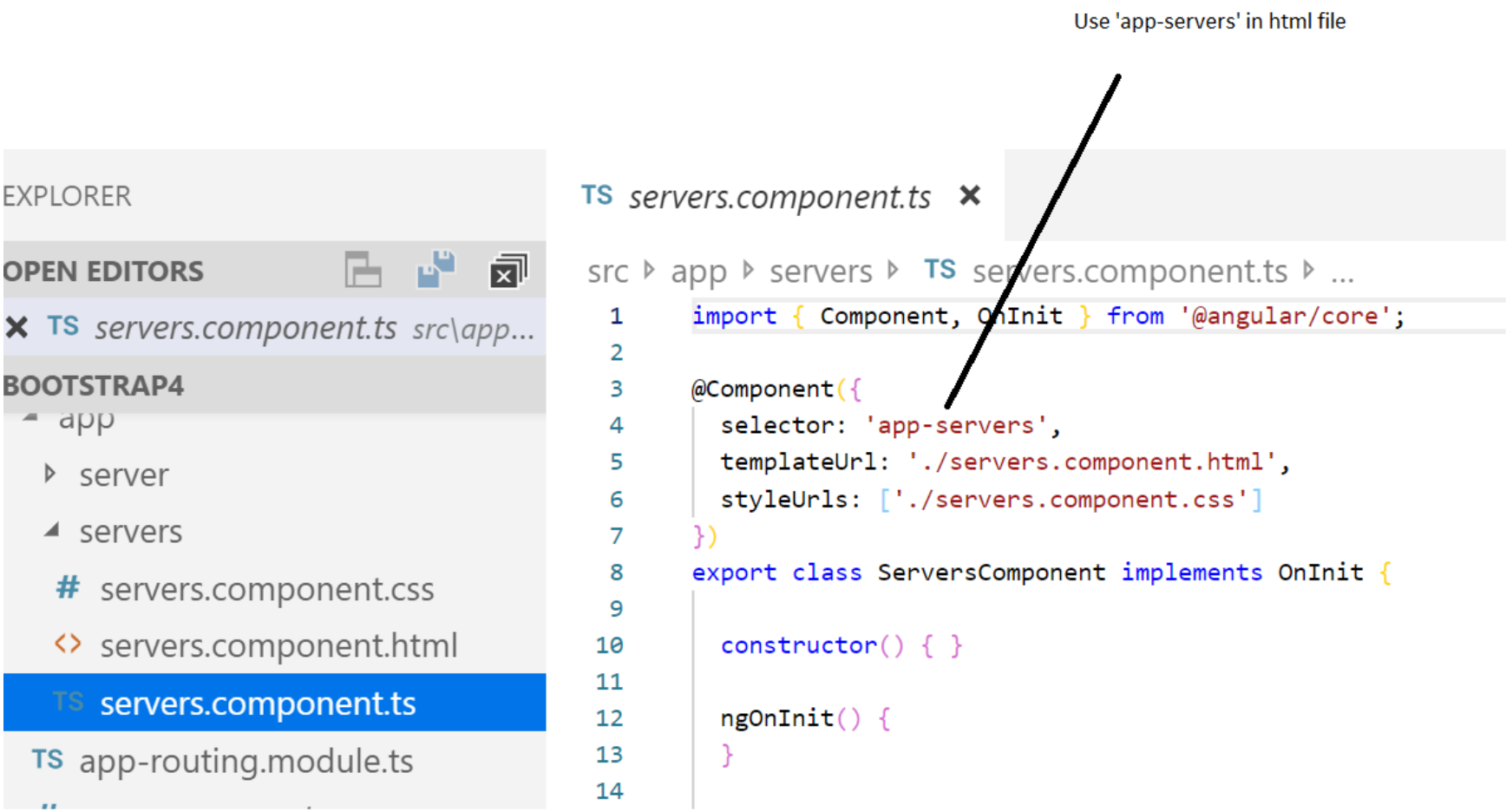


Delete 'server.component.spec.ts' because it is a testing file

Generate Custom Components

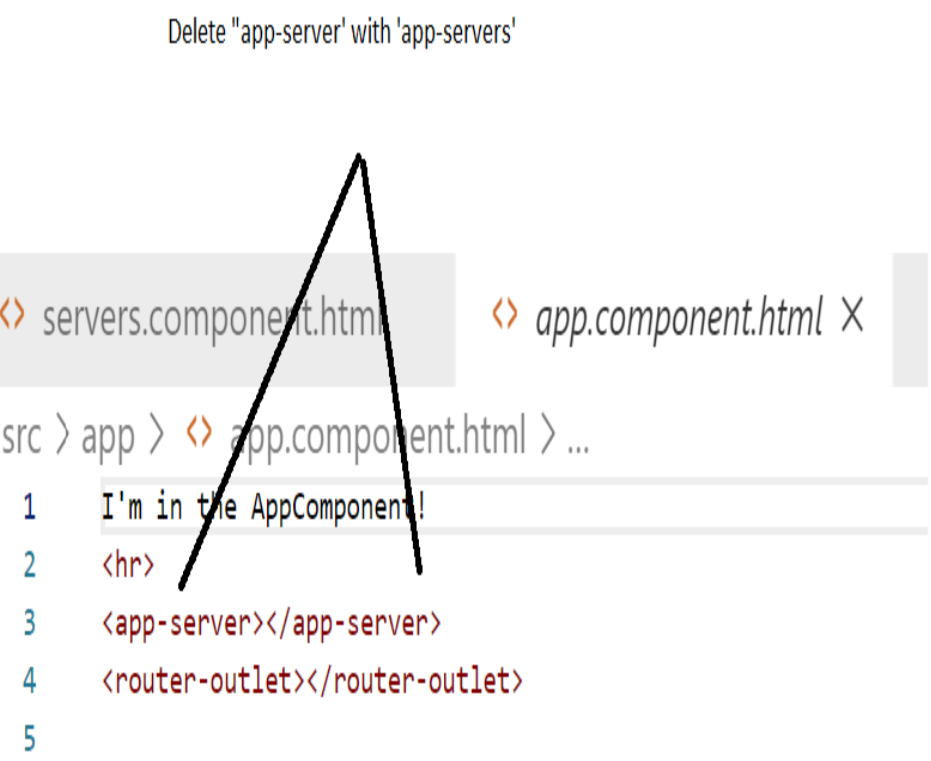


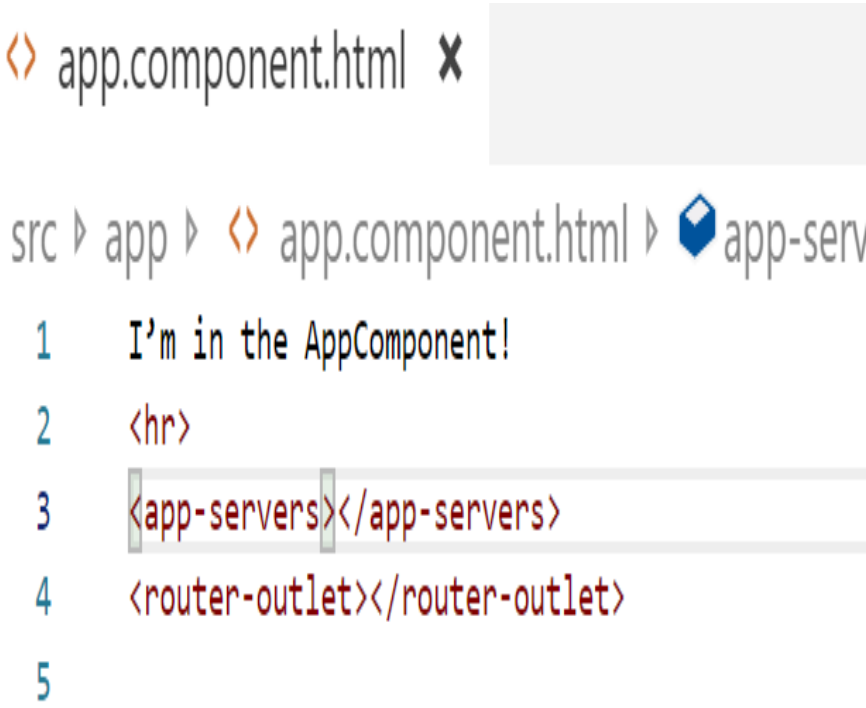
Generate Custom Components



Generate Custom Components

- In C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version4\bootstrap4\src\app\app.component.html use “app-servers” instead of “app-server”.





Generate Custom Components

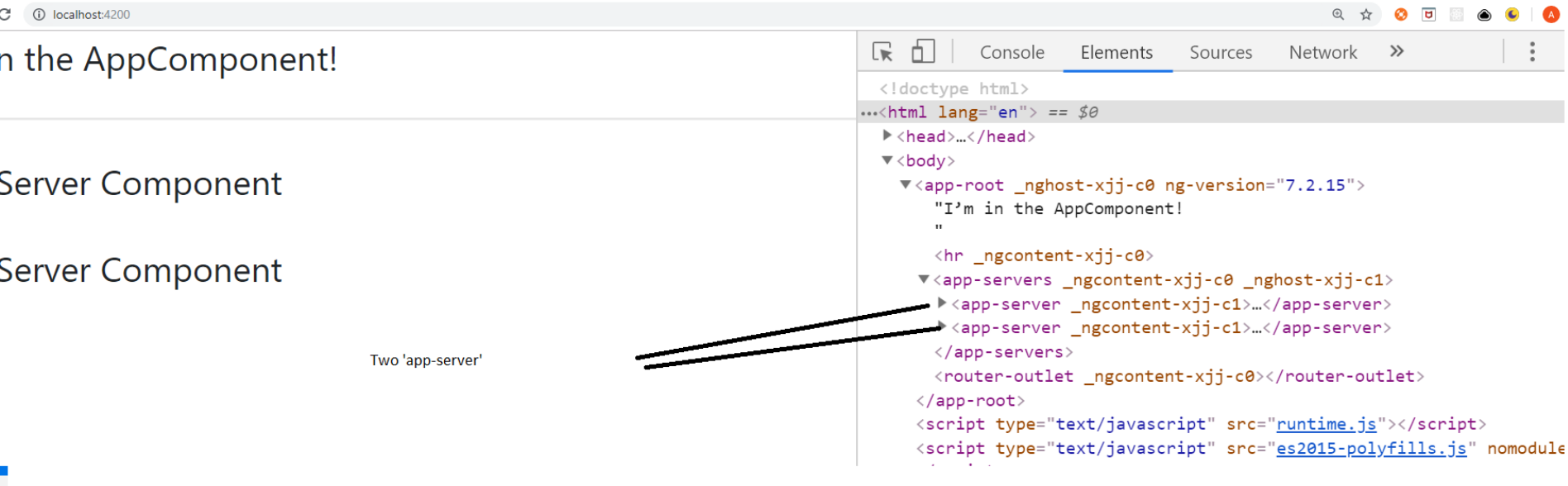
- In servers.component.html:

app.component.html | servers.component.html

src ▸ app ▸ servers ▸ servers.component.html ▸ ...

```
1    <app-server></app-server>
2    <app-server></app-server>
3
```

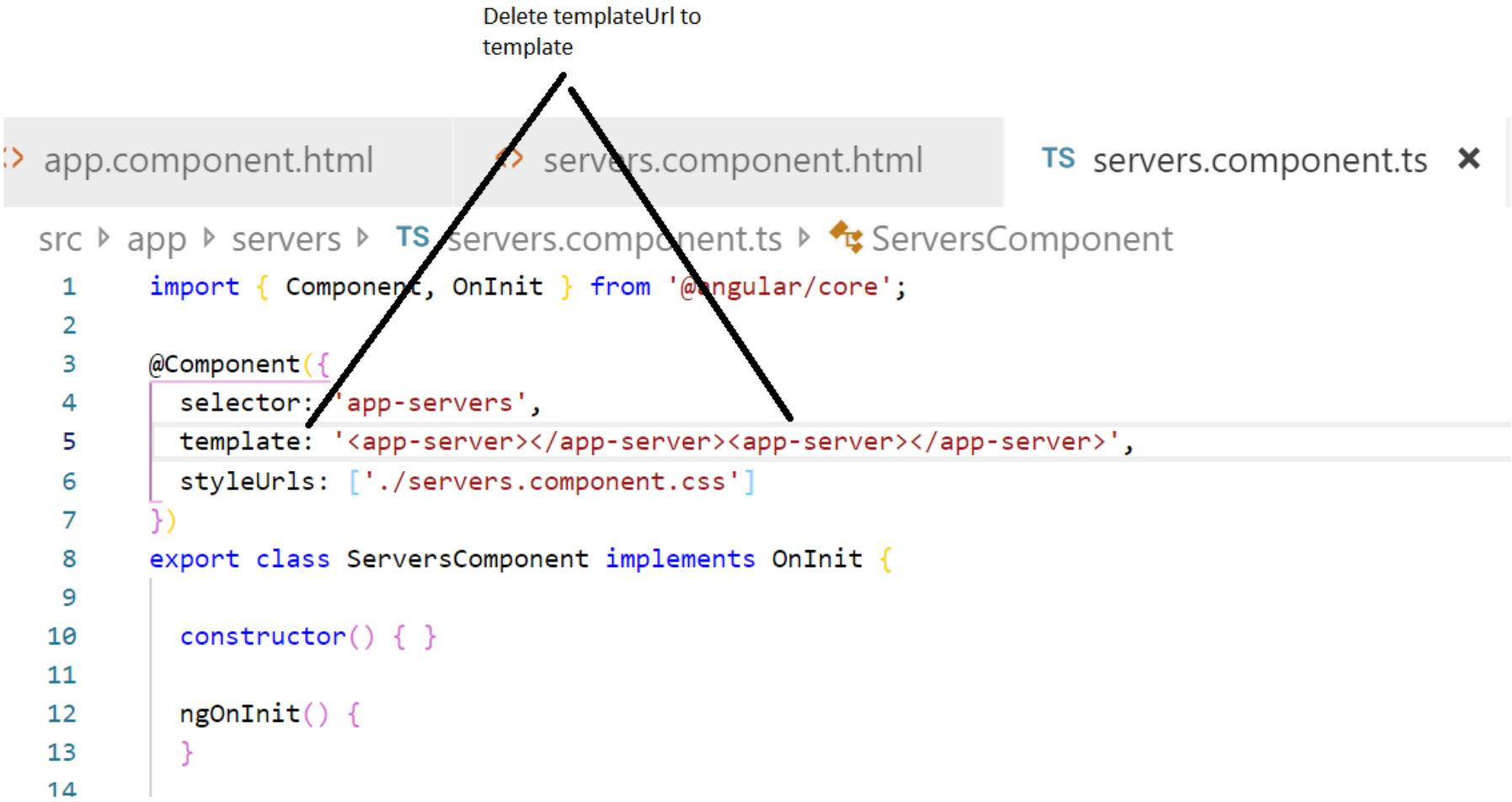
Generate Custom Components



Working with Component Templates

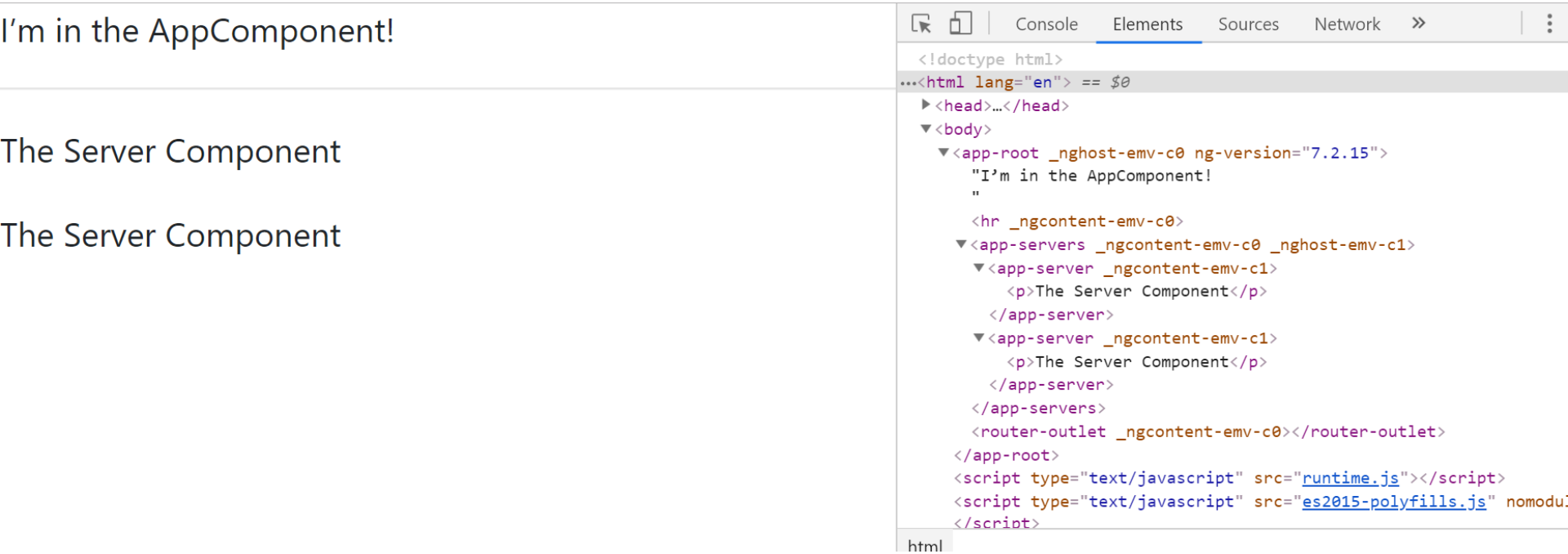
- Instead of using external template file, you can use inline template file.
- Go to servers.component.ts and change templateUrl to “template”.
- Copy C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version4\bootstrap4 to C:\ workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version5\bootstrap4
- Change C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version5\bootstrap4\src\app\servers\servers.component.ts.

