



# ANGULAR 2 FUNDAMENTALS

---

*December 13, 2017*

# PART 2

# DIGGING DEEPER WITH ANGULAR

---

- What it really does?
- What happens behind the scenes?
- How angular app gets started?
- Which building blocks it can offer?
- Which features we can use?

# CREATE A NEW PROJECT

---

- `ng new angular-app2`
- `cd angular-app2`
- `ng serve`
- <http://localhost:4200>

# ADD STYLING

---

## ➤ Bootstrap

```
npm install --save bootstrap
```

## ➤ JQuery

```
npm install --save jquery
```

## ➤ Open angular-cli.json

Add:

```
"styles": [  
  "../node_modules/bootstrap/dist/css/bootstrap.min.css",  
  "styles.css"  
],
```

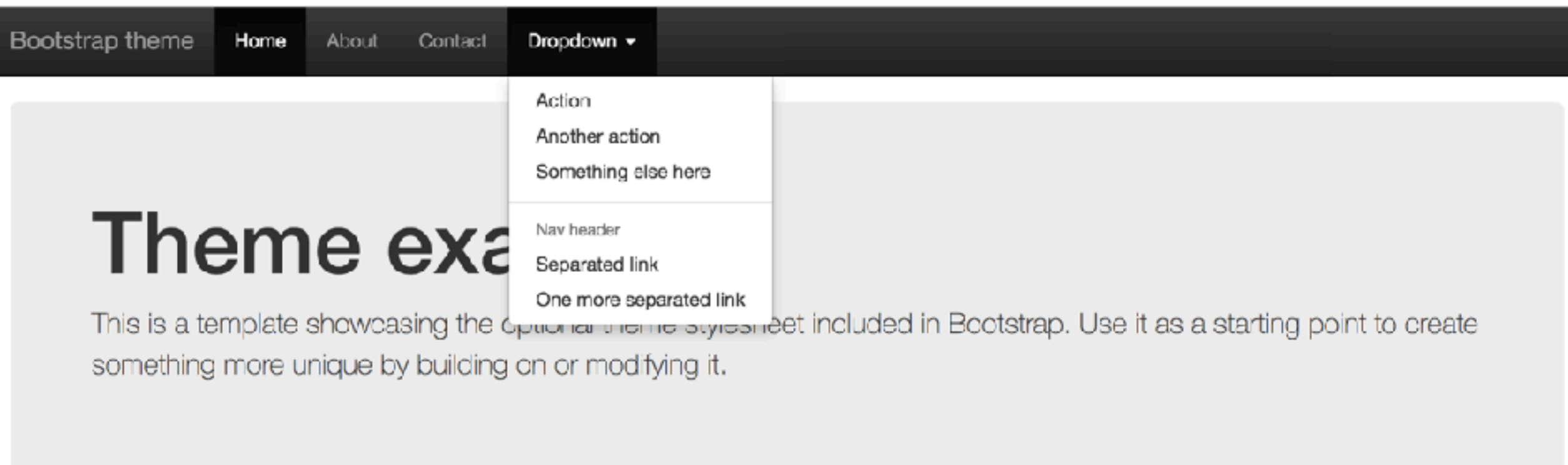
&

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

# TEMPLATING TO HELP YOU IN DEVELOPMENT

---

- <http://getbootstrap.com/docs/3.3/>
- <http://getbootstrap.com/docs/3.3/examples/theme/>



## Buttons





# STARTING THE ANGULAR APP

---

- <http://localhost:4200>
- Content can be changed at the app/app.component.html
- However it is the index.html is the file being served

`<> index.html x`

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>AngularApp2</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
```

# APP.COMPONENT.TS

---

- @Component decorator
- selector
- templateUrl
- styleUrls

TS app.component.ts x

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'app';
10 }
11
```



# HOW ARE THE COMPONENTS INJECTED?

.....

