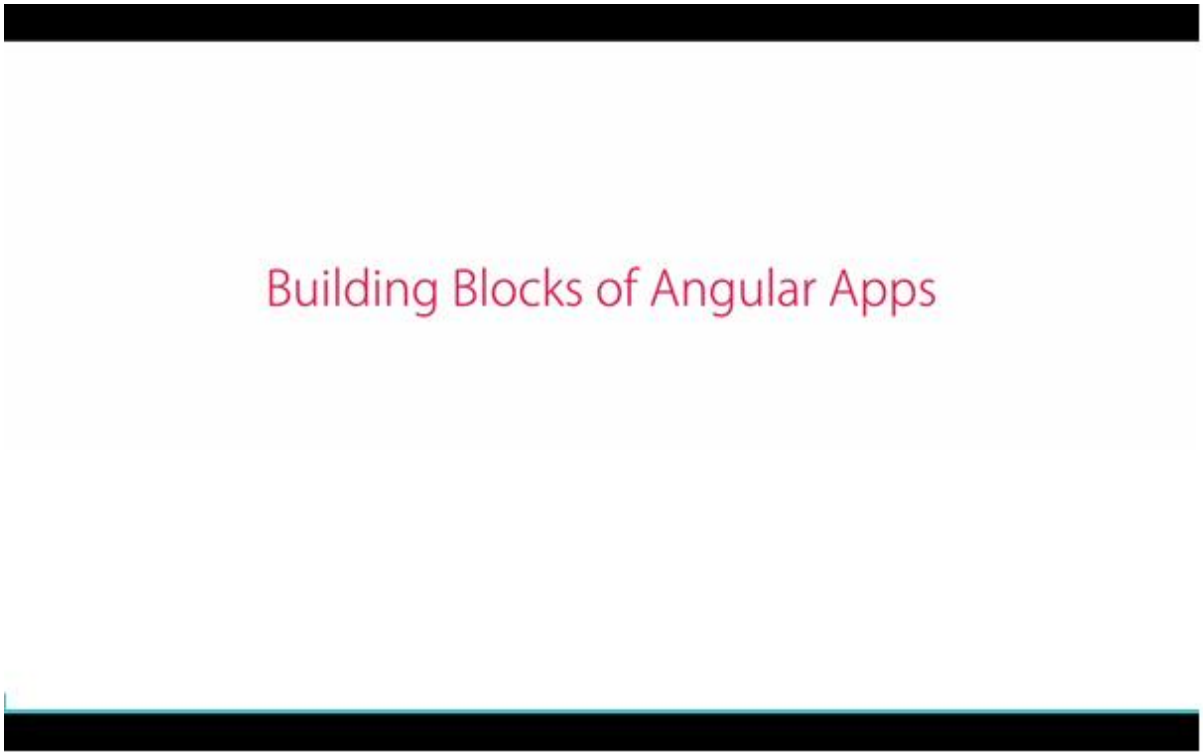
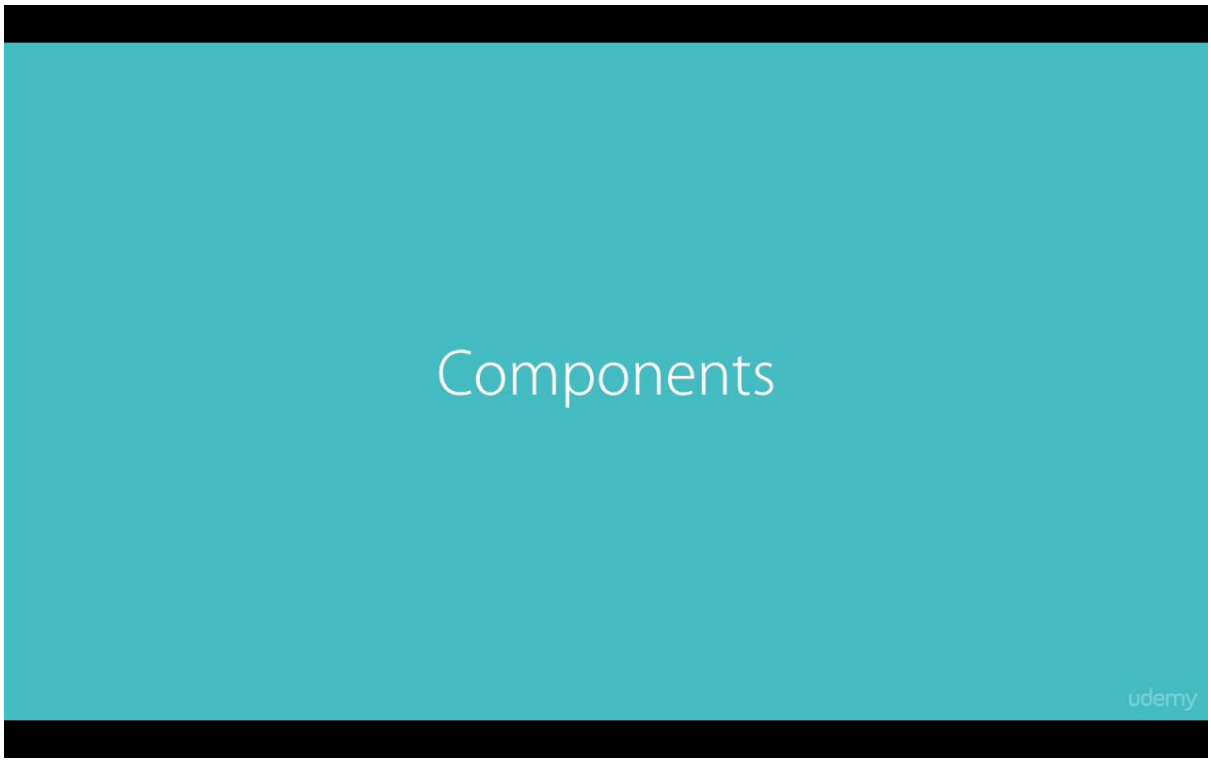
# Angular Fundamentals