Working with Component Templates



Working with Component Templates

- Check in browser still working:



Working with Component Styles

- Copy C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version4\bootstrap4 to C:\ workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version6
- Edit the following:
- C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version6\bootstrap6\src\app\app.component.html

Working with Component Styles

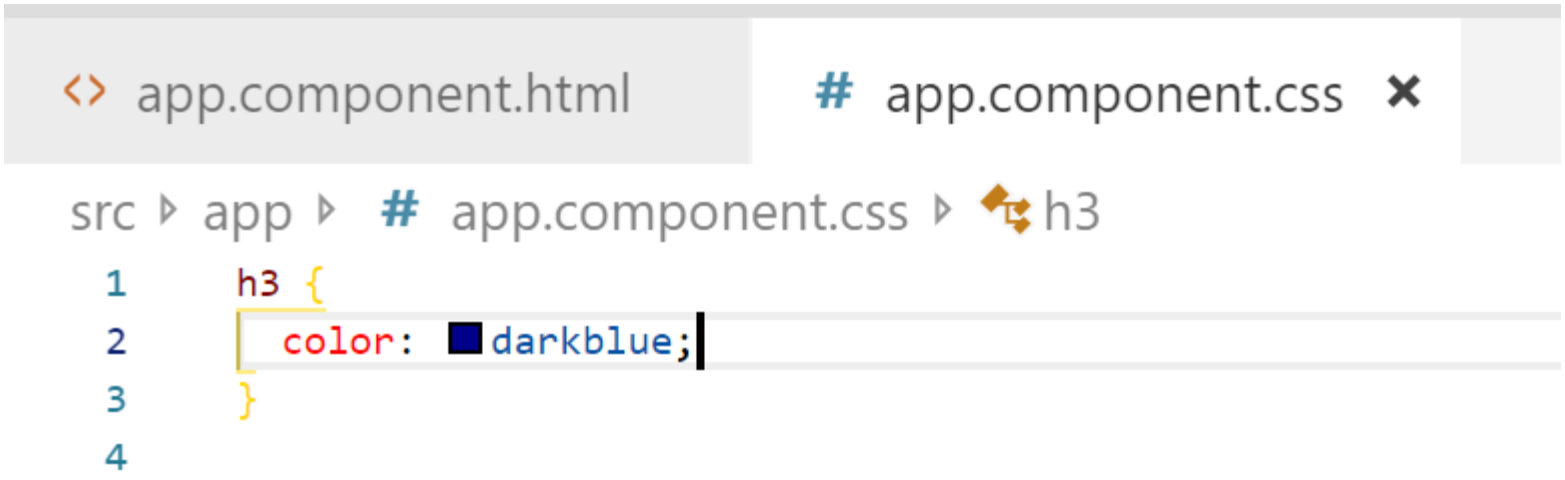
app.component.html

src ▸ app ▸ app.component.html ▸ router-outlet

```
1   <div class="container">
2     <div class="row">
3       <div class="col-xs-12">
4         <h3>I'm in the AppComponent!</h3>
5         <hr>
6         <app-servers></app-servers>
7       </div>
8     </div>
9   </div>
10  <router-outlet></router-outlet>
11
```

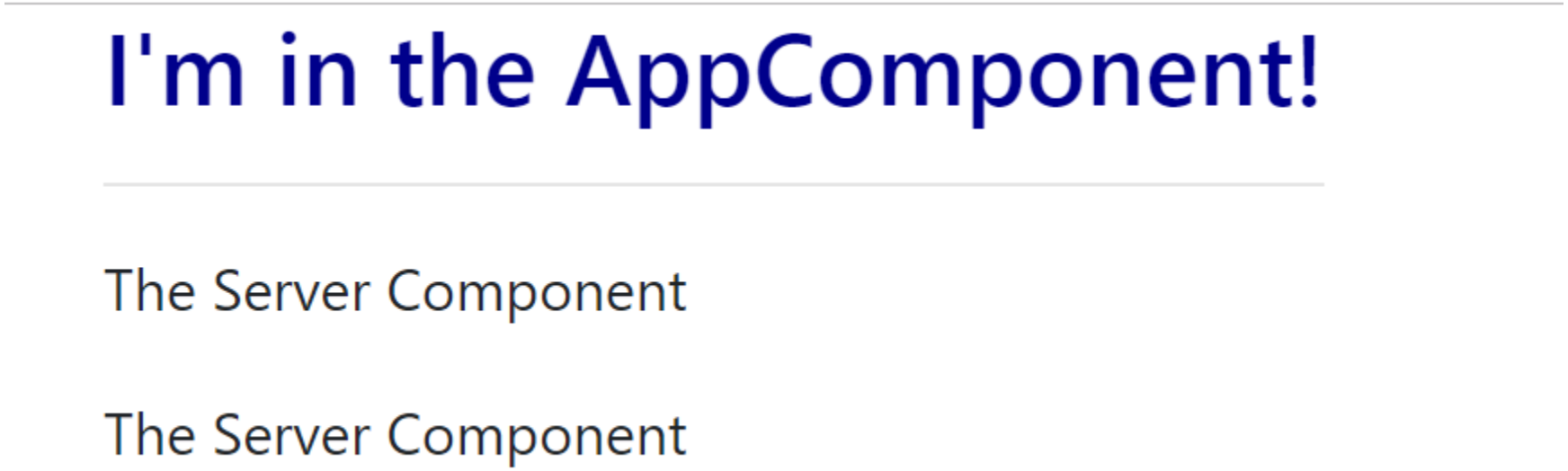
Working with Component Styles

- Edit the following C:\workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version6\bootstrap4\src\app\app.component.css



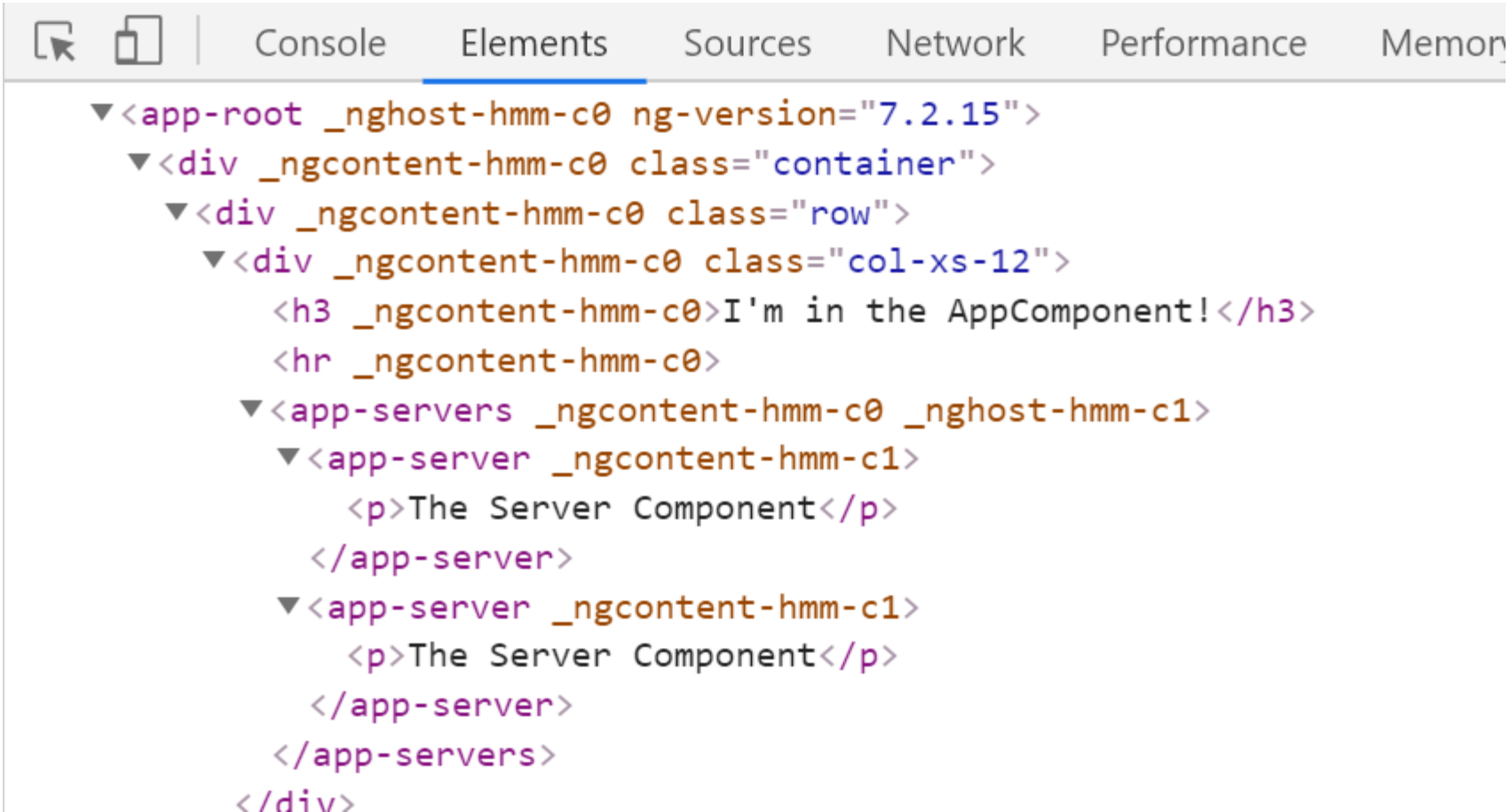
Working with Component Styles

- Edit the following C:\workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version6\bootstrap4\src\app\app.component.css



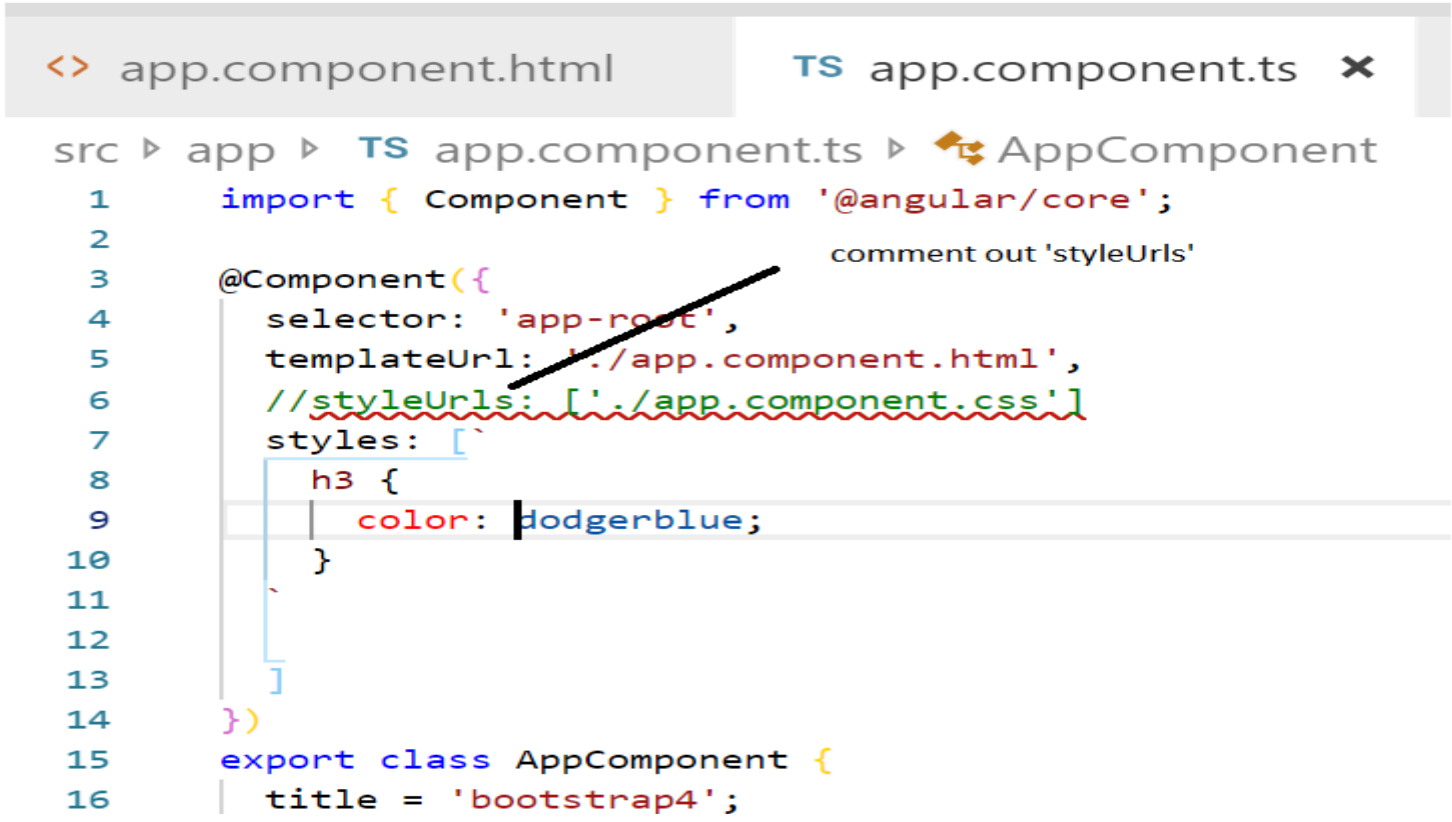
Working with Component Styles

- Edit the following C:\workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version6\bootstrap4\src\app\app.component.css



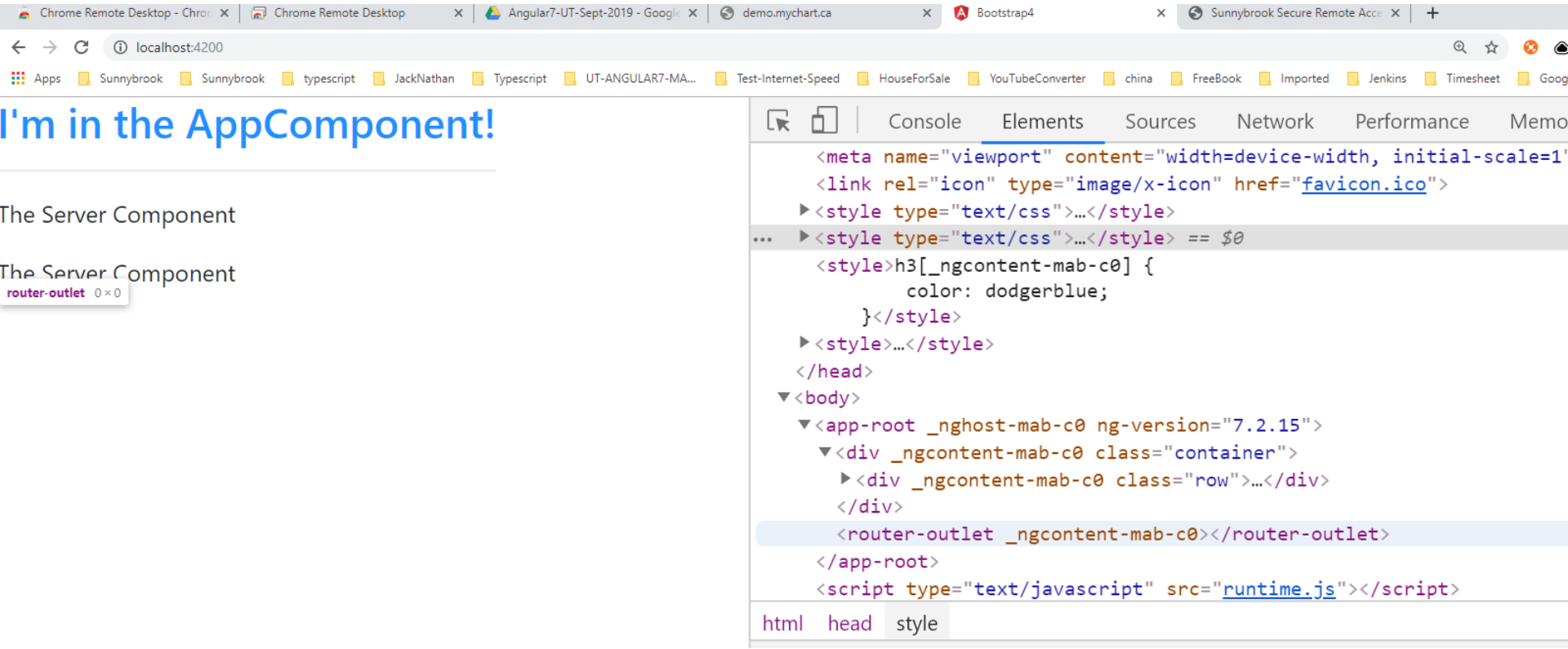
Working with Component Styles

- Edit C:\workspace-UT-Angular7-May-2019\session7\example1-bootstrap4-version6\bootstrap4\src\app\app.component.ts