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>AngularApp2</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
<script type="text/javascript" src="inline.bundle.js"></script><script type="text/javascript" src="polyfills.bundle.js"></script><script
type="text/javascript" src="scripts.bundle.js"></script><script type="text/javascript" src="styles.bundle.js"></script><script type="text/javascript"
src="vendor.bundle.js"></script><script type="text/javascript" src="main.bundle.js"></script></body>
</html>
```

*Javascript Bundles are added by the cli.*

# MAIN.TS (FIRST CODE THAT GETS EXECUTED)

---

```
main.ts — angular-app2

EXPLORER

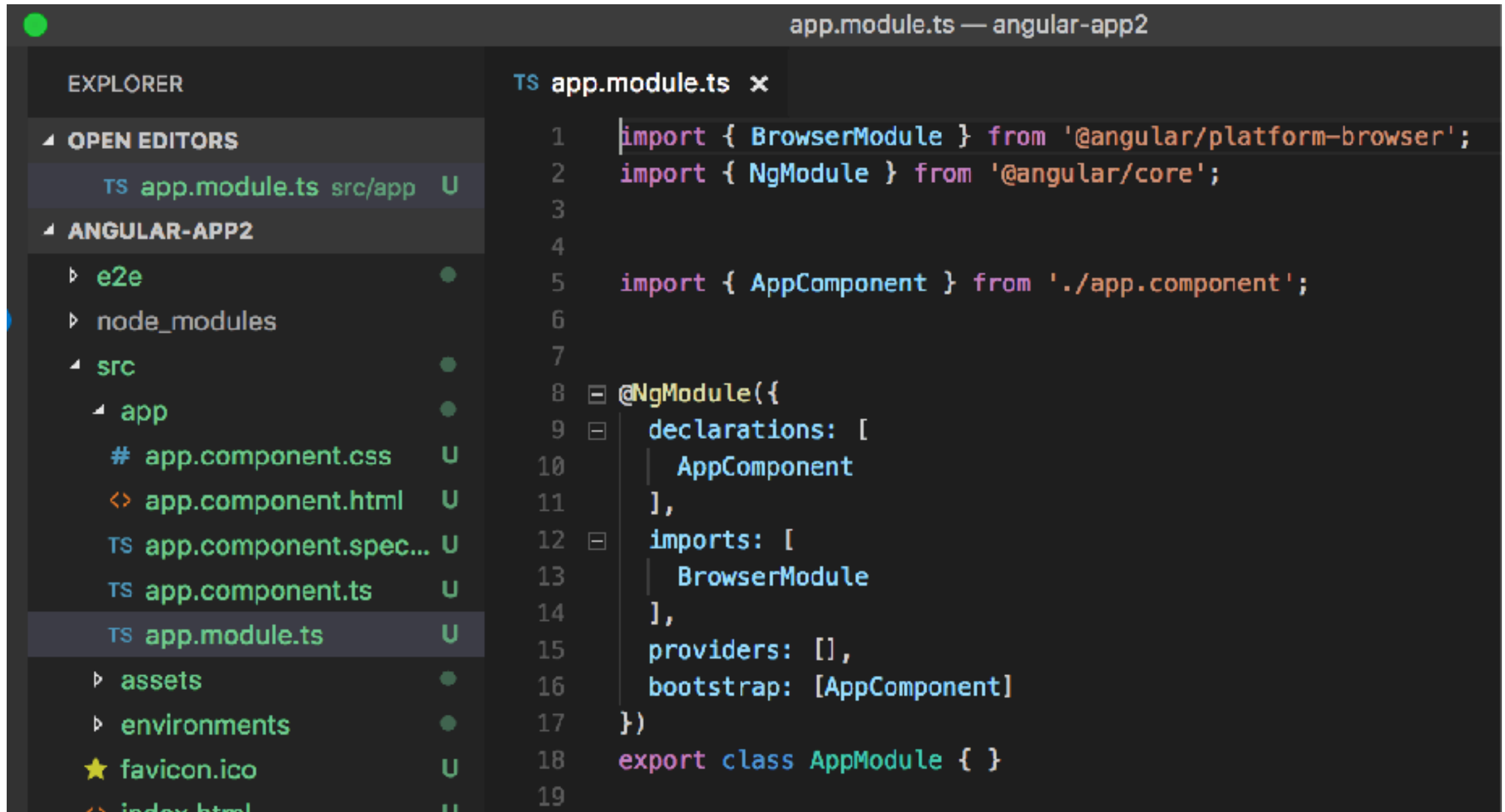
OPEN EDITORS
  TS main.ts src U

ANGULAR-APP2
  e2e
  node_modules
  src
    app
      app.component.css U
      app.component.html U
      app.component.spec... U
      app.component.ts U
      app.module.ts U
    assets
    environments
    favicon.ico U
    index.html U
    TS main.ts U
    TS polyfills.ts U

TS main.ts x

1 import { enableProdMode } from '@angular/core';
2 import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4 import { AppModule } from './app/app.module';
5 import { environment } from './environments/environment';
6
7 if (environment.production) {
8   enableProdMode();
9 }
10
11 platformBrowserDynamic().bootstrapModule(AppModule)
12   .catch(err => console.log(err));
13
```

# APP.MODULE.TS



```
app.module.ts — angular-app2

EXPLORER
└─ OPEN EDITORS
   └─ TS app.module.ts src/app U
└─ ANGULAR-APP2
   ├── e2e
   ├── node_modules
   └─ src
      ├── app
      │   ├── app.component.css U
      │   ├── app.component.html U
      │   ├── app.component.spec... U
      │   ├── app.component.ts U
      │   └─ TS app.module.ts U
      ├── assets
      ├── environments
      ├── favicon.ico U
      └─ index.html U

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

*Bootstrap: [AppComponent] - array which list all components that should be known to angular at the point in time it analyses the index.html file*

# COMPONENTS

---

- A key feature in angular
- Root component (app component) holds our entire application



# MORE ON COMPONENTS

---

- Each component has its own template, styling and business logic
- Split up complex application into reusable parts, use it more than once

# CREATING A NEW COMPONENT

---

- Create a server folder inside app
- Create server.component.ts and server.component.html

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

@Component({
  selector: 'app-server',
  templateUrl: './server.component.html'
})
export class ServerComponent {