## In this section...

- Components
- Templates
- Directives
- Services

# Building Blocks of Angular Apps

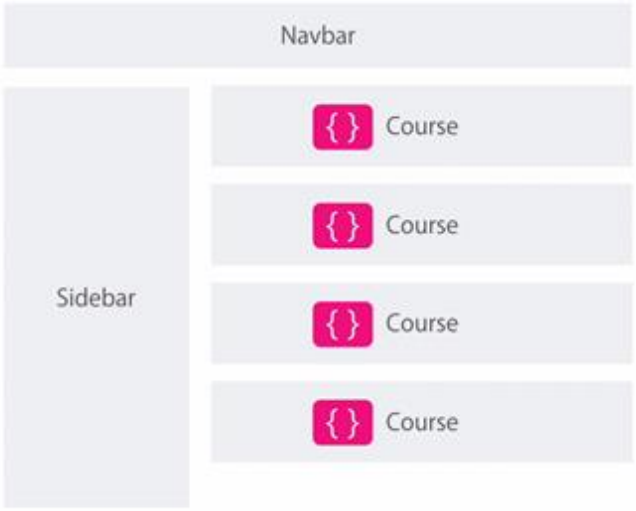


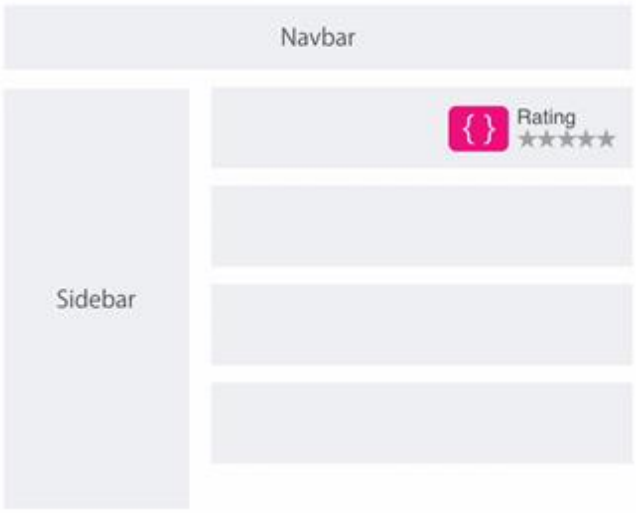



# Component



Navbar
Sidebar
Courses


Navbar
Sidebar
Course
Course
Course
Course

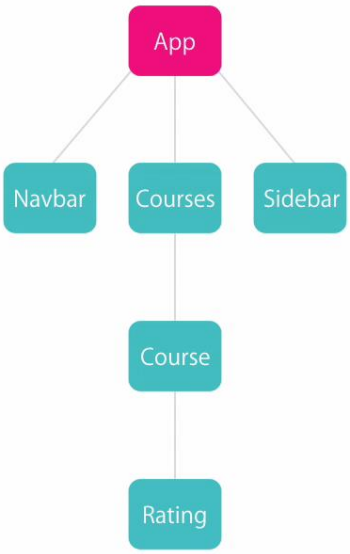



# Component

Data

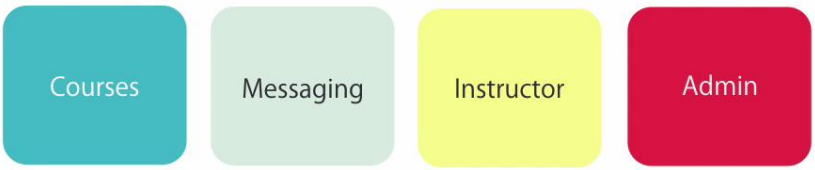
HTML Template

Logic




# Modules

# Modules



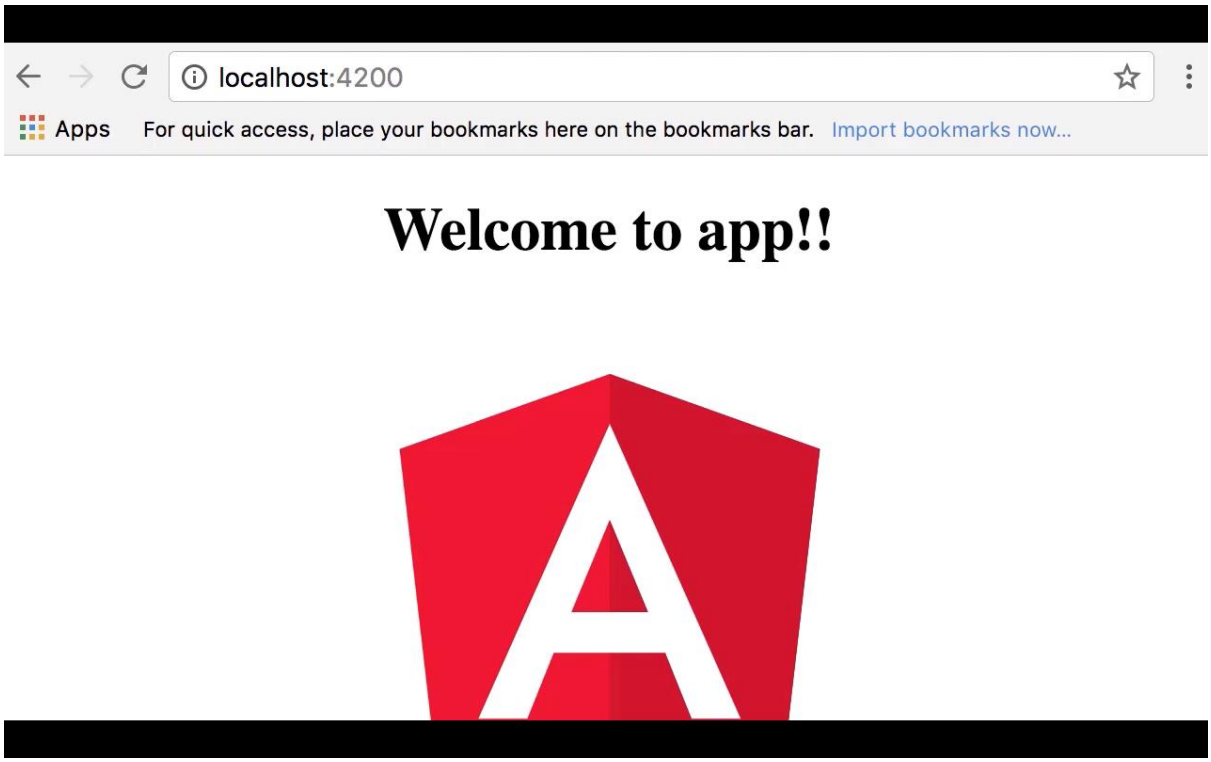

# Components

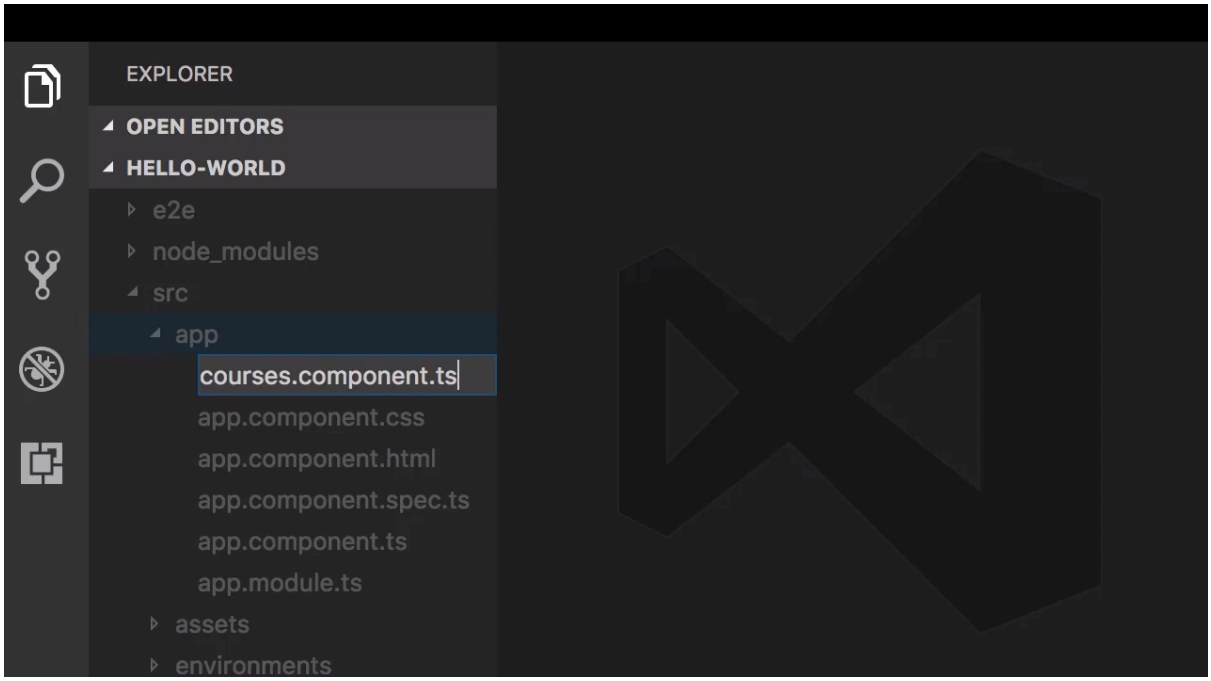
## Creating Components

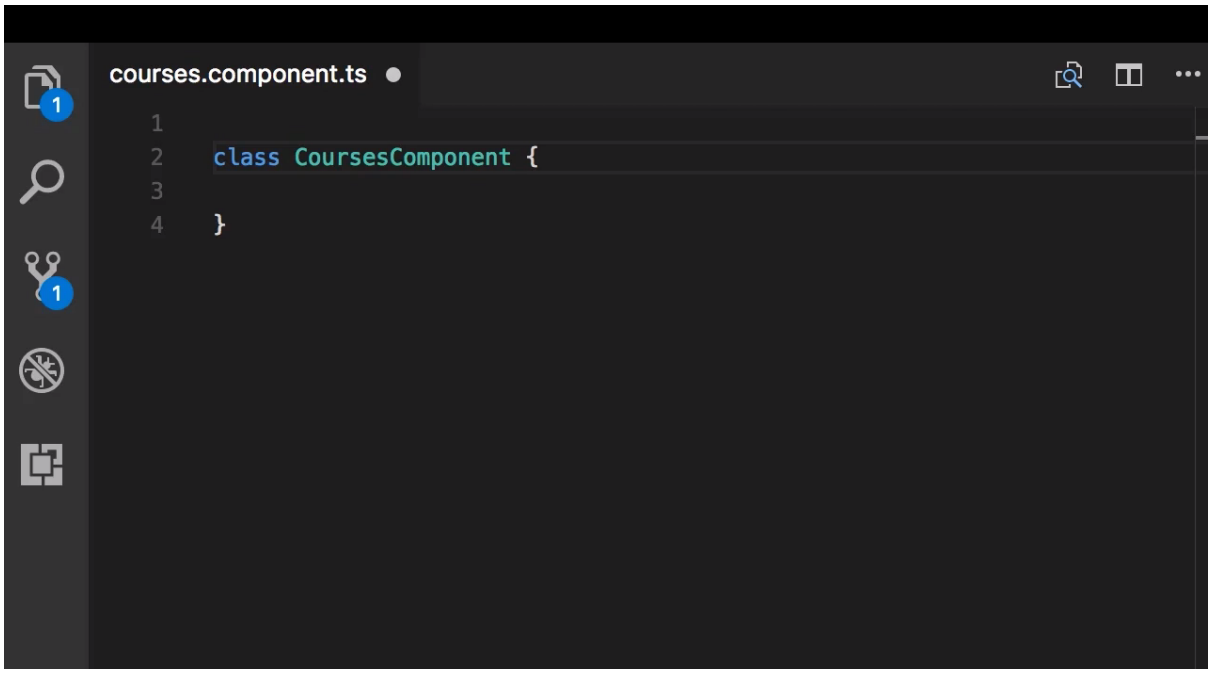
## Components

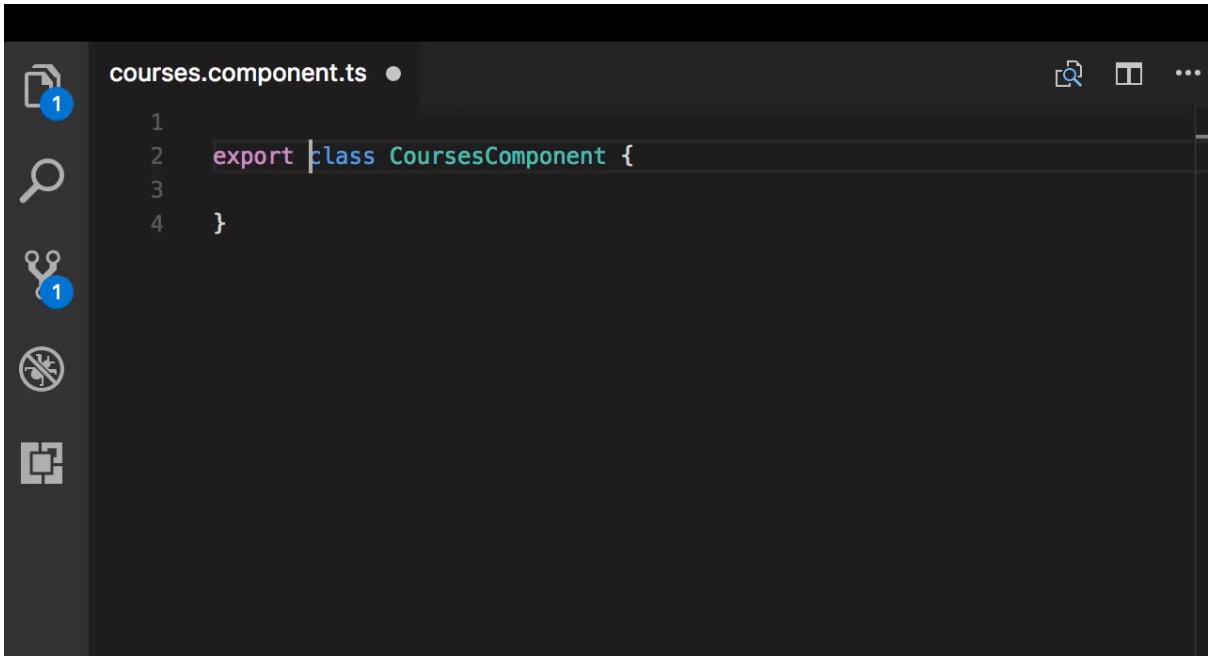
- **Create** a component
- **Register** it in a module
- Add an element in an **HTML markup**

```
hello-world $
hello-world $ng serve
** NG Live Development Server is listening on localhost:4200, open your
 browser on http://localhost:4200 **
Hash: 90d90e5e77621daaaecf                                              d
Time: 8196ms
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 1
58 kB {4} [initial] [rendered]
chunk    {1} main.bundle.js, main.bundle.js.map (main) 5.29 kB {3} [ini
tial] [rendered]
chunk    {2} styles.bundle.js, styles.bundle.js.map (styles) 10.5 kB {4
} [initial] [rendered]
chunk    {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.11 MB [i
nitial] [rendered]
chunk    {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [e
ntry] [rendered]
webpack: Compiled successfully.
```