Working with Component Styles

- Edit C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version6\bootstrap4\src\app\app.component.ts



Fully Understanding The Component Selector

- Copy C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version6\bootstrap4 to C:\ workspace-UT-Angular8-SEPT-2020\session4\example1-bootstrap4-version7\bootstrap4
- Edit C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version7\bootstrap4\src\app\servers\servers.component.ts

Fully Understanding The Component Selector

TS servers.component.ts × <> app.component.html

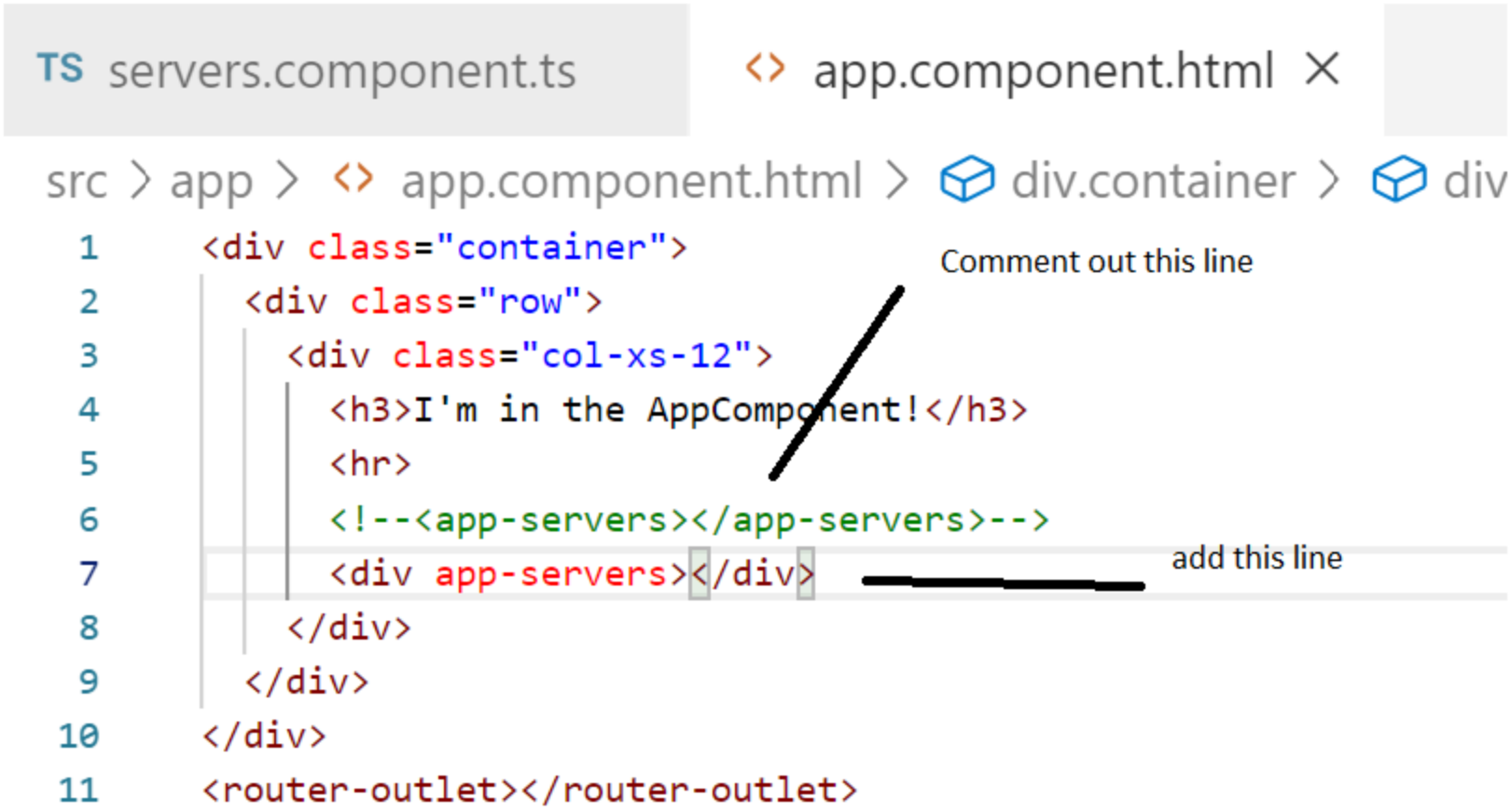
src > app > servers > TS servers.component.ts > ServersComponent

```
1   import { Component, OnInit } from '@angular/core';
2
3   @Component({
4     selector: 'app-servers',
5     selector: '[app-servers]',
6     templateUrl: './servers.component.html',
7     styleUrls: ['./servers.component.css']
8   })
9   export class ServersComponent implements OnInit {
10
11    constructor() { }
12
13    ngOnInit() {
14    }
15
16  }
17
```

Comment this out and add this line '[app-servers]'

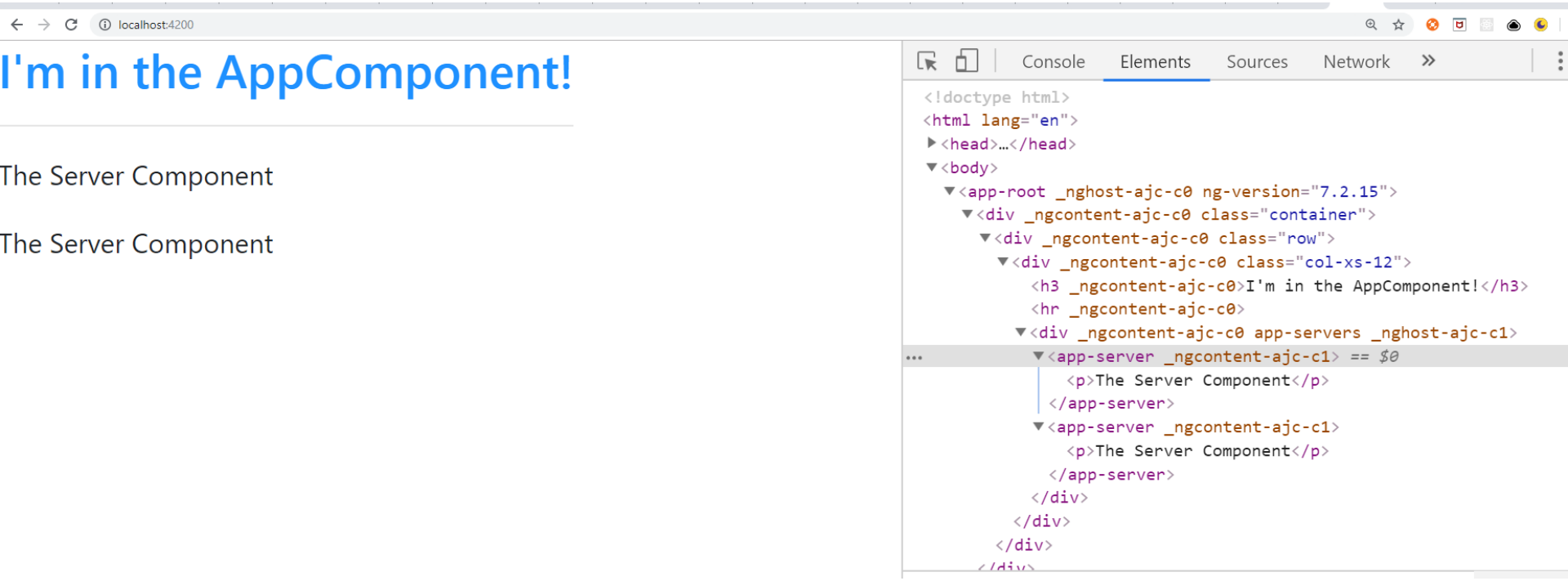
Fully Understanding The Component Selector

- Edit C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version7\bootstrap4\src\app\app.component.html



Fully Understanding The Component Selector

- Check browser:

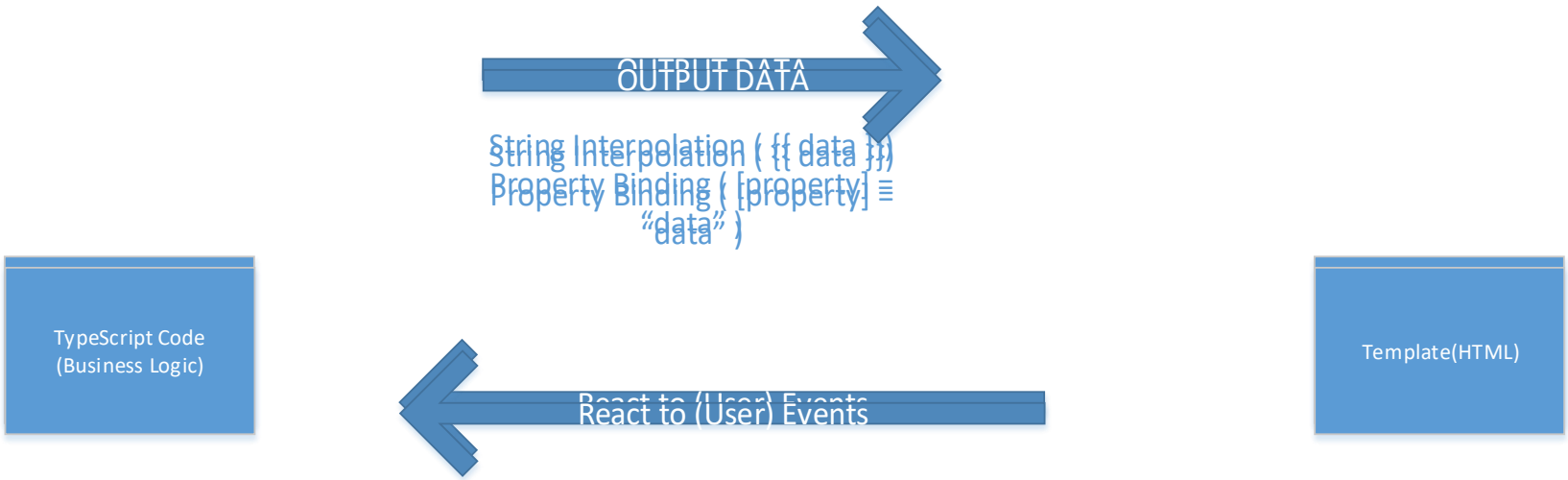


What is databinding

- Copy C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version4\bootstrap4 to C:\ workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version8
- What is databinding?

What is databinding

Databinding ≡ Communication



Combination of Both: Two-Way-Binding([(ngModel)] ≡ 'data')

What is databinding

- C:\ workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version8\bootstrap4\src\app\server\server.component.ts

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

@Component({
  selector: 'app-server',
  templateUrl: './server.component.html'
})
export class ServerComponent {
    serverId: number = 10;
    serverStatus: string = 'offline';

    getServerStatus(): string {
      return this.serverStatus;
    }
}
```

What is databinding

- C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version8\bootstrap4\src\app\server\server.component.html

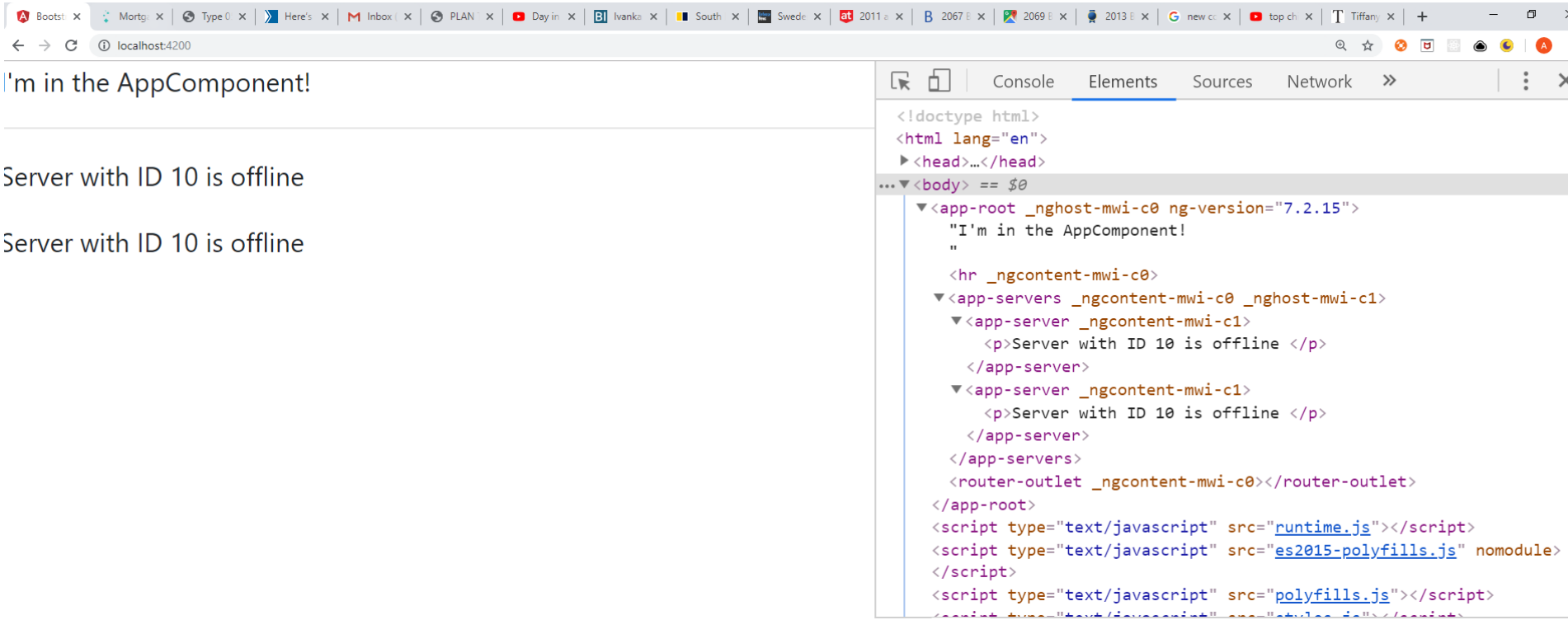
server.component.html ✖ TS server.component.ts

src ▸ app ▸ server ▸ server.component.html ▸ p