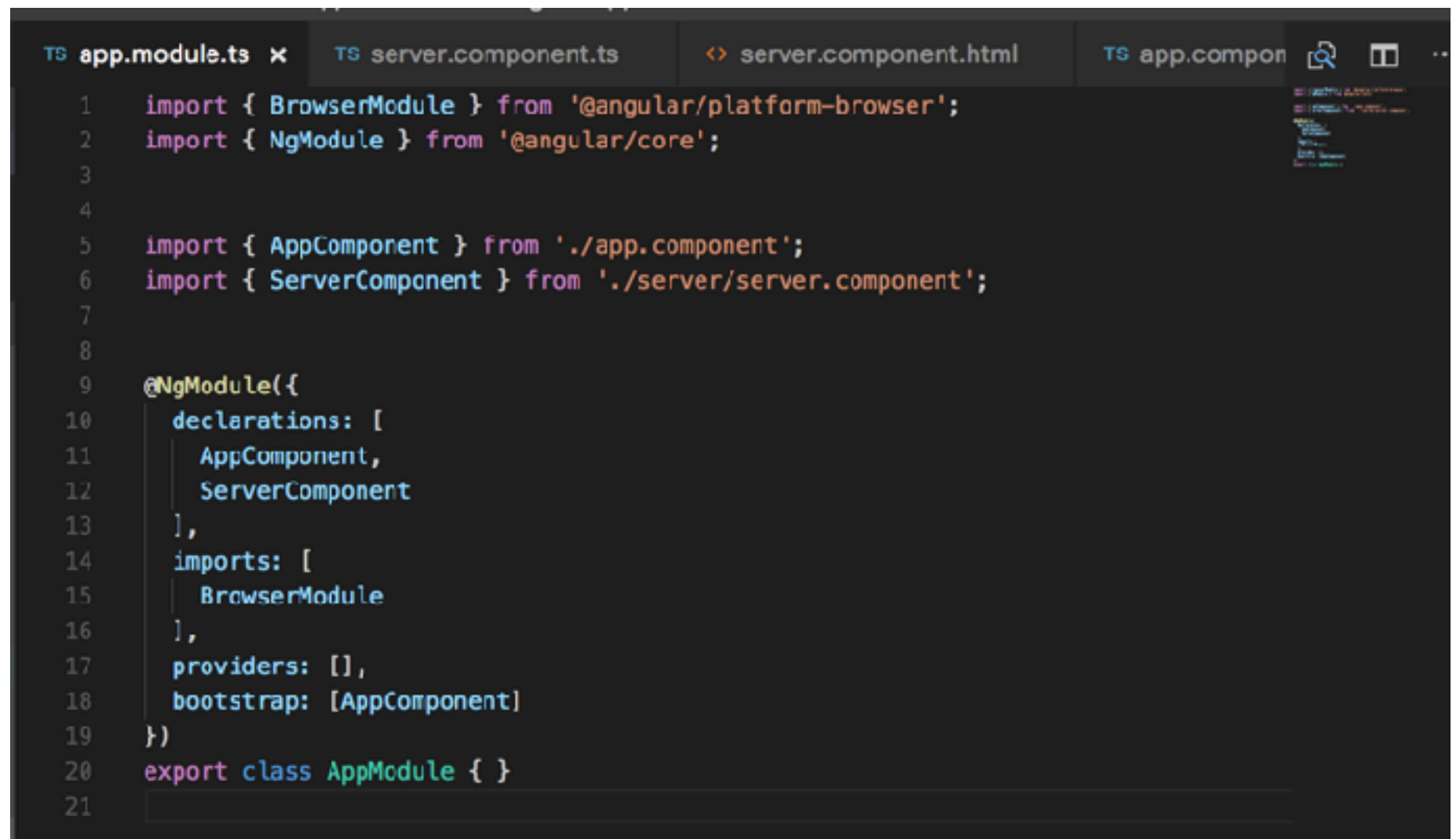
}
```



# THE ROLE OF APP.MODULE.TS AND COMPONENTS

---

- App module bundles components of your app into packages
- @NgModule decorator is imported
- Import new component



```
TS app.module.ts x TS server.component.ts <> server.component.html TS app.component
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4
5 import { AppComponent } from './app.component';
6 import { ServerComponent } from './server/server.component';
7
8
9 @NgModule({
10   declarations: [
11     AppComponent,
12     ServerComponent
13   ],
14   imports: [
15     BrowserModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
21
```

## USE THE NEW COMPONENT

```
1 <!--The content below is only a placeholder and can be replaced.-->
2 <div style="text-align:center">
3   <h1>
4     Welcome to {{ title }}!
5   </h1>
6   
8
9 <app-server></app-server>
10
11 <h2>Here are some links to help you start: </h2>
12 <ul>
13   <li>
14     <h2><a target="_blank" rel="noopener" href="https://angular.io/tutorial">Tour
15   </li>
16   <li>
17     <h2><a target="_blank" rel="noopener" href="https://github.com/angular/angular"
18   </li>
19   <li>
20     <h2><a target="_blank" rel="noopener" href="https://blog.angular.io/">Angular
21   </li>
22 </ul>
```

# CREATING COMPONENTS USING THE CLI

---

- `ng generate component component_name` or `ng g c component_name`
- Try not out

# COMPONENT STYLES

---

<> servers.component.html x

```
1 <p class="sample">
2   servers works!
3 </p>
4
```

<> servers.component.html # servers.component.css x

```
1 .sample {
2   color: green;
3 }
```

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

# ASSIGNMENT

---

- Create 2 new components (manually or with cli)
- Output them in the app component
- One is a warning box
- Another is a success box
- Style them accordingly

**END OF PART 2**