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 query

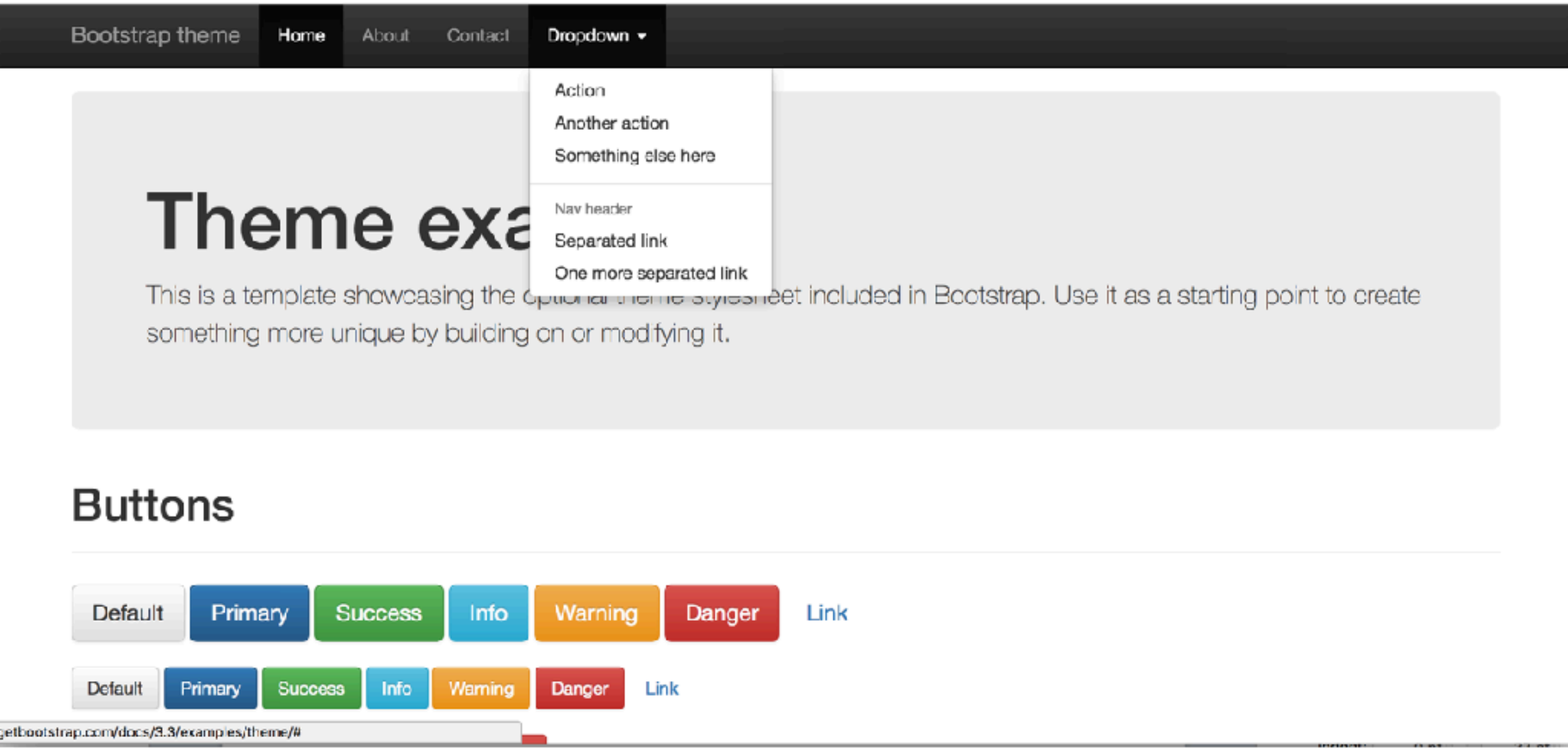
- Open angular-cli.json

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/



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

```
<!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>
</body>
</html>