```
1   <p>{{'Server'}} with ID {{ serverId }} is {{ getServerStatus()  }} </p>
2
```

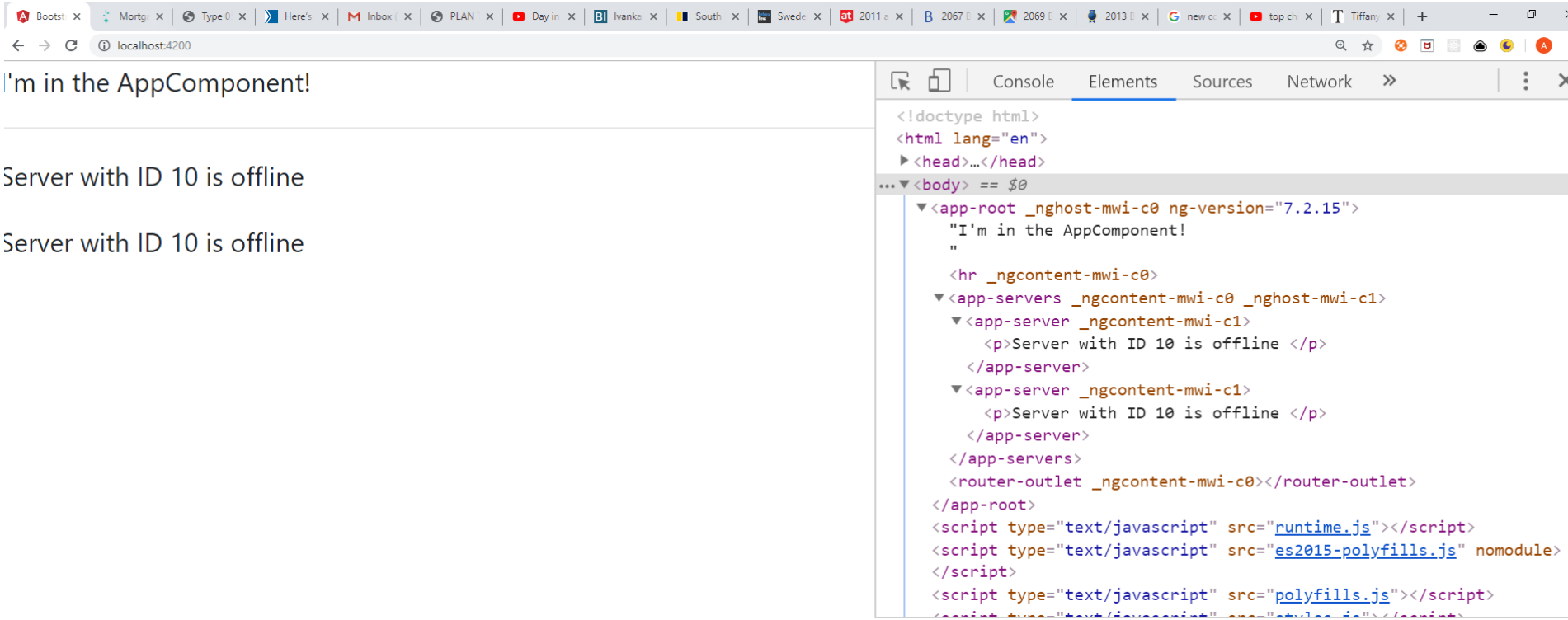
What is databinding

- C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version8\bootstrap4\src\app\server\server.component.html



What is databinding

- C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version8\bootstrap4\src\app\server\server.component.html



Property Binding

- Copy C:\workspace-UT-Angular8-SEPT-2020 \session4\example1-bootstrap4-version8\bootstrap4 to C:\ workspace-UT-Angular8-JAN-2020\session4\example2-propertybinding-version1
- Edit C:\workspace-UT-Angular8-SEPT-2020 \session4\example2-propertybinding-version1\bootstrap4\src\app\servers\servers.component.ts

Property Binding

servers.component.html | TS servers.component.ts | # servers.co

src ▸ app ▸ servers ▸ TS servers.component.ts ▸ ServersComponent ▸

```
1   import { Component, OnInit } from '@angular/core';
2
3   @Component({
4     selector: 'app-servers',
5     //template: '<app-server></app-server><app-server></app-server>',
6     templateUrl: './servers.component.html',
7     styleUrls: ['./servers.component.css']
8   })
9   export class ServersComponent implements OnInit {
10    allowNewServer = false;
11
12    constructor() {
13      setTimeout(() => {
14        this.allowNewServer = true;
15      }, 2000);
16    }
17
18    ngOnInit() {
19    }
20
21  }
```

Add this new line of code and constructor

Property Binding

- C:\workspace-UT-Angular8-JAN-2020 \session4\example2-propertybinding-version1\bootstrap4\src\app\servers\servers.component.html

"[]" indicate to Angular we're using property binding.

We want to dynamically bind some property.

Property Binding

- The button is enabled after 2 seconds:

```
▼<body>
  ▼<app-root _nghost-hqk-c0 ng-version="7.2.15">
     "I'm in the AppComponent!
     "
     <hr _ngcontent-hqk-c0>
    ▼<app-servers _ngcontent-hqk-c0 _nghost-hqk-c1>
       <button _ngcontent-hqk-c1 class="btn btn-primary" disabled>Add Server</button> ==
      ▶<app-server _ngcontent-hqk-c1>…</app-server>
      ▶<app-server _ngcontent-hqk-c1>…</app-server>
     </app-servers>
     <router-outlet _ngcontent-hqk-c0></router-outlet>
  </app-root>
```

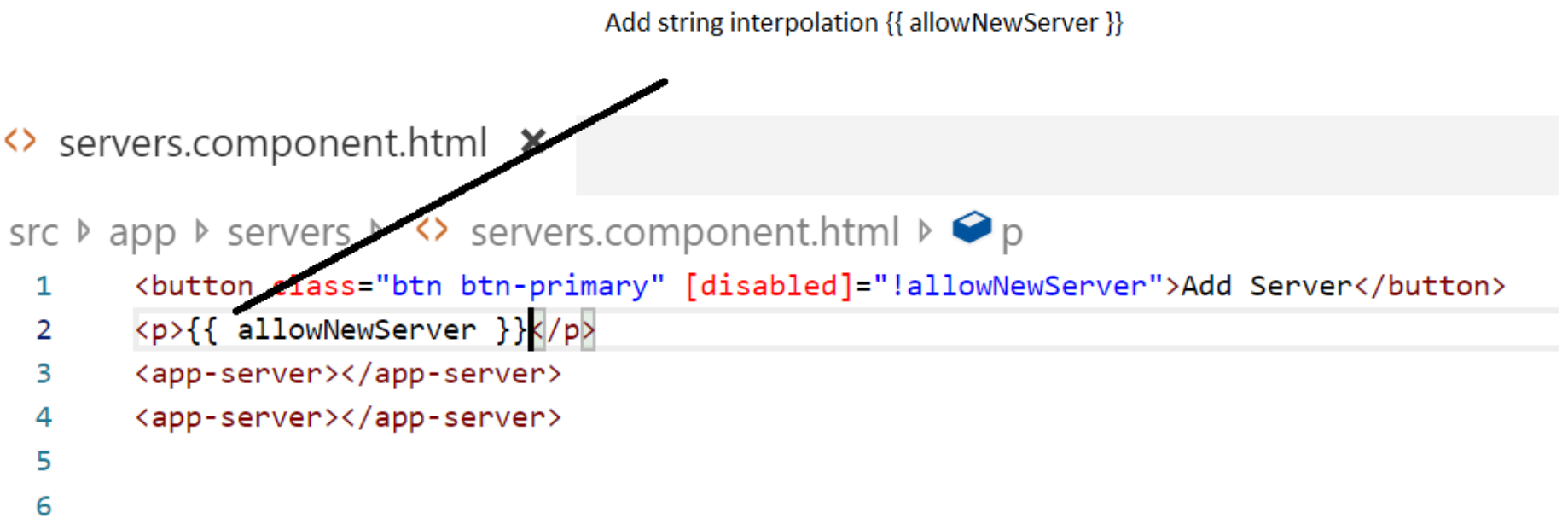
word "disabled" added at the beginning of load and than disappear

Property Binding vs String Interpolation

- Copy C:\workspace-UT-Angular8-JAN-2020 \session4\example2-propertybinding-version1\bootstrap4 to C:\workspace-UT-Angular8-JAN-2020 \session4\example3-propertybinding-vs-stringInterpolation
- Edit C:\workspace-UT-Angular8-JAN-2020 \session4\example3-propertybinding-vs-stringInterpolation\bootstrap4\src\app\servers\servers.component.html

Property Binding vs String Interpolation

- Edit C:\workspace-UT-Angular8-JAN-2020 \session4\example3-propertybinding-vs-stringInterpolation\bootstrap4\src\app\servers\servers.component.html



Property Binding vs String Interpolation

- Edit C:\ workspace-UT-Angular8-JAN-2020 \session4\example3-propertybinding-vs-stringInterpolation\bootstrap4\src\app\servers\servers.component.html

You could simply bind to a property of this element the inner Text property and set this equal to allow a news server.

The inner text property of an elemnt is just what's between the opening and closing tag. So in this case we were able to easily replace string interpolation with property binding.

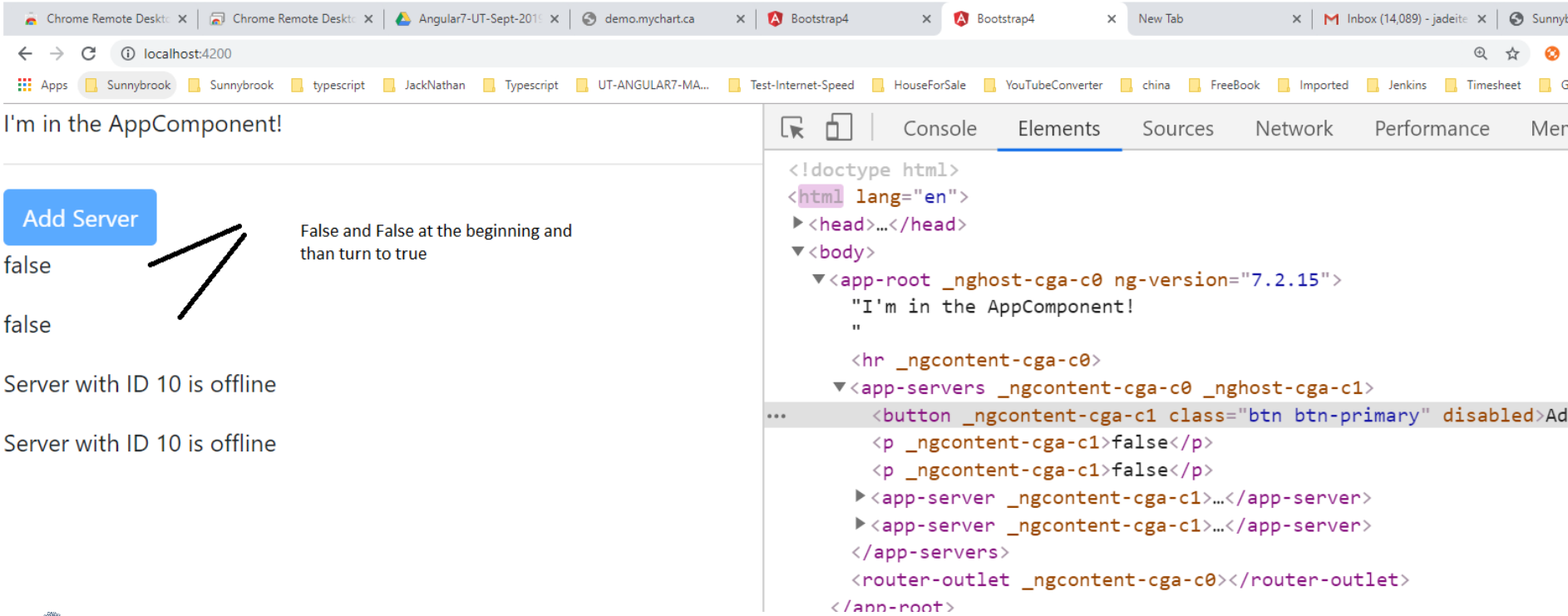
servers.component.html

src ▸ app ▸ servers ▸ servers.component.html ▸ ...

```
1   <button class="btn btn-primary" [disabled]="!allowNewServer">Add Server</button>
2   <p>{{ allowNewServer }}</p>
3   <p [innerText]="allowNewServer"></p>
4   <app-server></app-server>
5   <app-server></app-server>
6
7
```

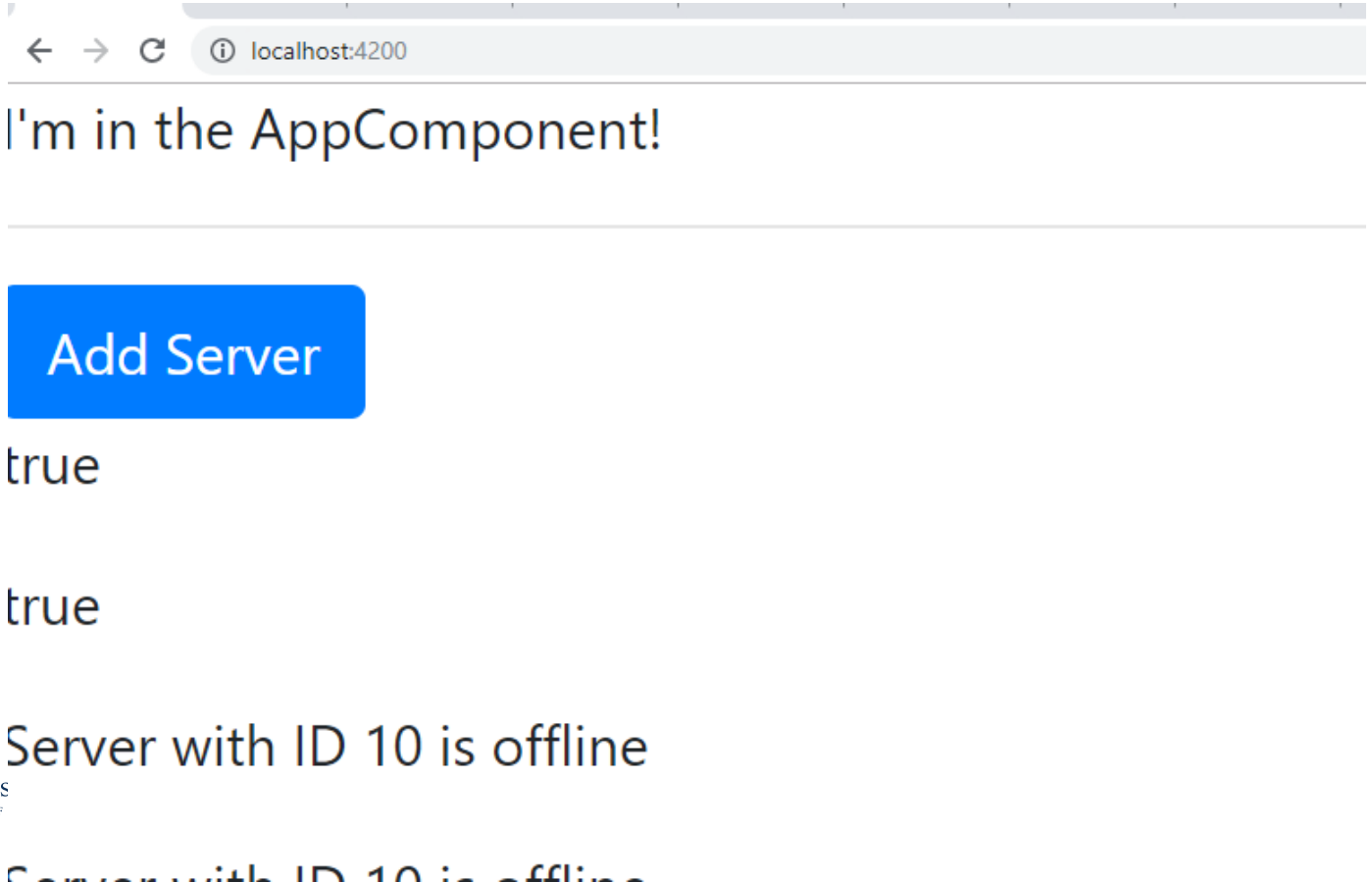
Property Binding vs String Interpolation

- Edit C:\workspace-UT-Angular8-JAN-2020 \session7\example3-propertybinding-vs-stringInterpolation\bootstrap4\src\app\servers\servers.component.html



Property Binding vs String Interpolation

- Edit C:\workspace-UT-Angular8-JAN-2020 \session7\example3-propertybinding-vs-stringInterpolation\bootstrap4\src\app\servers\servers.component.html



Property Binding vs String Interpolation

- So when should you use which of the two?
- Well basically if you want to output something in your template print some text to it using string interpolation.
- If you want to change some property be that of an html element you will typically use property binding.

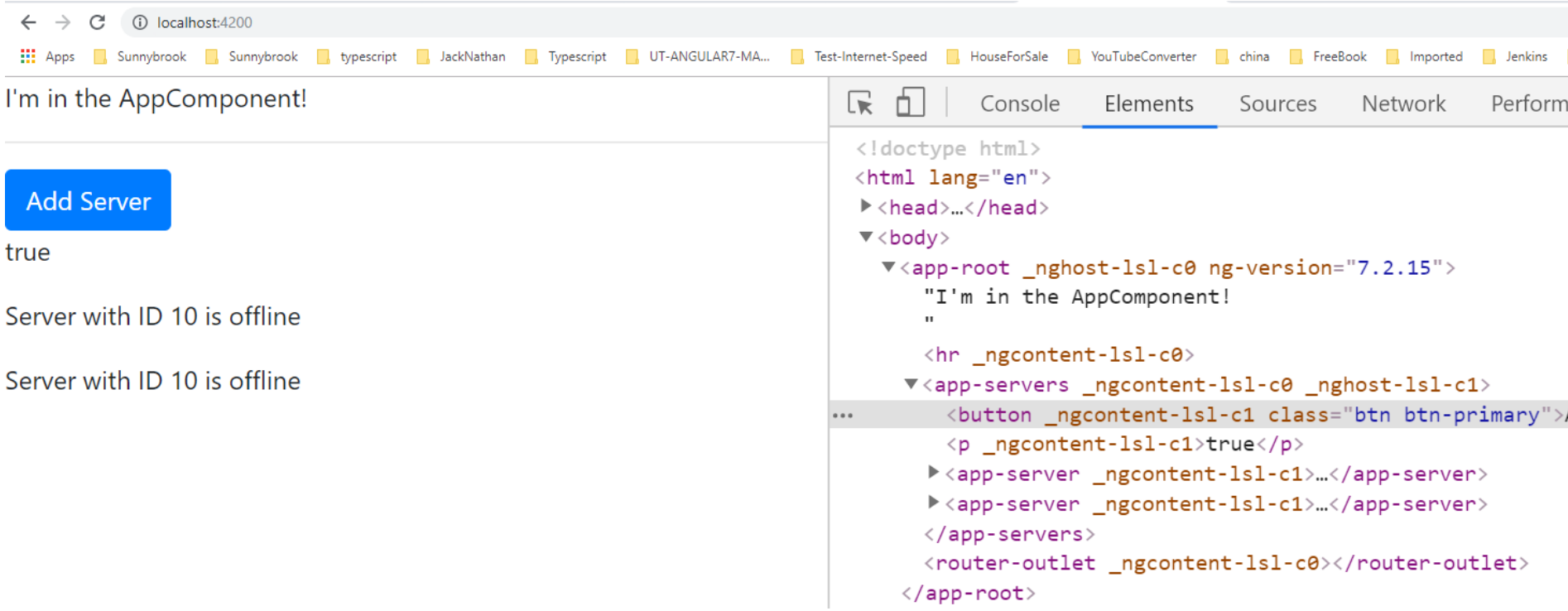
servers.component.html

src ▸ app ▸ servers ▸ servers.component.html ▸ ...