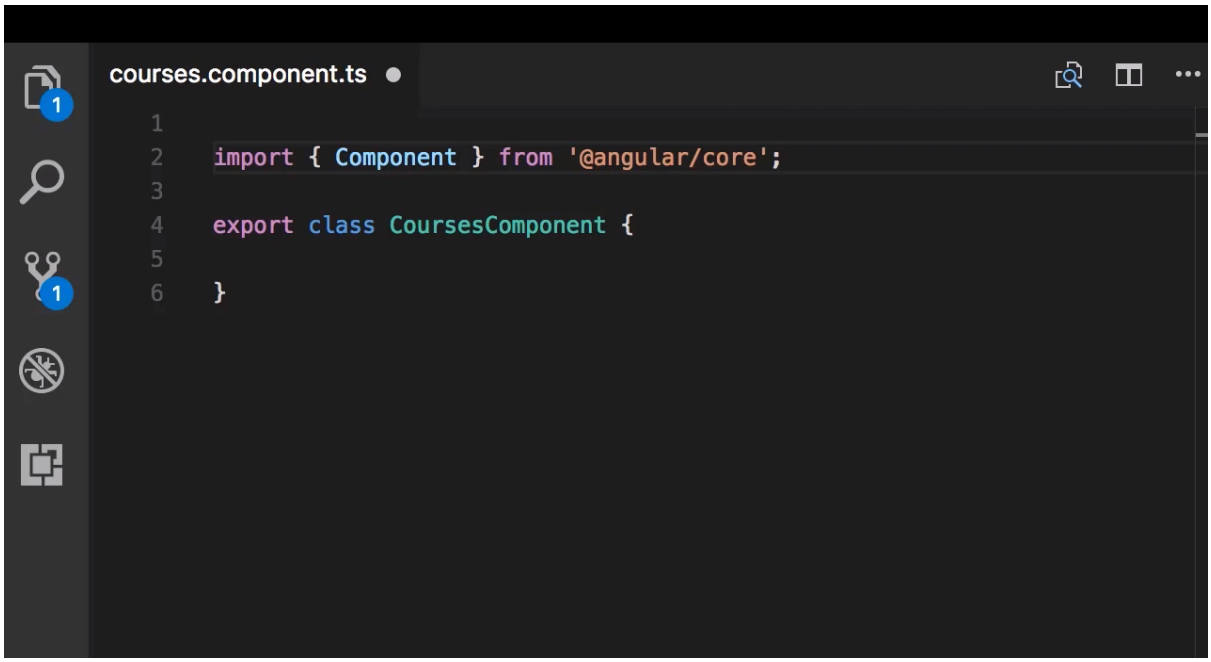

EXPLORER
OPEN EDITORS
HELLO-WORLD
e2e
node_modules
src
app
courses.component.ts
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts
assets
environments