```

APP.COMPONENT.TS

- @Component decorator
- selector
- templateUrl
- styleUrls

TS app.component.ts ×

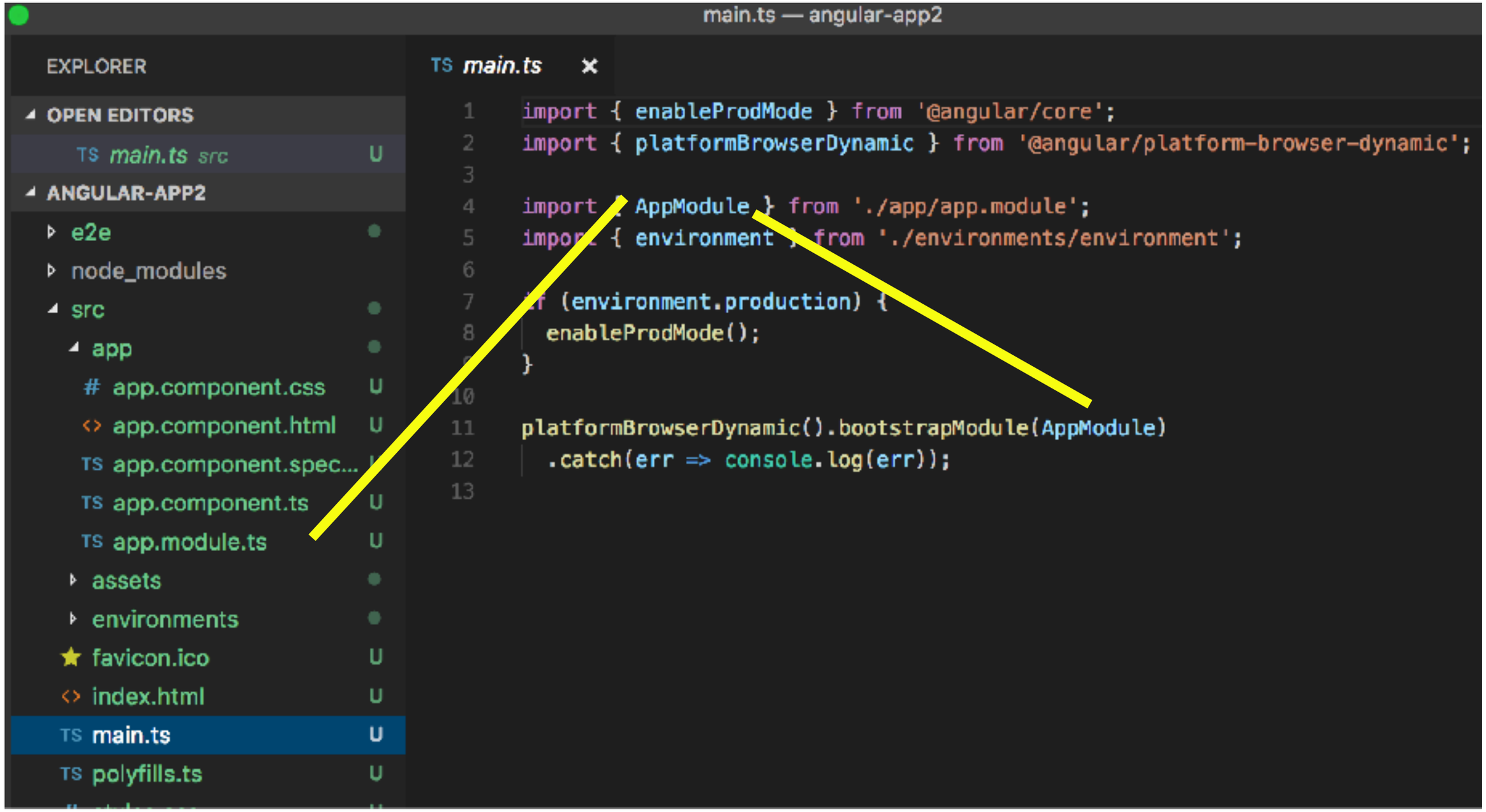
```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app';
}

```

HOW ARE THE COMPONENTS INJECTED?