```
1    <button class="btn btn-primary" [disabled]="!allowNewServer">Add Server</button>
2    <!--<p>{{ allowNewServer }}</p>-->
3    <p [innerText]="allowNewServer"></p>
4    <app-server></app-server>
5    <app-server></app-server>
6
7
```

Property Binding vs String Interpolation

- So when should you use which of the two?
- Well basically if you want to output something in your template print some text to it using string interpolation.
- If you want to change some property be that of an html element you will typically use property binding.



Eventbinding

- Copy C:\workspace-UT-Angular8-JAN-2020 \session4\example3-propertybinding-vs-stringInterpolation\bootstrap4 to C:\workspace-UT-Angular8-JAN-2020\session4\example4-eventbinding\bootstrap4\src\app\servers\servers.component.ts

Eventbinding

servers.component.ts

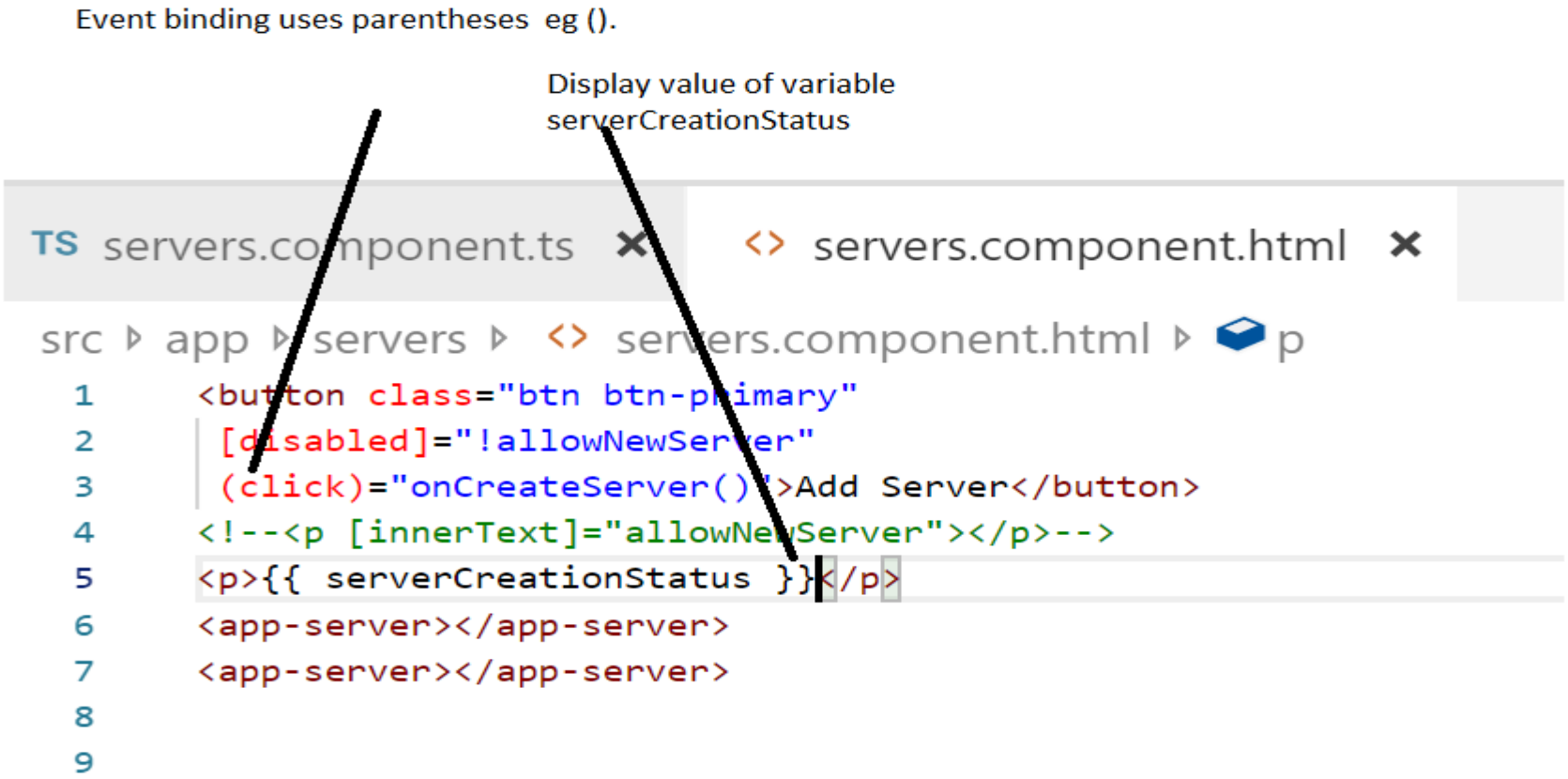
src ▸ app ▸ servers ▸ TS servers.component.ts ▸ ServersComponent ▸

```ts
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-servers',
  templateUrl: './servers.component.html',
  styleUrls: ['./servers.component.css']
})
export class ServersComponent implements OnInit {
  allowNewServer = false;
  serverCreationStatus = 'No server was created!';
  constructor() {
    setTimeout(() => {
      this.allowNewServer = true;
    }, 2000);
  }

  ngOnInit() {
  }

  onCreateServer() {
    this.serverCreationStatus = 'Server was created!';
  }

```

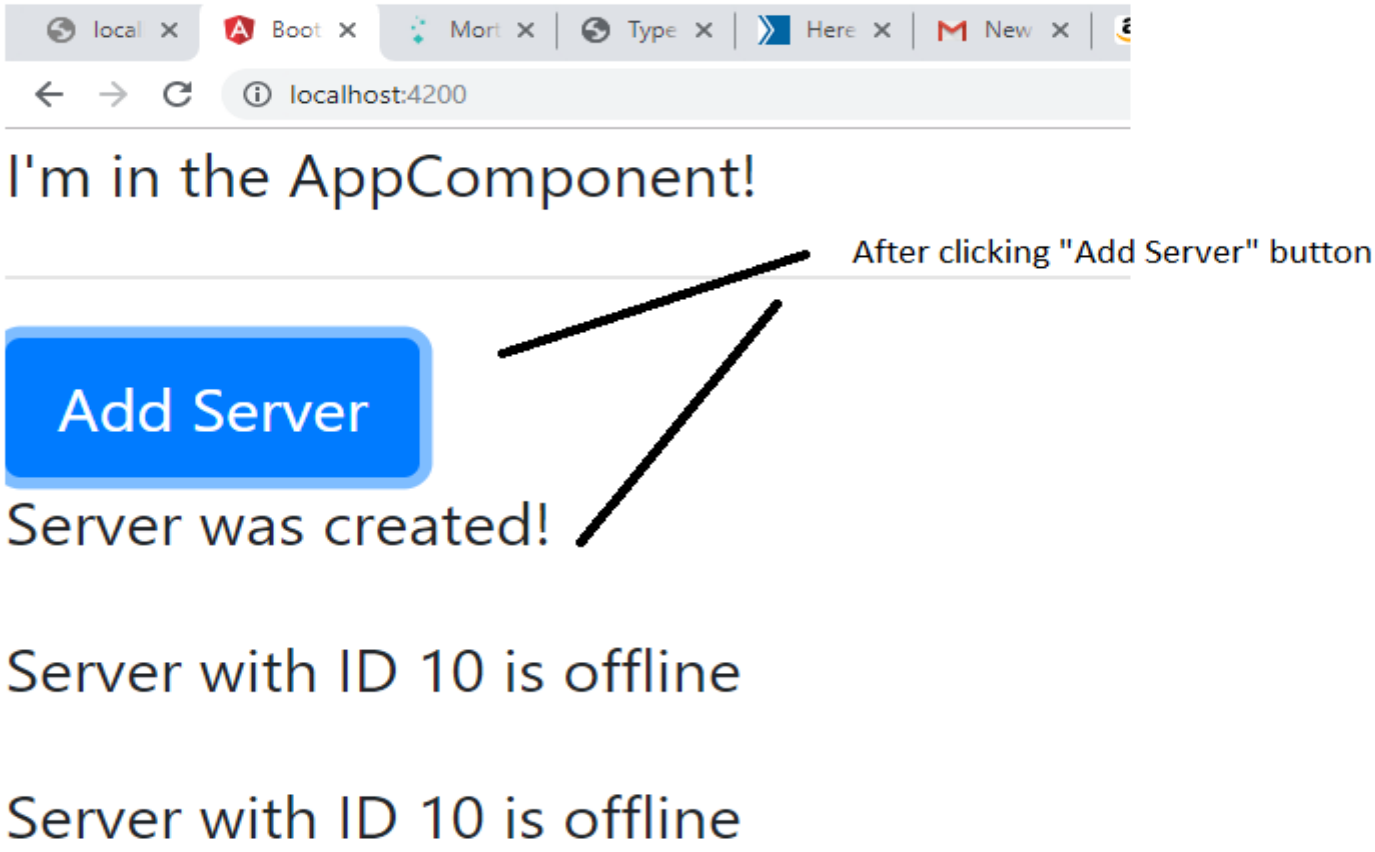
Add these line of code

Eventbinding

- Edit C:\workspace-UT-Angular8-JAN-2020 \session7\example4-eventbinding\bootstrap4\src\app\servers\servers.component .html



Eventbinding



Passing and Using Data with Event Binding(example5-PassingAndUsingDataWithEventBinding)

- Copy C:\workspace-UT-Angular8-JAN-2020\session4\example4-eventbinding\bootstrap4 to C:\workspace-UT-Angular8-JAN-2020\session4\example5-PassingAndUsingDataWithEventBinding
- Edit C:\workspace-UT-Angular8-JAN-2020\session4\example5-PassingAndUsingDataWithEventBinding\bootstrap4\src\app\servers\servers.component.html

Passing and Using Data with Event Binding(example5-PassingAndUsingDataWithEventBinding)

input and click are default events provided by the dom. Input event gives us some data about the event. We can capture this data if $event passed as an argument to the method we're calling.

servers.component.html

src ▸ app ▸ servers ▸ servers.component.html ▸ ...

```
1   <label>Server Name</label>
2   <input type="text" class="form-control" (input)="onUpdateServerName($event)">
3   <button class="btn btn-primary"
4     [disabled]="!allowNewServer"
5     (click)="onCreateServer()">Add Server</button>
6   <!--<p [innerText]="allowNewServer"></p>-->
7   <p>{{ serverCreationStatus }}</p>
8   <app-server></app-server>
9   <app-server></app-server>
10
```

Passing and Using Data with Event Binding(example5-PassingAndUsingDataWithEventBinding)

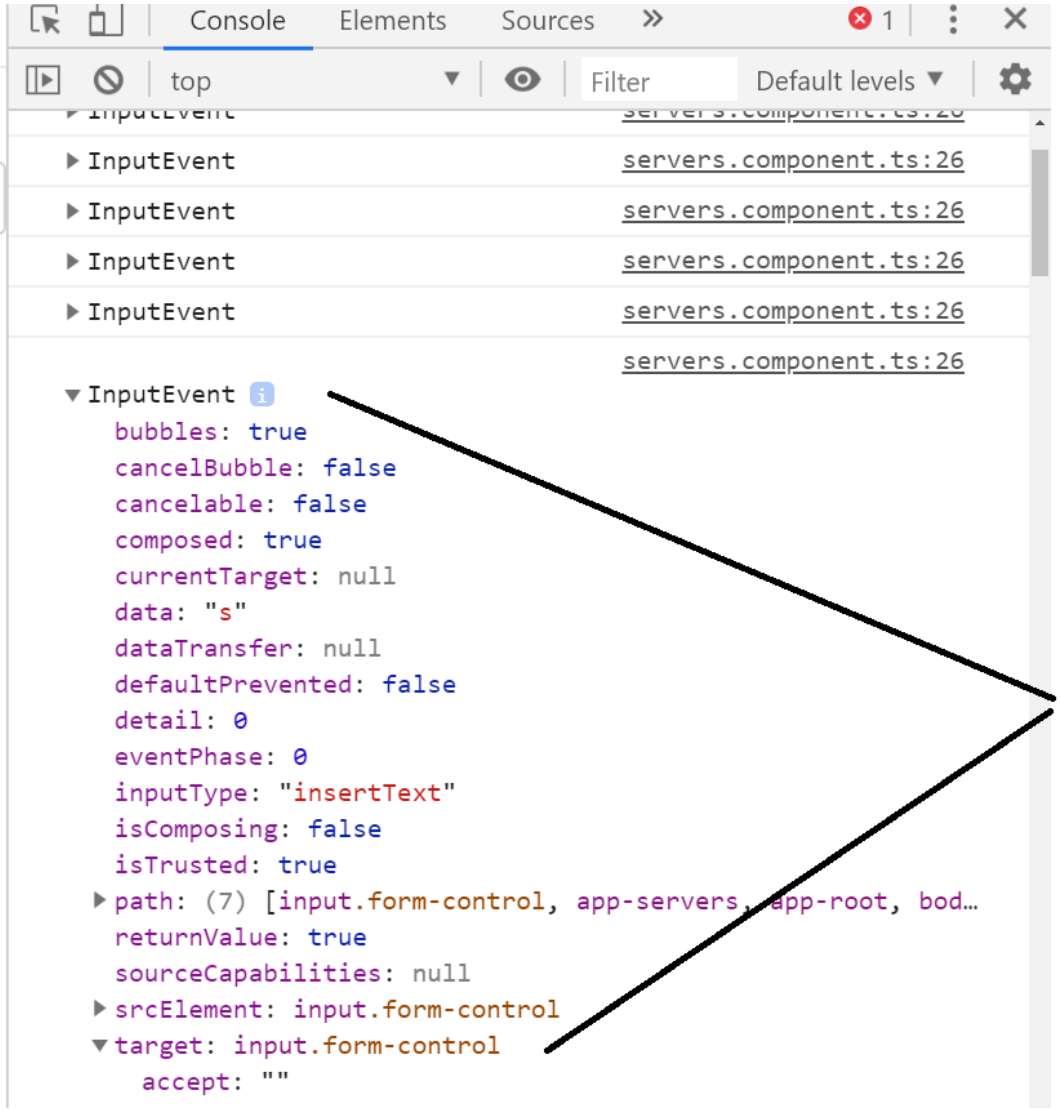
- Edit C:\workspace-UT-Angular8-JAN-2020 \session7\example5-PassingAndUsingDataWithEventBinding\bootstrap4\src\app \servers\servers.component.ts