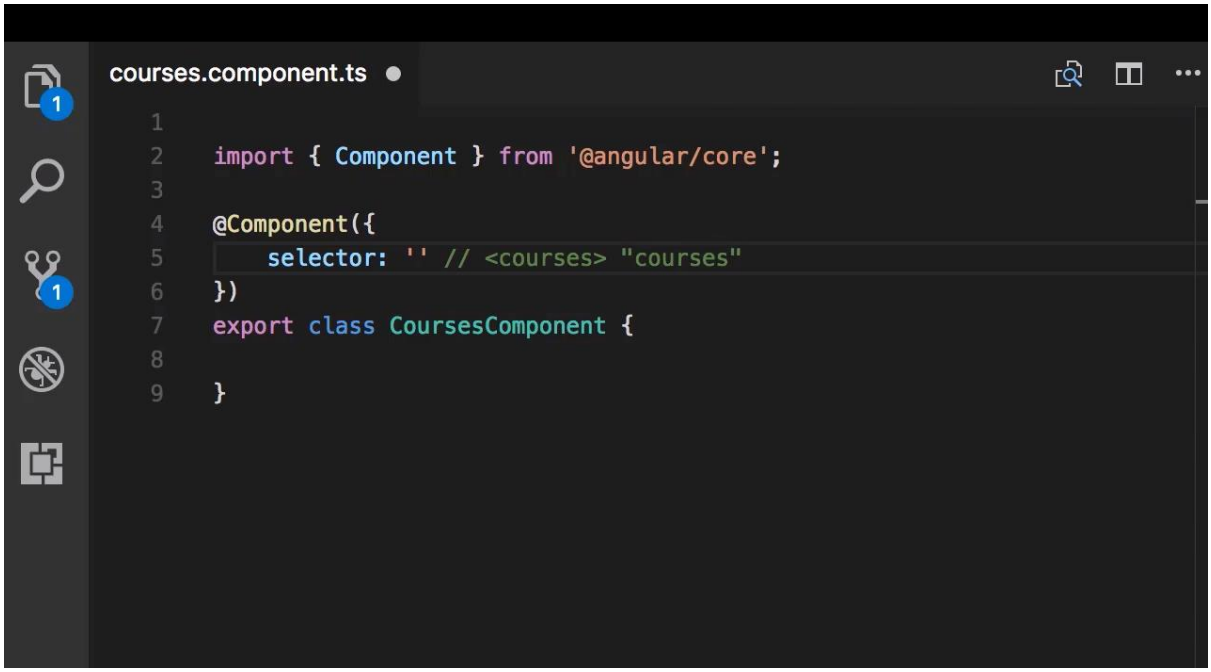

courses.component.ts
1
2 class CoursesComponent {
3
4 }

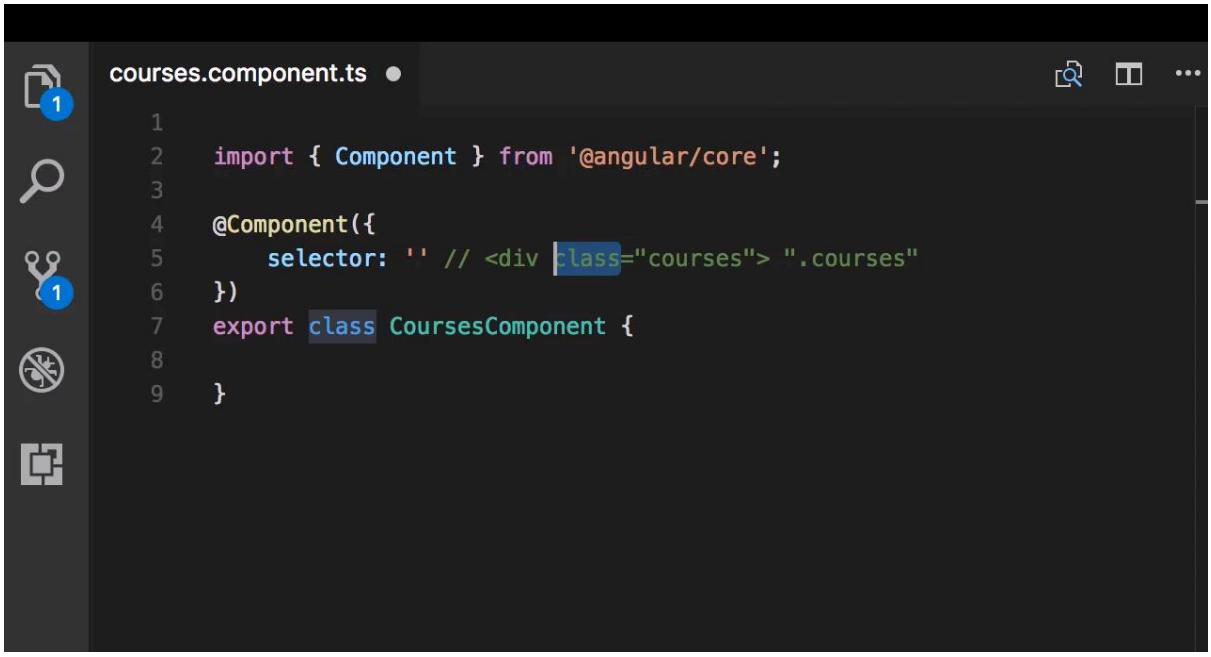


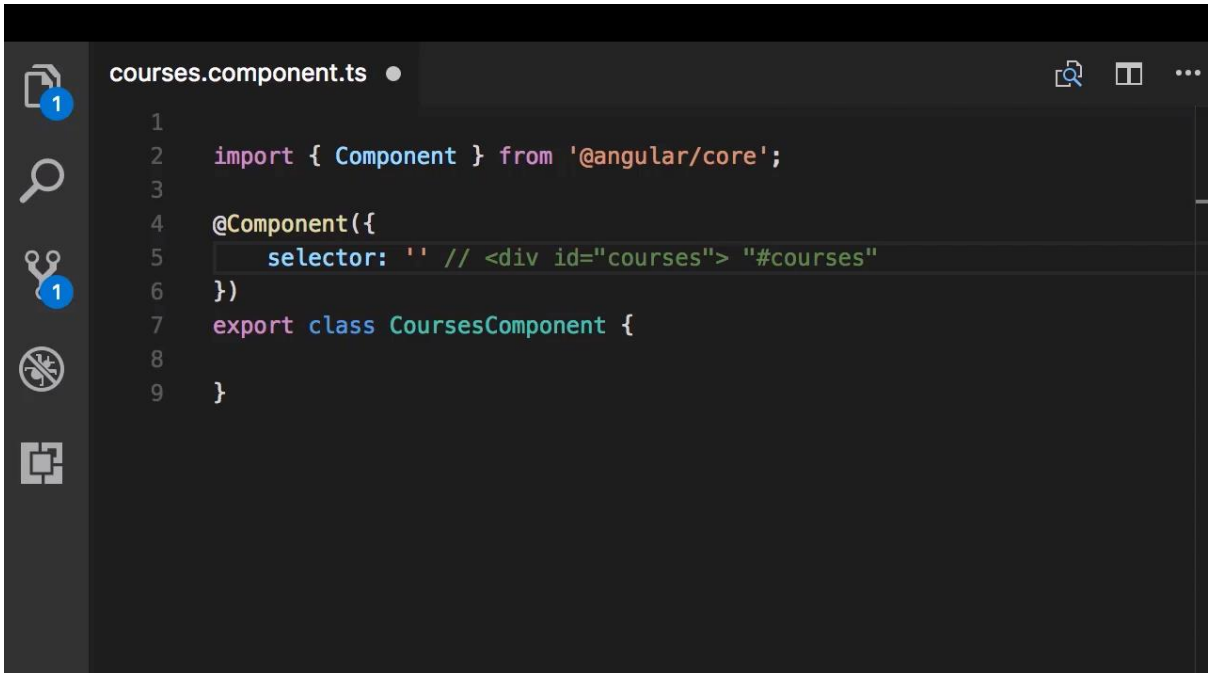

Component decorator makes a component visible to angular:

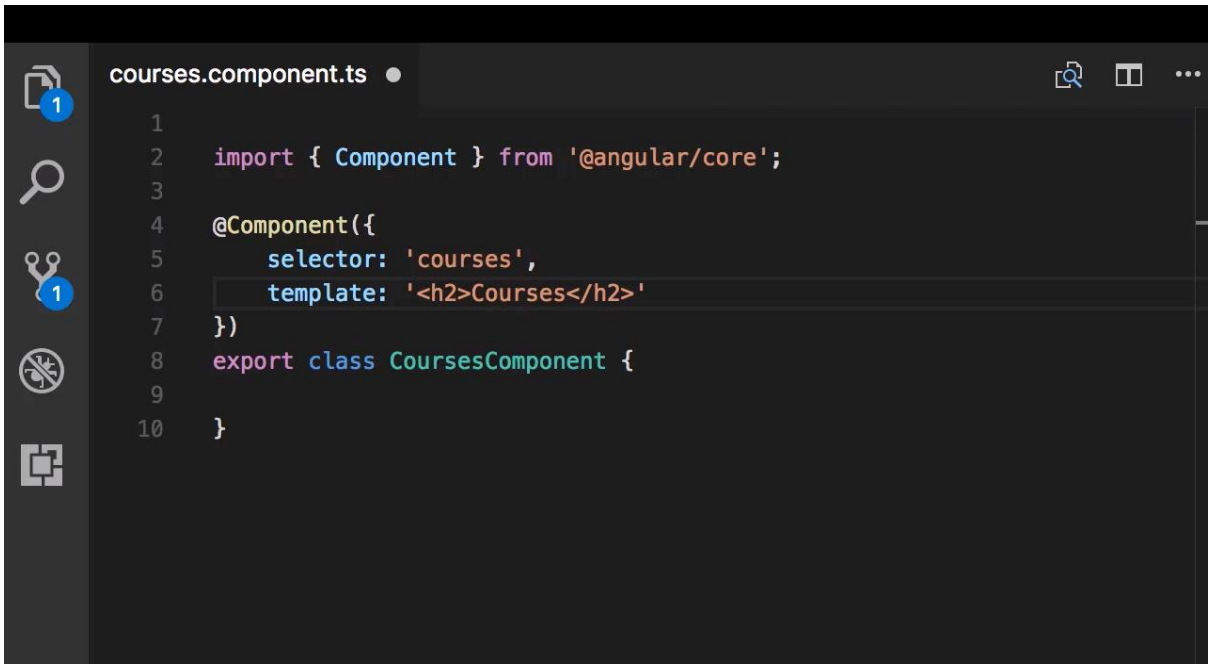
So import it and apply it



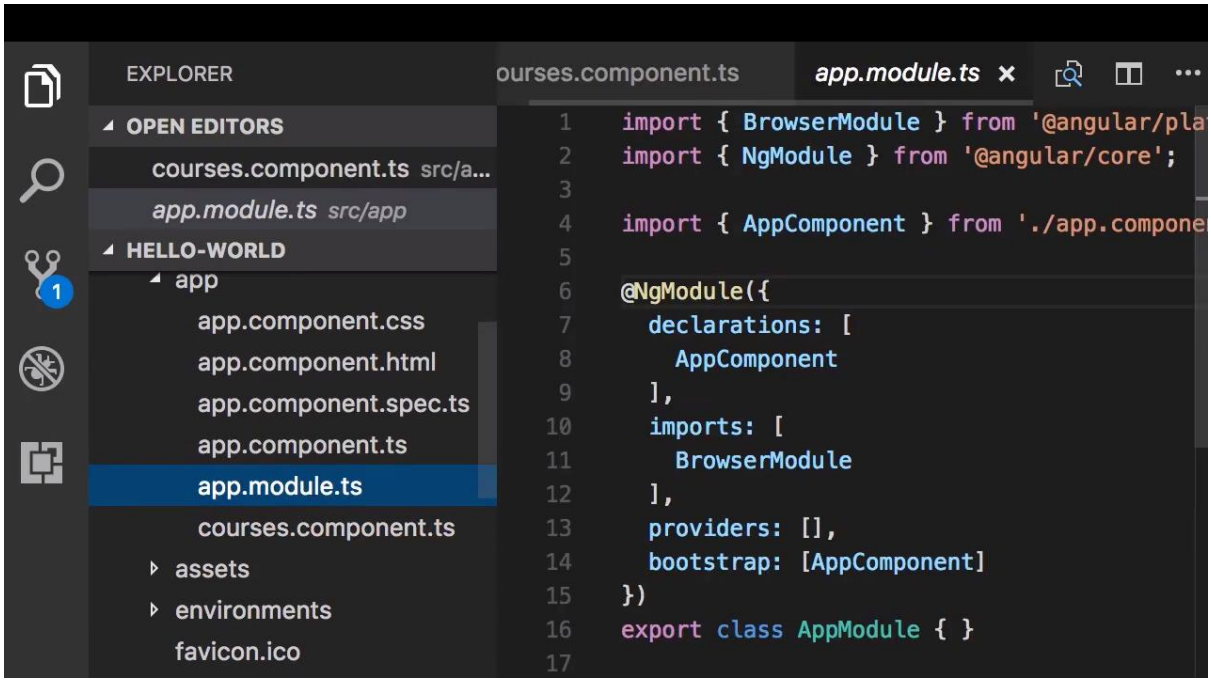

courses.component.ts
import { Component } from '@angular/core';
@Component({
selector: '' // <courses> "courses"
})
export class CoursesComponent {
}