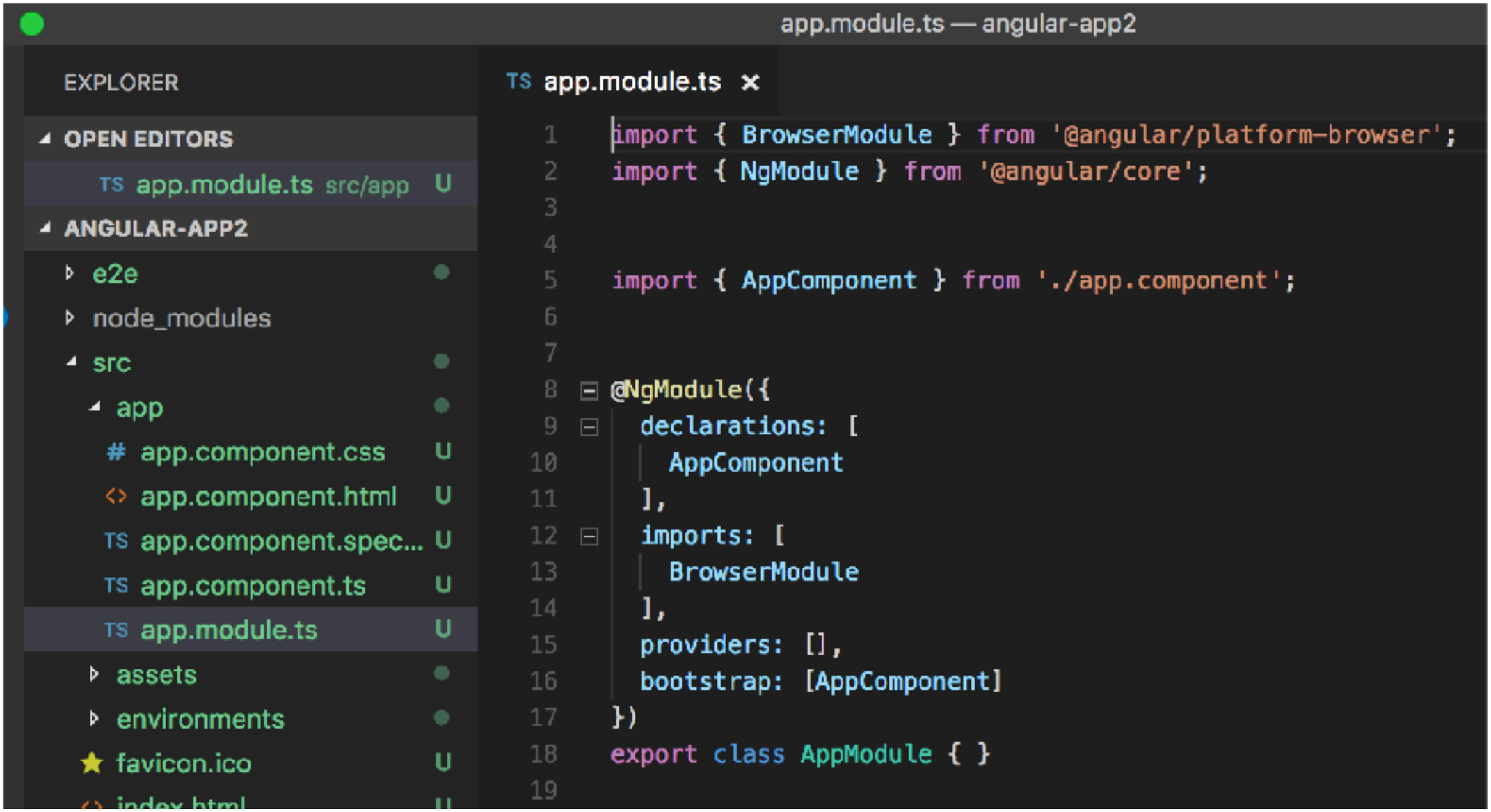
```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>AngularApp2</title>
  <base href="/">

  <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)



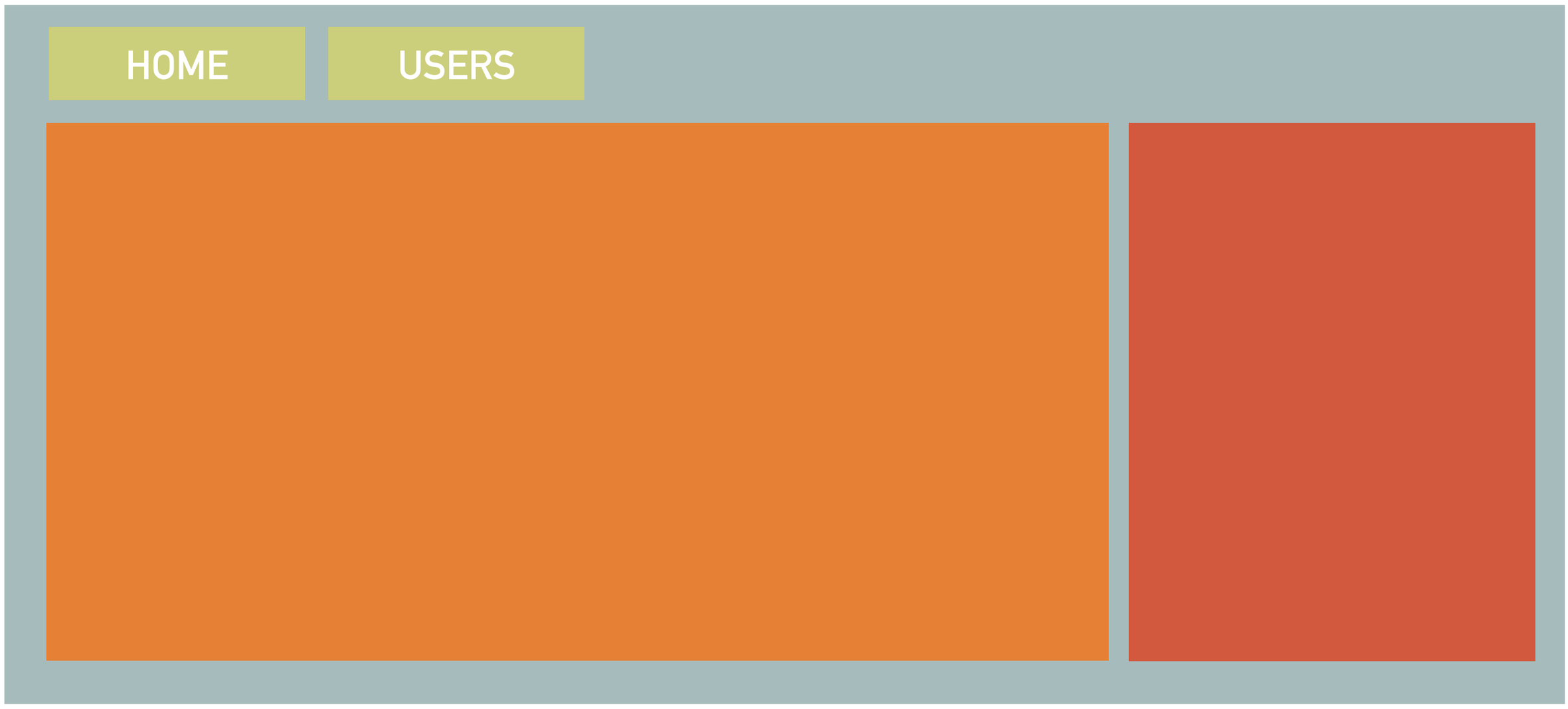
APP.MODULE.TS



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

app.module.ts | server.component.ts | server.component.html | app.compon

```
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

```
<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center">
  <h1>
    Welcome to {{ title }}!
  </h1>
  <img width="300" alt="Angular Logo" src="
</div>

<app-server></app-server>

<h2>Here are some links to help you start: </h2>
<ul>
  <li>
    <h2><a target="_blank" rel="noopener" href="https://angular.io/tutorial">Tour
  </li>
  <li>
    <h2><a target="_blank" rel="noopener" href="https://github.com/angular/angular
  </li>
  <li>
    <h2><a target="_blank" rel="noopener" href="https://blog.angular.io/">Angular
  </li>
</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

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

servers.component.html | servers.component.css

```
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