```
onUpdateServerName(event: any) {
  console.log(event);
}
```

Passing and Using Data with Event Binding

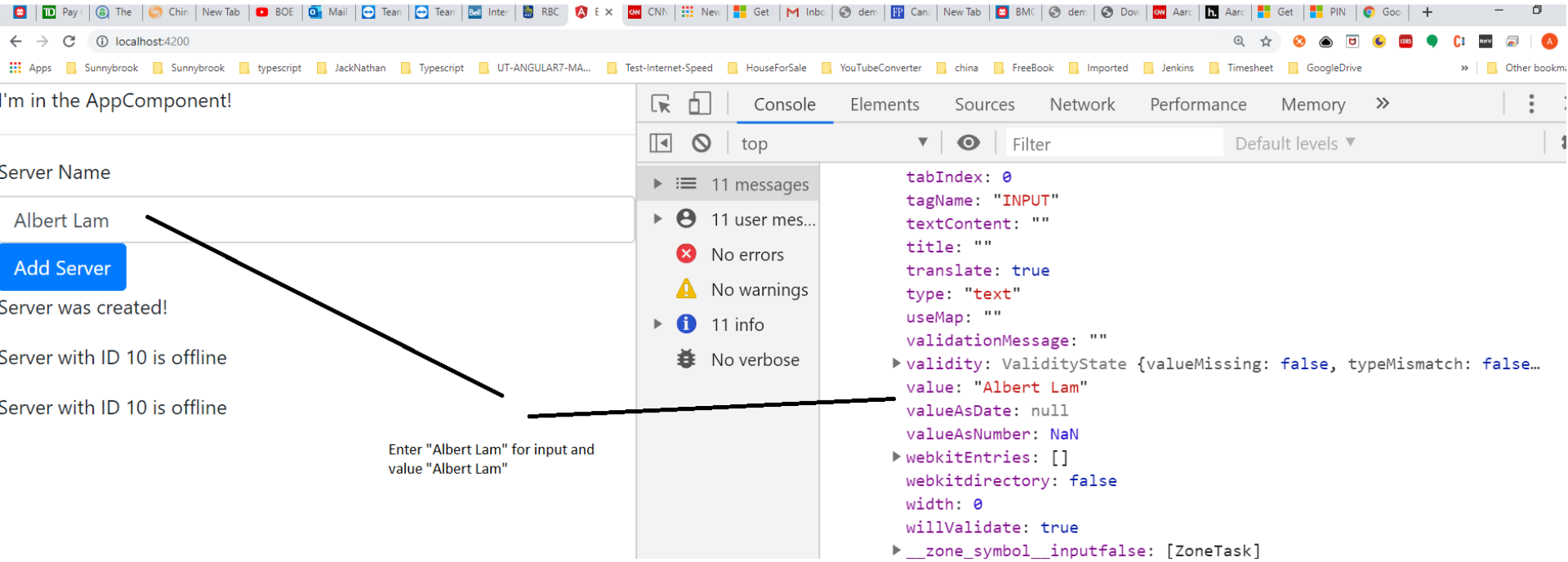
Passing and Using Data with Event Binding

Passing and Using Data with Event Binding



event.target.value

Passing and Using Data with Event Binding



example6-PassingAndUsingDataWithEventBinding

- Copy C:\workspace-UT-Angular8-JAN-2020 \session4\example5-PassingAndUsingDataWithEventBinding\bootstrap4 to C:\ workspace-UT-Angular8-JAN-2020 \session4\example6-PassingAndUsingDataWithEventBinding
- Edit C:\ workspace-UT-Angular8-JAN-2020 \session4\example6-PassingAndUsingDataWithEventBinding\bootstrap4\src\app \servers\servers.component.ts

example6-PassingAndUsingDataWithEventBinding

TS servers.component.ts ✖ <> servers.component.html

src ▸ app ▸ servers ▸ TS servers.component.ts ▸ ServersComponent ▸

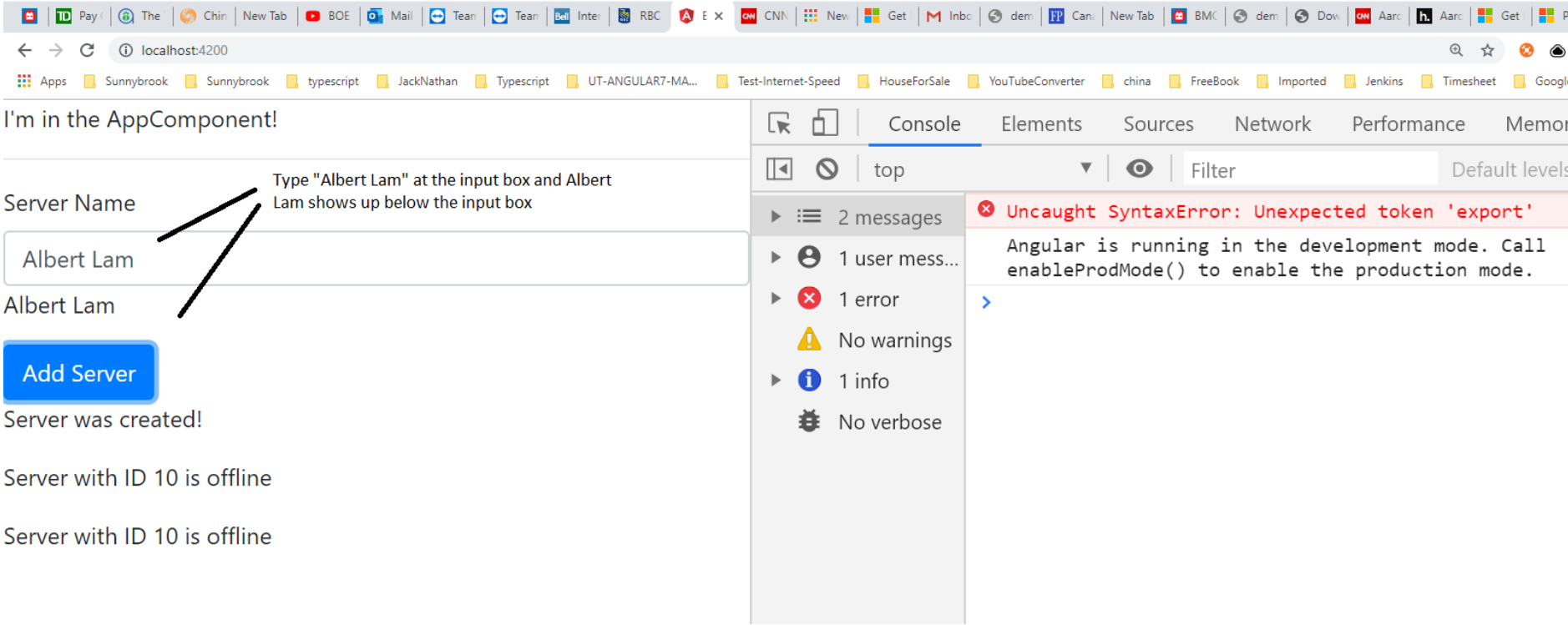
```
7   })
8   export class ServersComponent implements OnInit {
9     allowNewServer = false;
10    serverCreationStatus = 'No server was created!';
11    serverName = '';
12    constructor() {
13      setTimeout(() => {
14        this.allowNewServer = true;
15      }, 2000);
16    }
17
18    ngOnInit() {
19    }
20
21    onCreateServer() {
22      this.serverCreationStatus = 'Server was created!';
23    }
24    /*
25    onUpdateServerName(event: any) {
26      console.log(event);
27    }
28    */
29    onUpdateServerName(event: Event) {
30      this.serverName = (<HTMLInputElement>event.target).value;
31    }
```

Add new variable "serverName" and method "onUpdateServerName.

example6-PassingAndUsingDataWithEventBinding

- Edit C:\ workspace-UT-Angular8-JAN-2020 \session4\example6-PassingAndUsingDataWithEventBinding\bootstrap4\src\app \servers\servers.component.html

example6-PassingAndUsingDataWithEventBinding



example7-TwoWayDatabinding

- Copy C:\workspace-UT-Angular8-JAN-2020 \session7\example6-PassingAndUsingDataWithEventBinding\bootstrap4 to C:\ workspace-UT-Angular8-JAN-2020 \session4\example7-TwoWayDatabinding
- With two way databinding we combine property and event binding.
- Edit C:\workspace-UT-Angular8-JAN-2020 \session4\example7-TwoWayDatabinding\bootstrap4\src\app\servers\servers.component.html

example7-TwoWayDatabinding

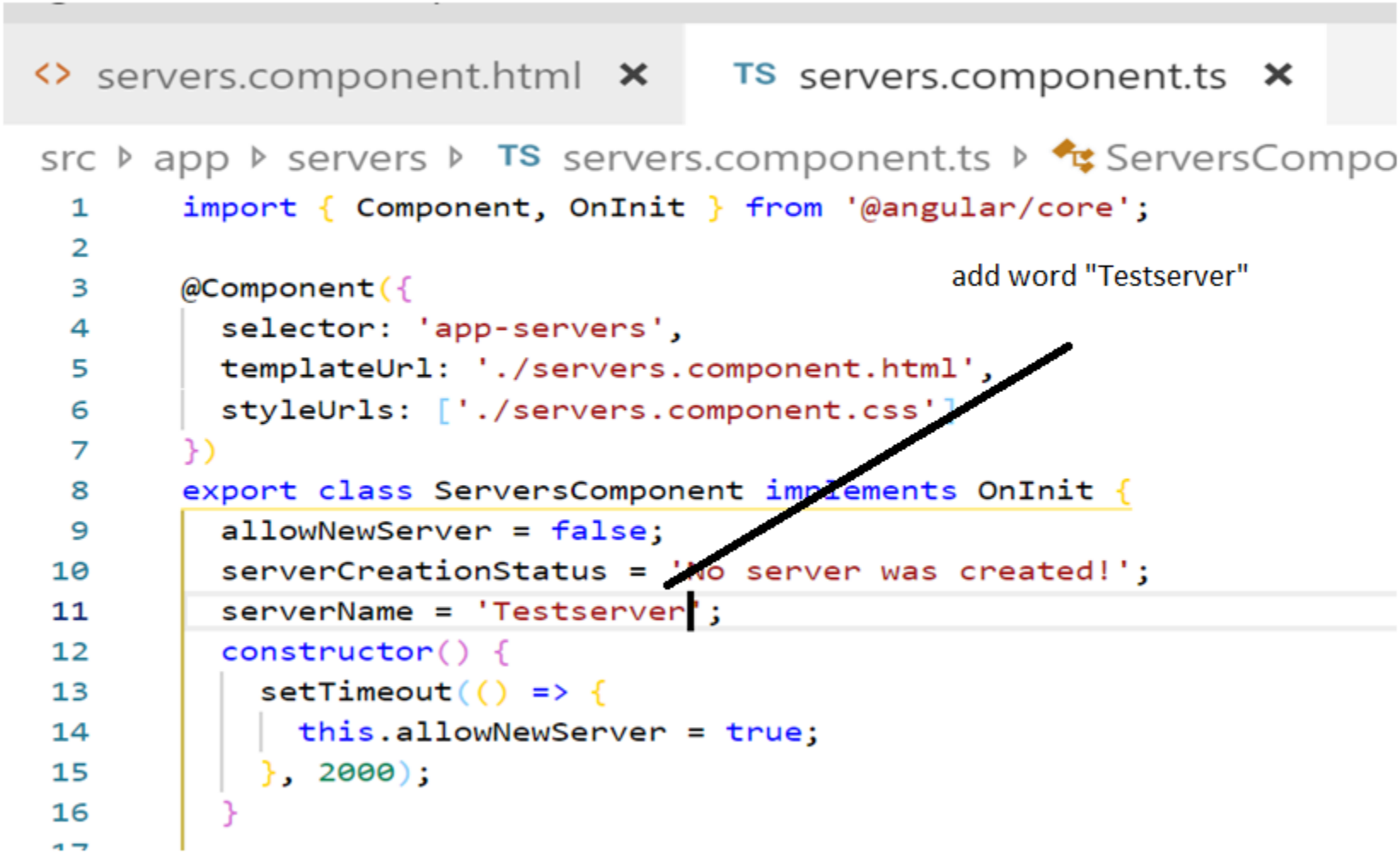
servers.component.html

src ▸ app ▸ servers ▸ servers.component.html ▸ ...

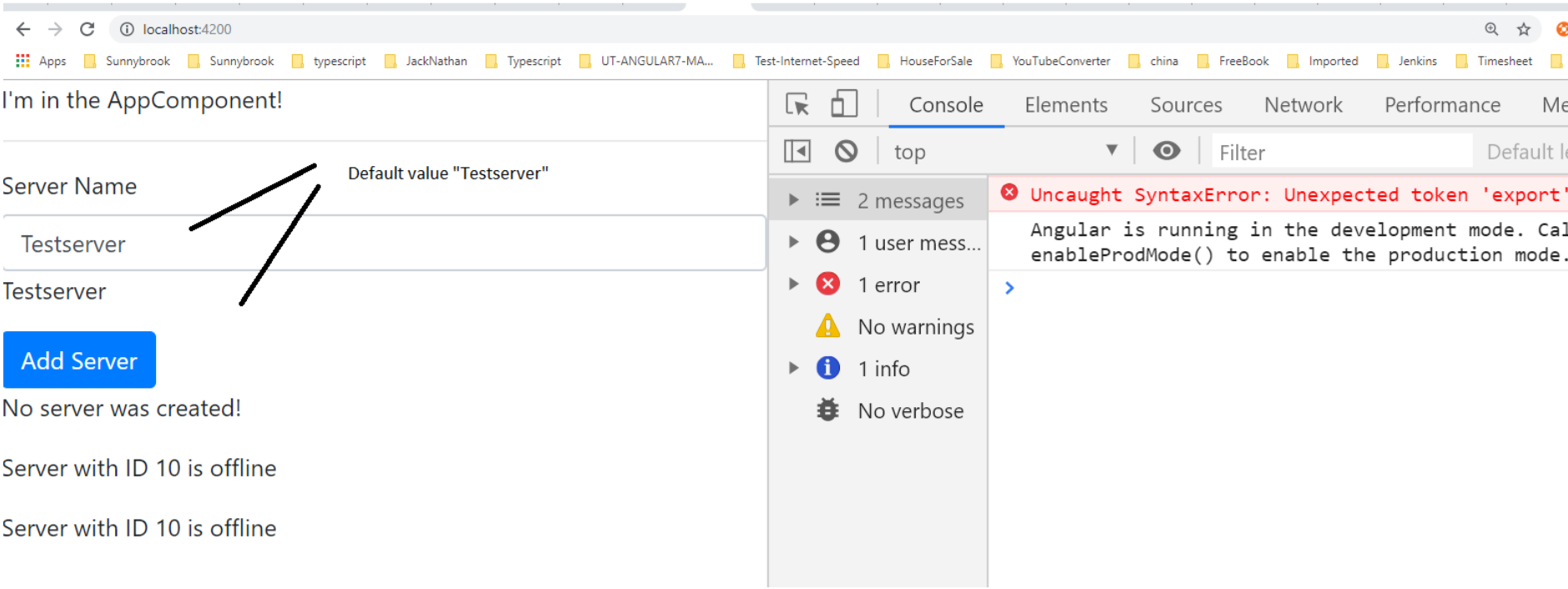
```
1   <label>Server Name</label>
2
3   <!--<input type="text" class="form-control" (input)="onUpdateServerName($event)">-->
4   <input type="text" class="form-control" [(ngModel)]="serverName">
5   <p>{{ serverName }} </p>
6   <button class="btn btn-primary"
7    [disabled]="!allowNewServer"
8    (click)="onCreateServer()">Add Server</button>
9   <!--<p [innerText]="allowNewServer"></p>-->
10  <p>{{ serverCreationStatus }}</p>
11  <app-server></app-server>
12  <app-server></app-server>
13
14
```

Add this line of code

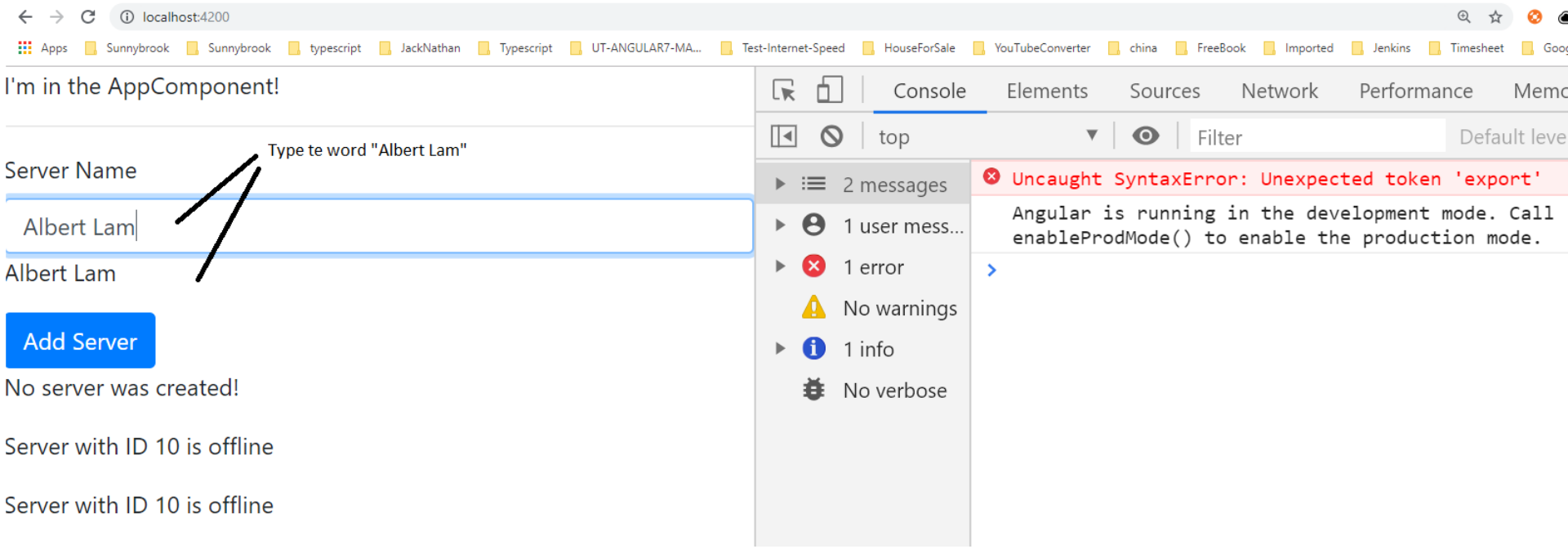
example7-TwoWayDatabinding



example7-TwoWayDatabinding



example7-TwoWayDatabinding



example8-CombiningAllFormsOfDataBinding

- Copy C:\workspace-UT-Angular8-JAN-2020 \session4\example7-TwoWayDatabinding\bootstrap4 to C:\ workspace-UT-Angular8-JAN-2020\session7\example8-CombiningAllFormsOfDataBinding
- Edit servers.component.html

Comment out "serverName"

servers.component.html

src ▸ app ▸ servers ▸ servers.component.html ▸ ...