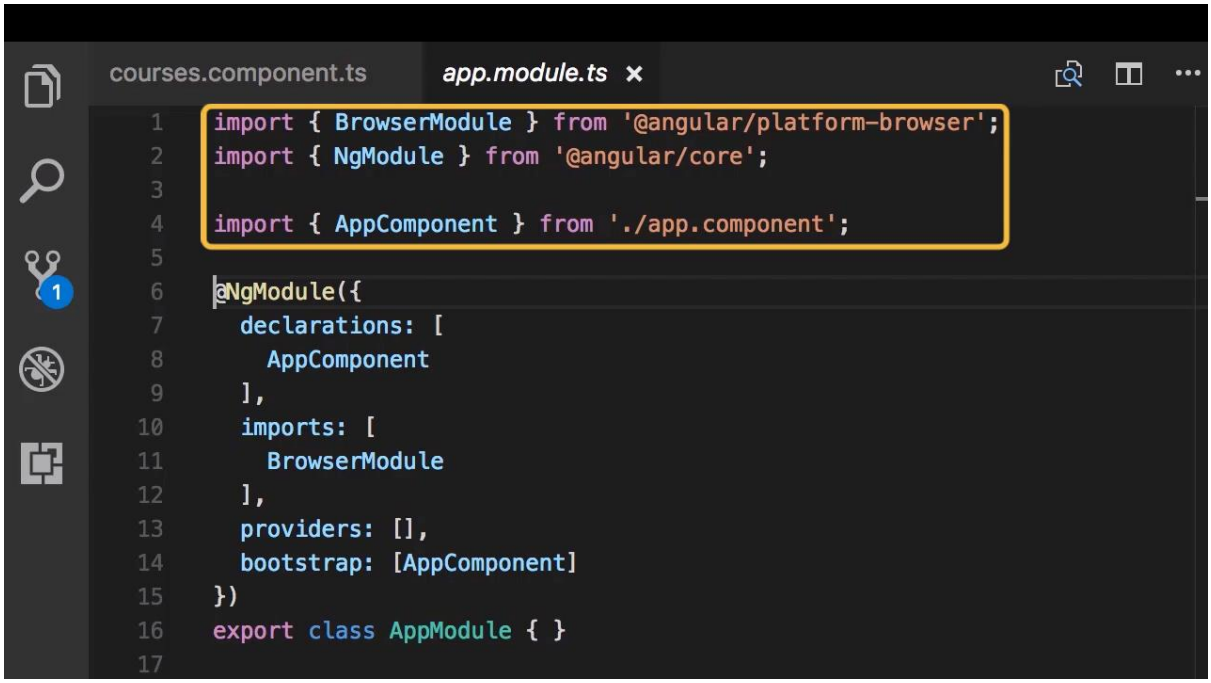

courses.component.ts
import { Component } from '@angular/core';
@Component({
selector: '' //  ".courses"
})
export class CoursesComponent {
}