```
1    <label>Server Name</label>
2
3    <!--<input type="text" class="form-control" (input)="onUpdateServerName($event)">-->
4    <input type="text" class="form-control" [(ngModel)]="serverName">
5    <!----<p>{{ serverName }} </p>-->
6    <button class="btn btn-primary"
7     [disabled]="!allowNewServer"
8     (click)="onCreateServer()">Add Server</button>
```

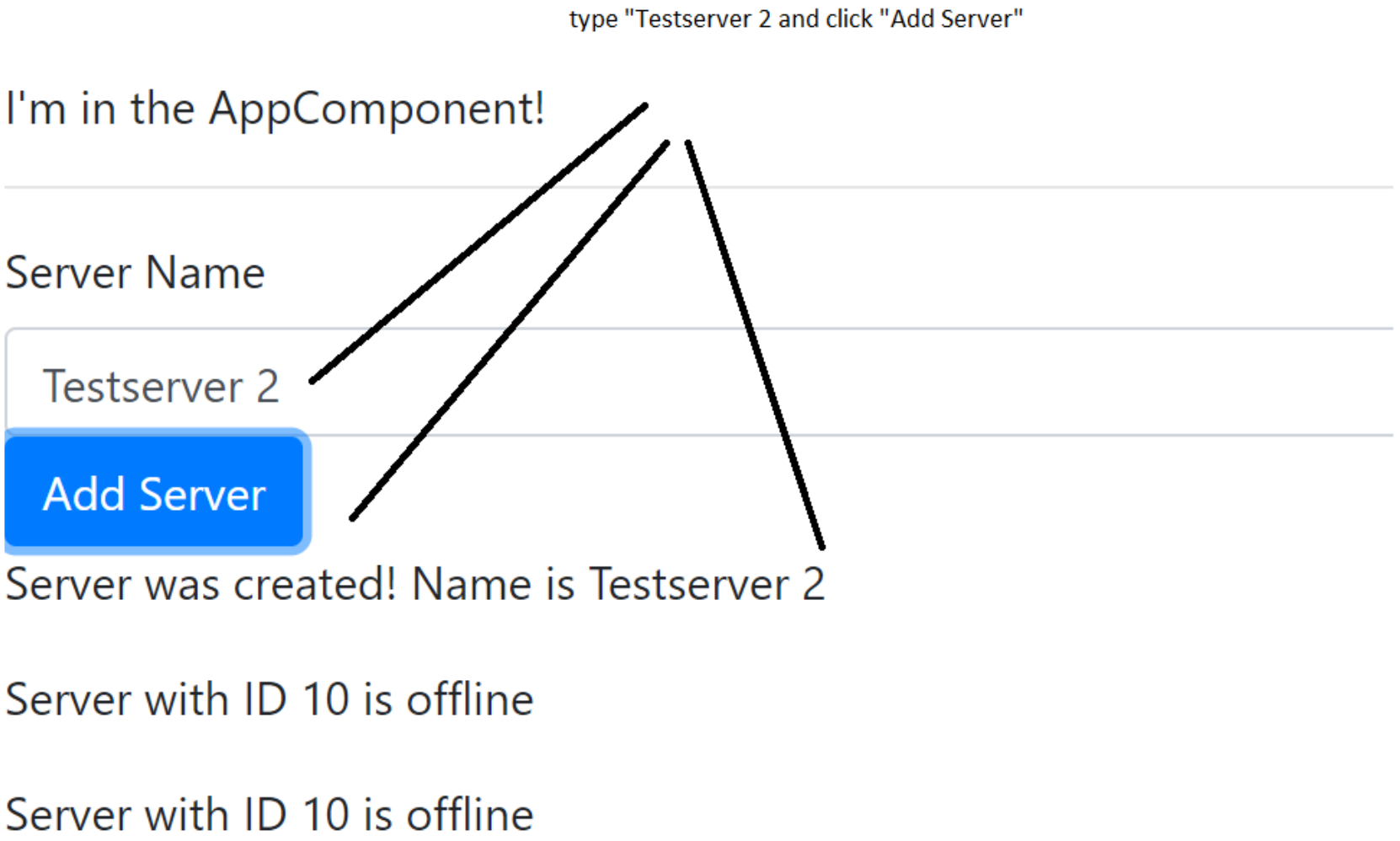
example8-CombiningAllFormsOfDataBinding

- Edit C:\workspace-UT-Angular8-JAN-2020\session4\example8-CombiningAllFormsOfDataBinding\bootstrap4\src\app\servers\servers.component.ts

Add "this.serverName;"

```
20
21    onCreateServer() {
22      this.serverCreationStatus = 'Server was created! Name is ' + this.serverName;
23    }
```

example8-CombiningAllFormsOfDataBinding

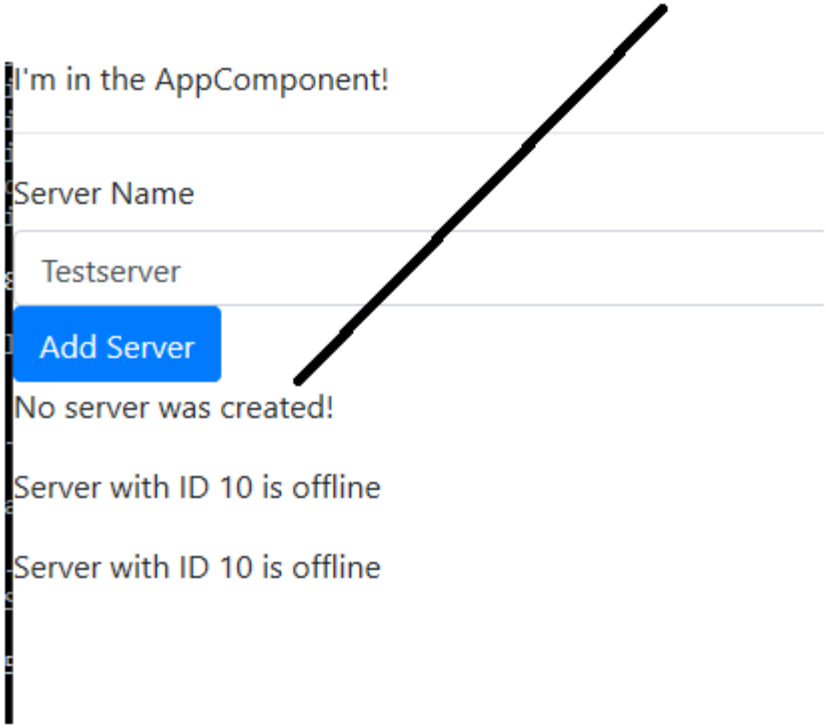


Directives(example9-UsingNgIfToOutputData)

- What are Directives?
- Directives are Instructions in the DOM!
- Components are directives but directives with a template.
- *ngIf is a structural directives which add or remove elements.
- Copy C:\workspace-UT-Angular8-JAN-2020\session4\example8-CombiningAllFormsOfDataBinding\bootstrap4 to C:\workspace-UT-Angular8-JAN-2020\session4\example9-UsingNgIfToOutputData

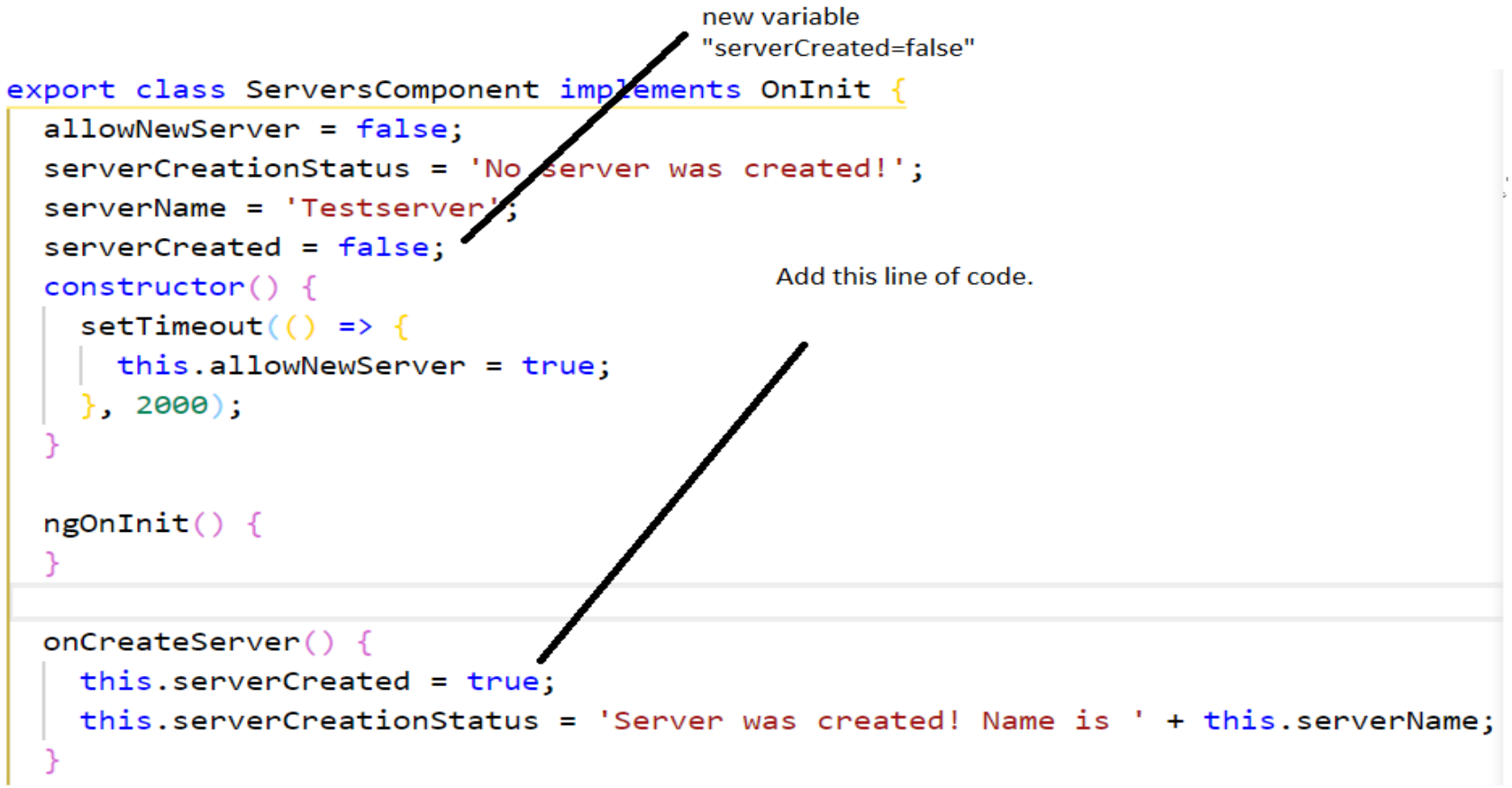
Directives(example9-UsingNgIfToOutputData)

Currently "No server was created!" display at the beginning. We need to conditionally show this message "No server was created!".



Directives(example9-UsingNgIfToOutputData)

- Edit C:\workspace-UT-Angular8-JAN-2020\session4\example9-UsingNgIfToOutputData\bootstrap4\src\app\servers\servers.component.ts



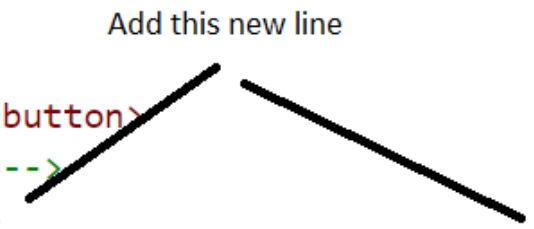

```
export class ServersComponent implements OnInit {
  allowNewServer = false;
  serverCreationStatus = 'No server was created!';
  serverName = 'Testserver';
  serverCreated = false;
  constructor() {
    setTimeout(() => {
      this.allowNewServer = true;
    }, 2000);
  }

  ngOnInit() {
  }

  onCreateServer() {
    this.serverCreated = true;
    this.serverCreationStatus = 'Server was created! Name is ' + this.serverName;
  }
```

Directives(example9-UsingNgIfToOutputData)

- Edit C:\workspace-UT-Angular8-JAN-2020\session4\example9-UsingNgIfToOutputData\bootstrap4\src\app\servers\servers.component.html

src ▸ app ▸ servers ▸ servers.component.html ▸ p

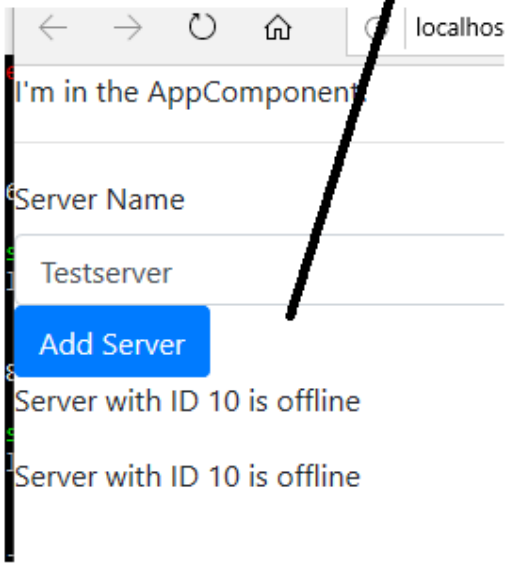
```
1    <label>Server Name</label>
2
3    <!--<input type="text" class="form-control" (input)="onUpdateServerName($event)">--
4    <input type="text" class="form-control" [(ngModel)]="serverName">
5    <!----<p>{{ serverName }} </p>-->
6    <button class="btn btn-primary"
7     [disabled]="!allowNewServer"
8     (click)="onCreateServer()">Add Server</button>
9    <!--<p [innerText]="allowNewServer"></p>-->
10   <!--<p>{{ serverCreationStatus }}</p>-->
11   <p *ngIf="serverCreated">Server was created, server name is {{ serverName }}</p>
12   <app-server></app-server>
13   <app-server></app-server>
14
```

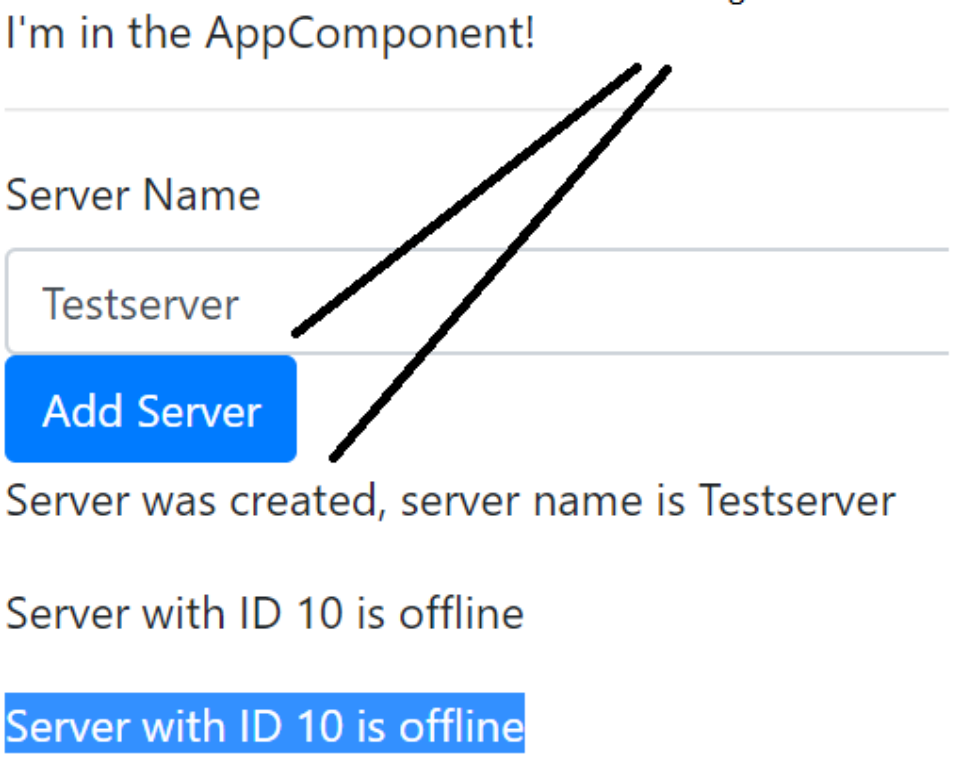


Directives(example9-UsingNgIfToOutputData)

line "Server was created, server name is Testserver" doesn't show.

The line "Server was created.." show only after clicking "Add Server" button.





example10-styligElementsDynamicallyWithNgStyle

- Copy C:\workspace-UT-Angular8-JAN-2020\session4\example9-UsingNgIfToOutputData\bootstrap4 to C:\workspace-UT-Angular7-Sept-2019\session4\example10-styligElementsDynamicallyWithNgStyle
- Unlike structural directives, attribute directives don't add or remove elements. They only change the element they were placed on.
- NgStyle allows us to dynamically assign a style. NgStyle allows us to change the css style itself.
- Edit C:\workspace-UT-Angular8-JAN-2020\session4\example10-styligElementsDynamicallyWithNgStyle\bootstrap4\src\app\server\server.component.ts

example10-styligElementsDynamicallyWithNgStyle

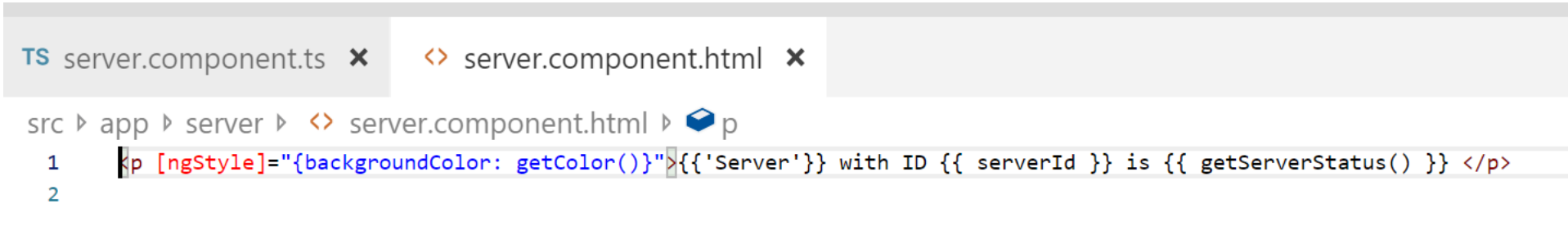
server.component.ts server.component.html

src > app > server > TS server.component.ts > ServerComponent > g

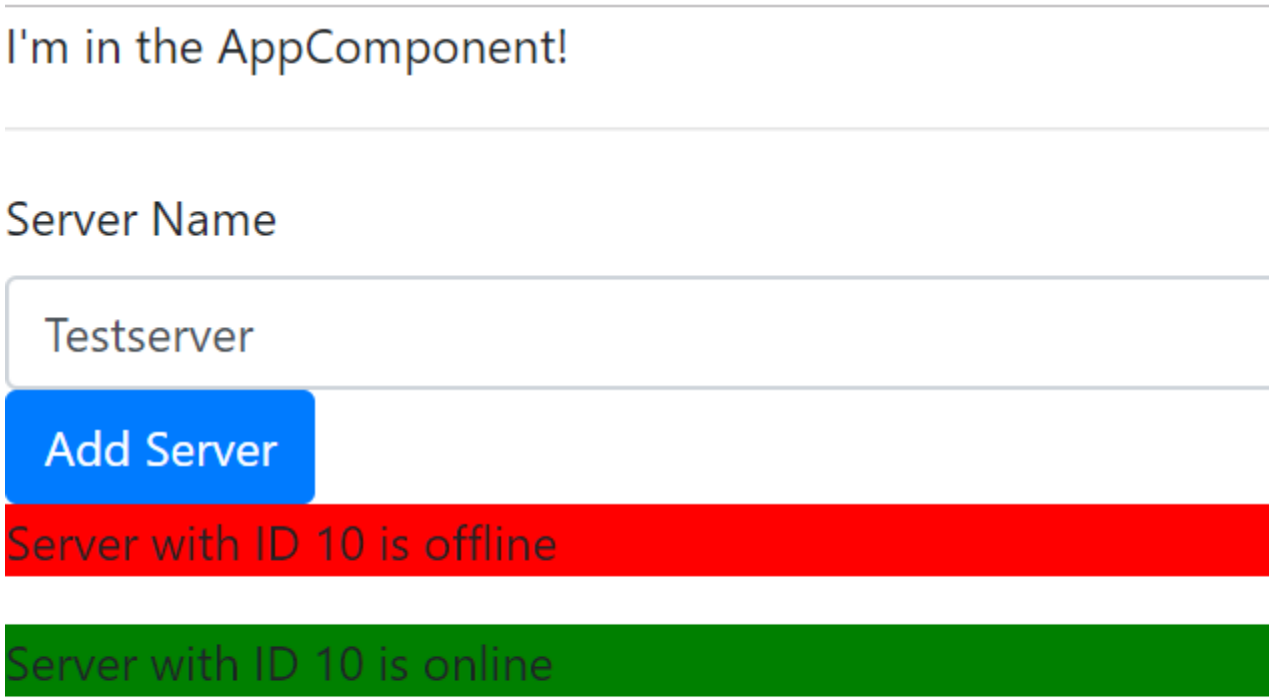
```
1   import { Component } from '@angular/core';
2
3   @Component({
4     selector: 'app-server',
5     templateUrl: './server.component.html'
6   })
7   export class ServerComponent {
8     serverId: number = 10;
9     serverStatus: string = 'offline';
10
11    constructor() {
12      this.serverStatus = Math.random() > 0.5 ? 'online' : 'offline';
13    }
14    getServerStatus(): string {
15      return this.serverStatus;
16    }
17
18    getColor() {
19      return this.serverStatus === 'online' ? 'green' : 'red';
20    }
21  }
```

example10-styligElementsDynamicallyWithNgStyle

- Edit C:\workspace-UT-Angular8-JAN-2020\Session4\example10-styligElementsDynamicallyWithNgStyle\bootstrap4\src\app\server\server.component.html



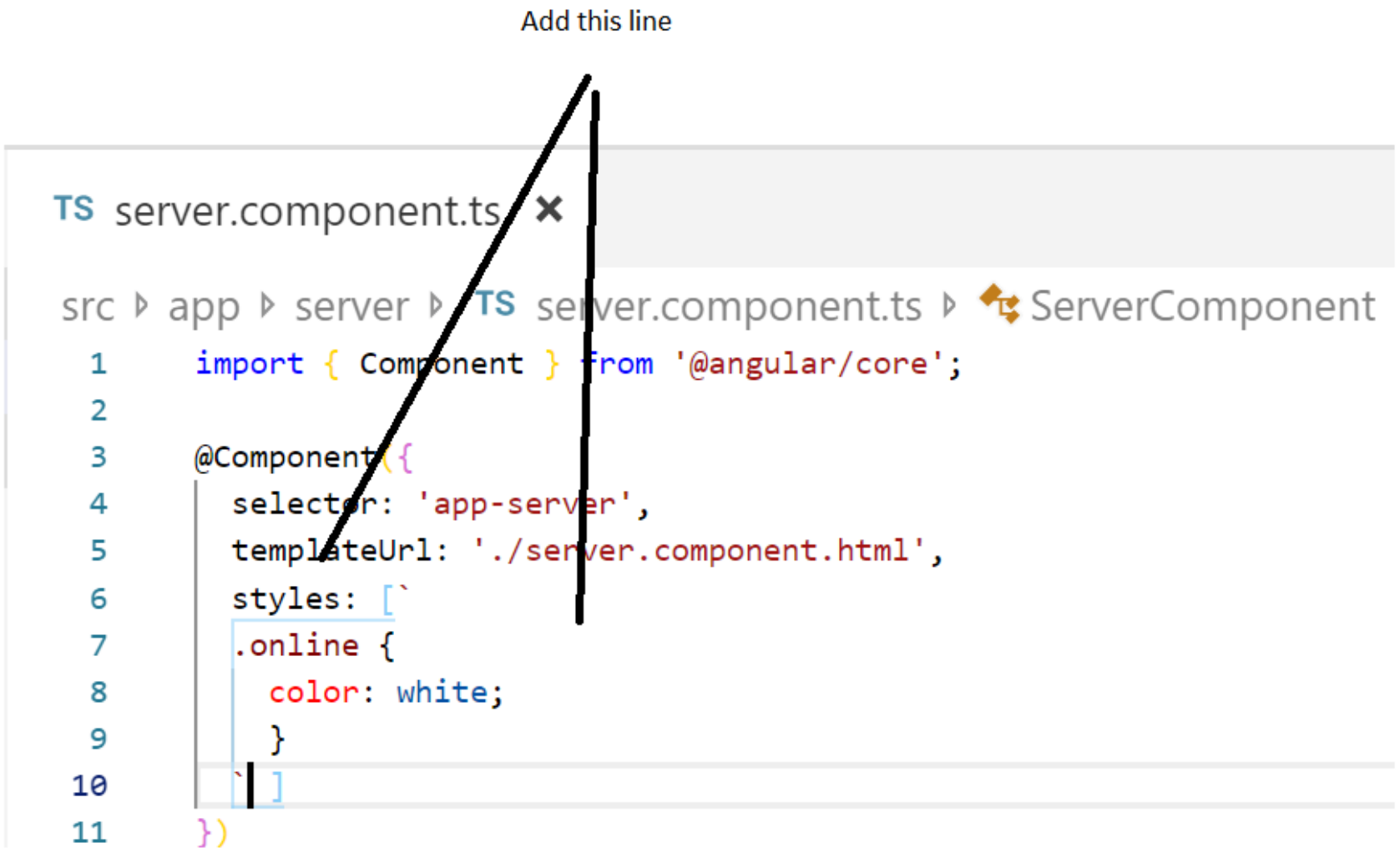
example10-styligElementsDynamicallyWithNgStyle



example11-NgClass

- Copy C:\workspace-UT-Angular8-JAN-2020\session4\example10-styligElementsDynamicallyWithNgStyle\bootstrap4 to C:\workspace-UT-Angular8-JAN-2020\session4\example11-NgClass
- NgClass allows us to dynamically add or remove CSS classes.
- Edit C:\workspace-UT-Angular8-JAN-2020\session4\example11-NgClass\bootstrap4\src\app\server\server.component.ts

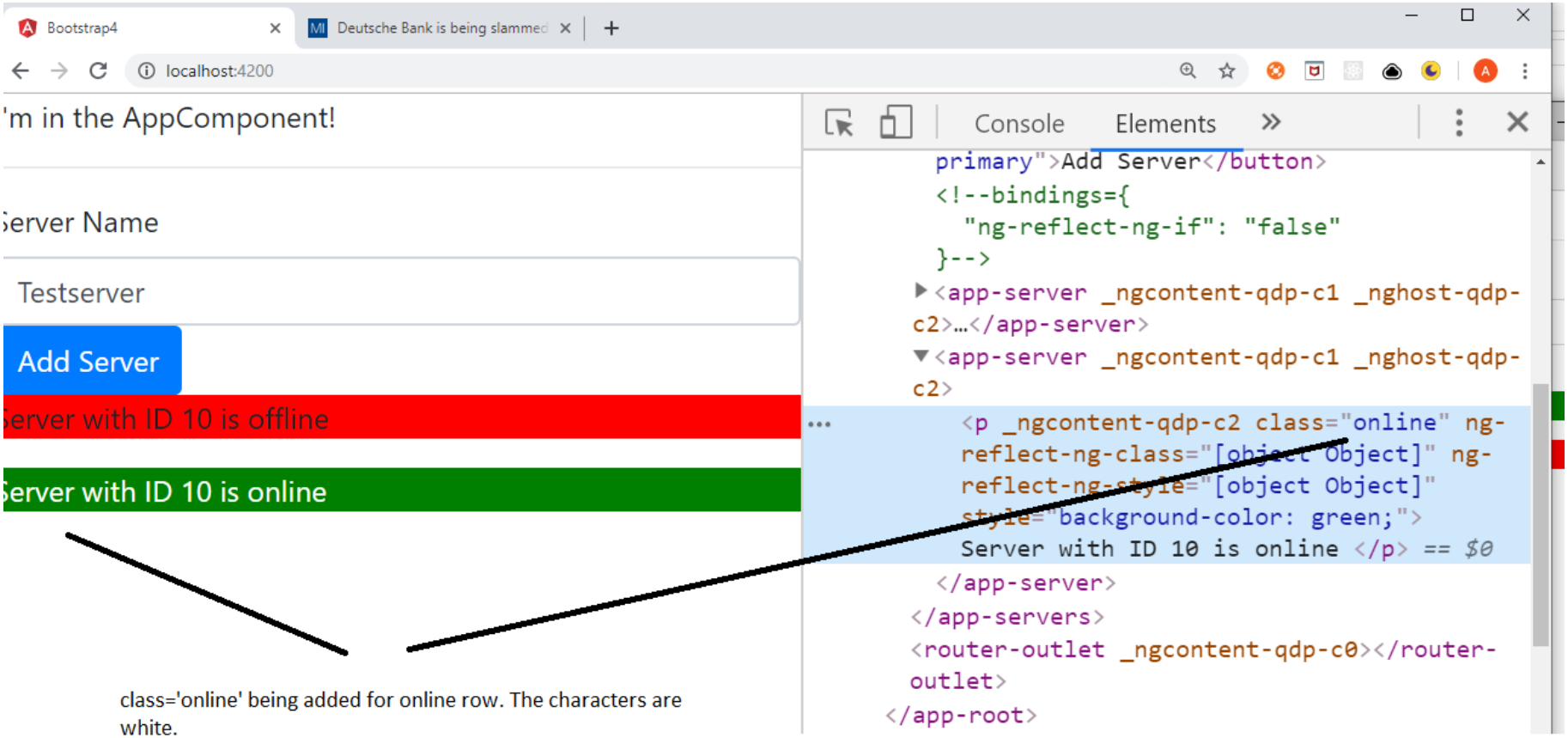
example11-NgClass



example11-NgClass

- C:\workspace-UT-Angular7-Sept-2019\session4\example11-NgClass\bootstrap4\src\app\server\server.component.html

example11-NgClass



example12-NgFor

- Copy C:\workspace-UT-Angular8-JAN-2020\session4\example11-NgClass\bootstrap4 to C:\workspace-UT-Angular8-JAN-2020\session4\example12-NgFor
- Edit C:\workspace-UT-Angular8-JAN-2020\session4\example12-NgFor\bootstrap4\src\app\servers\servers.component.ts

example12-NgFor

```
export class ServersComponent implements OnInit {
  allowNewServer = false;
  serverCreationStatus = 'No server was created!';
  serverName = 'Testserver';
  serverCreated = false;
  servers = ['Testserver', 'Testserver 2'];

  constructor() {
    setTimeout(() => {
      this.allowNewServer = true;
    }, 2000);
  }

  ngOnInit() {
  }

  onCreateServer() {
    this.serverCreated = true;
    this.servers.push(this.serverName);
    this.serverCreationStatus = 'Server was created! Nam
  }
```

Add new array "servers" and push new object to new array 'servers' when onCreateServer event is trigger.

example12-NgFor

example12-NgFor