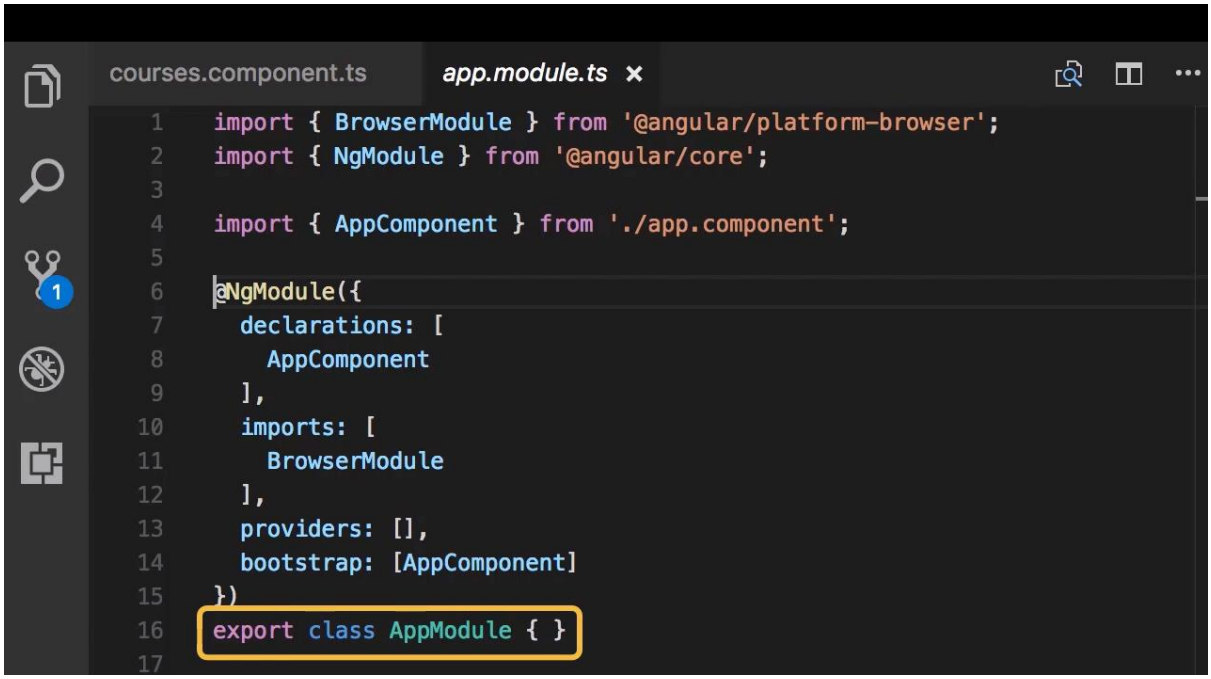


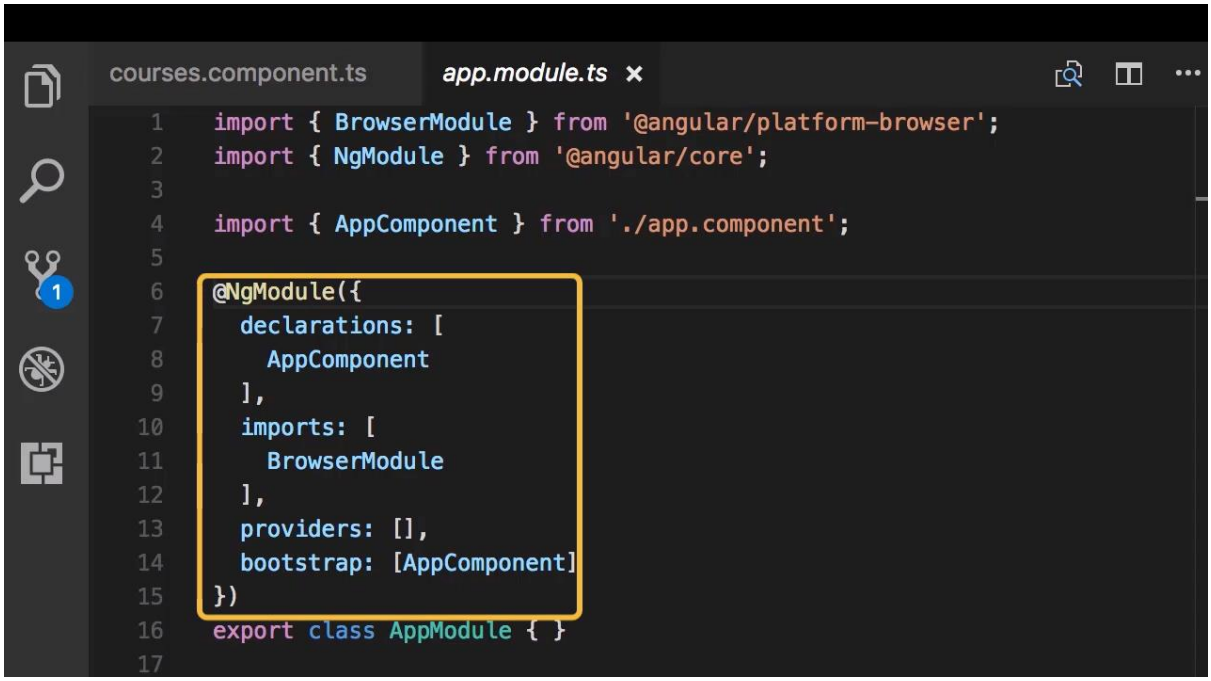

The next step is to register the component

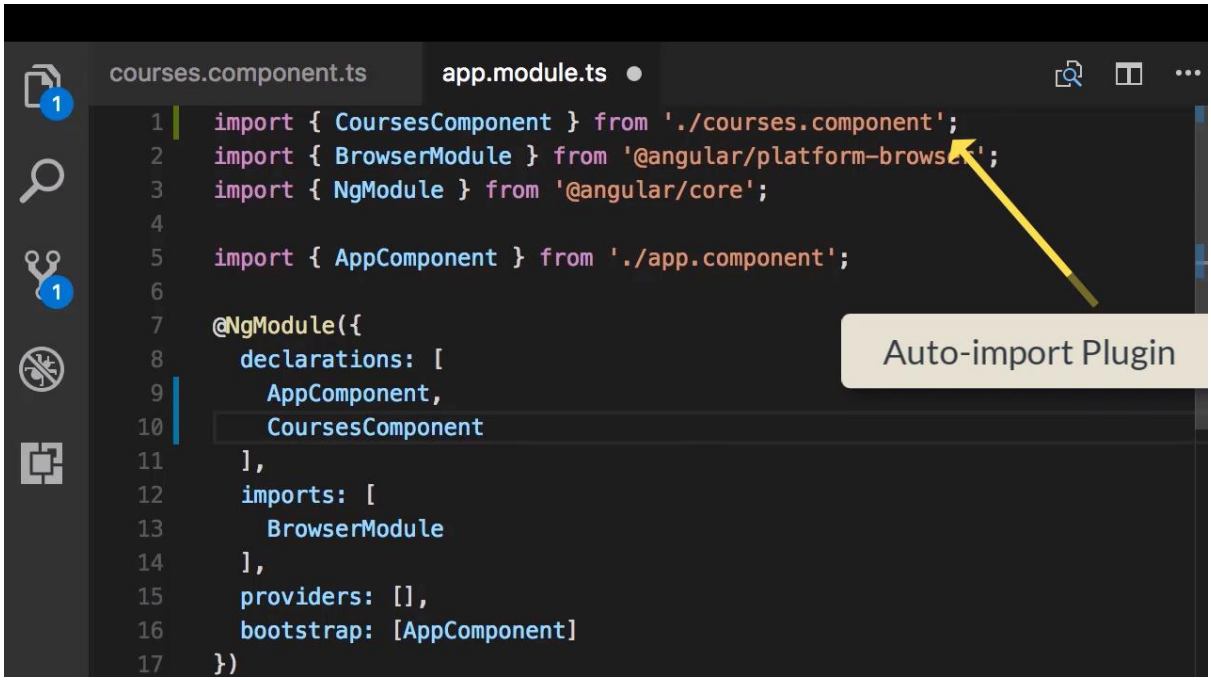

EXPLORER
ourses.component.ts
app.module.ts
OPEN EDITORS
courses.component.ts src/a...
app.module.ts src/app
HELLO-WORLD
app
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts
courses.component.ts
assets
environments
favicon.ico
import { BrowserModule } from '@angular/pla
import { NgModule } from '@angular/core';
import { AppComponent } from './app.compone
@NgModule({
declarations: [
AppComponent
],
imports: [
BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

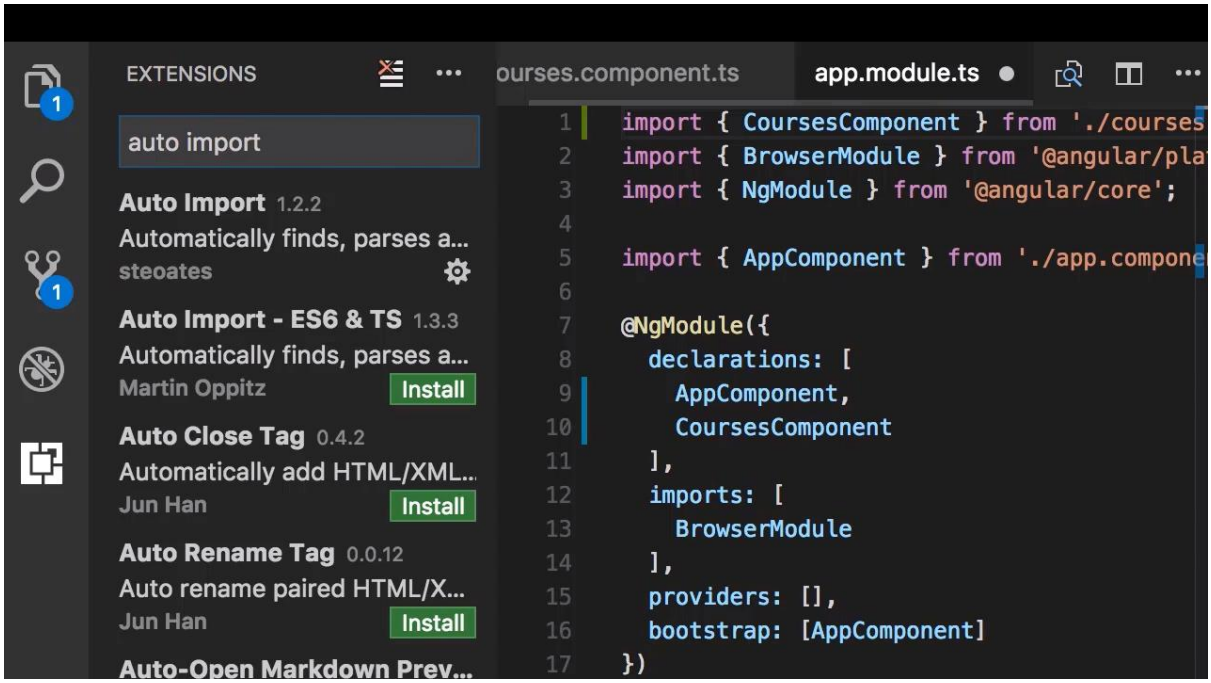

courses.component.ts
app.module.ts
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
@NgModule({
declarations: [
AppComponent
],
imports: [
BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

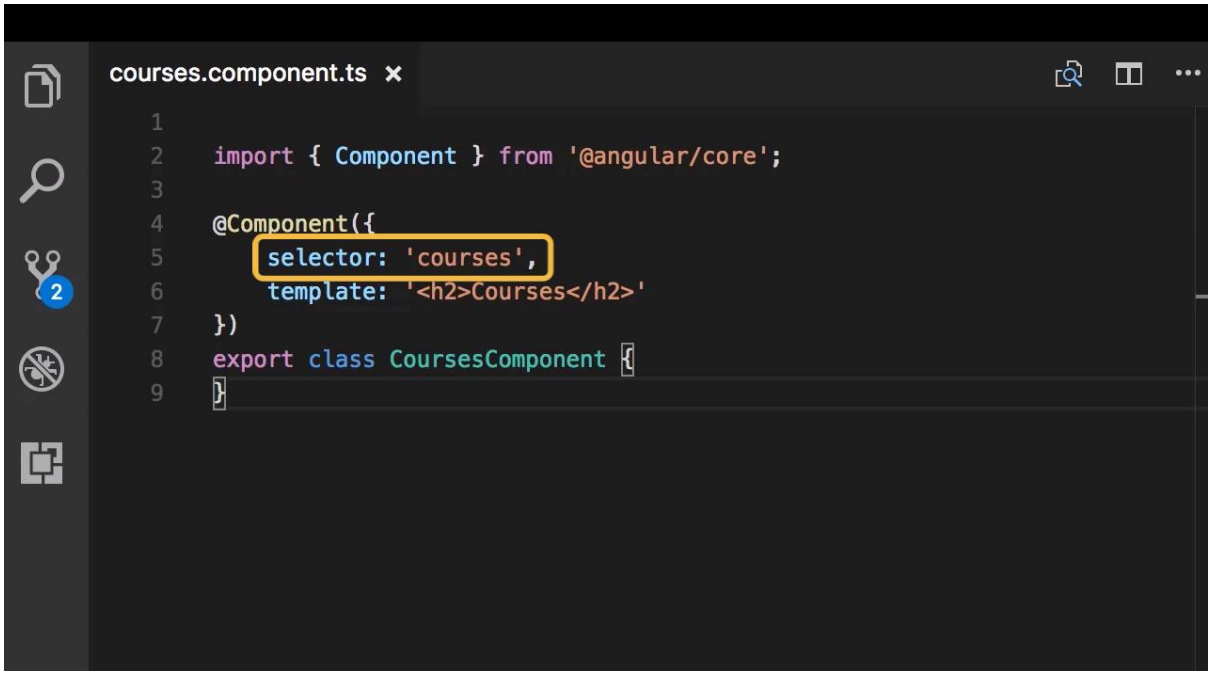


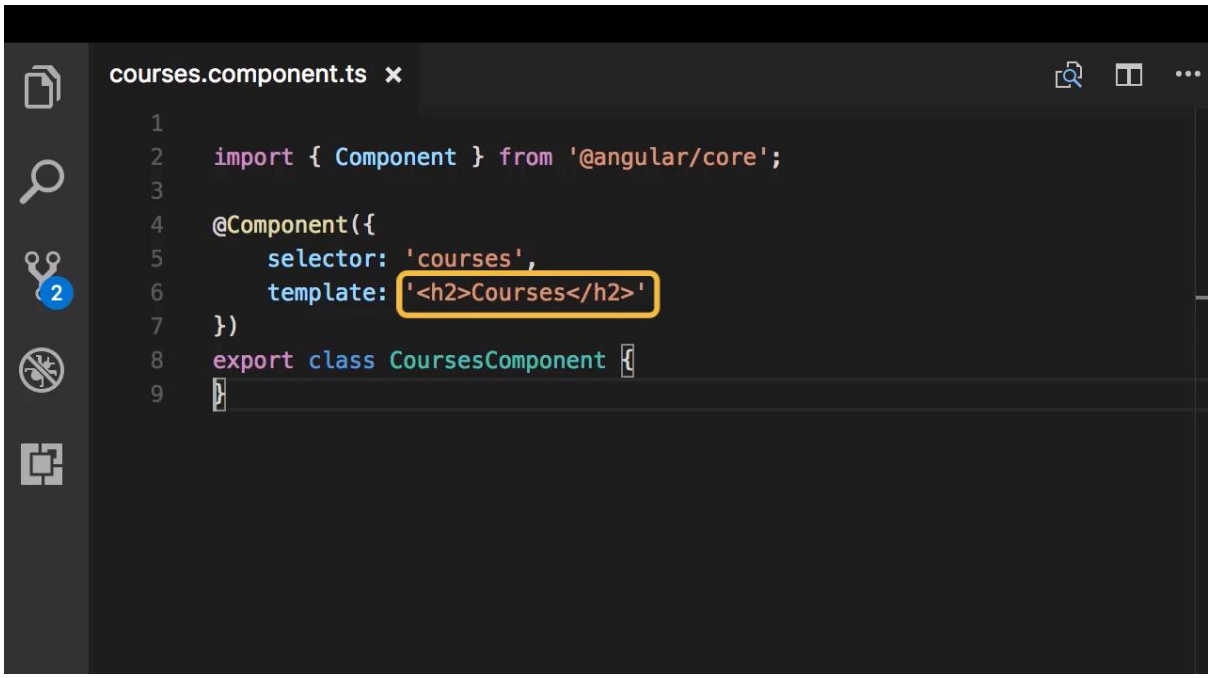

Decorator function @NgModule

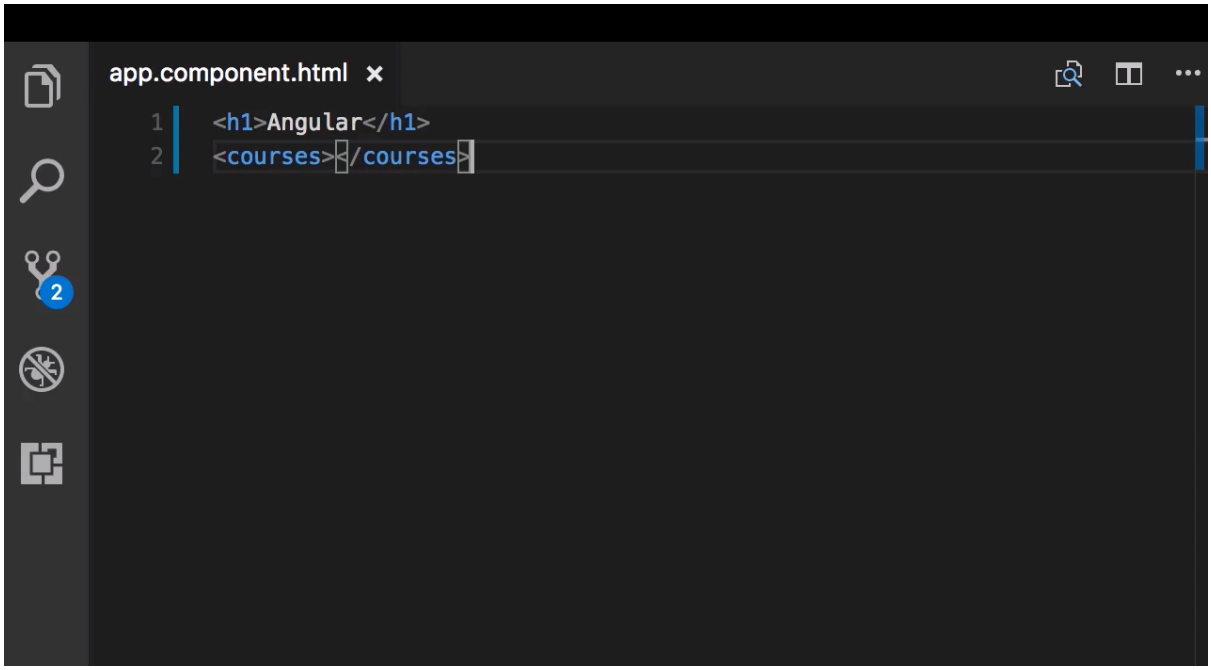


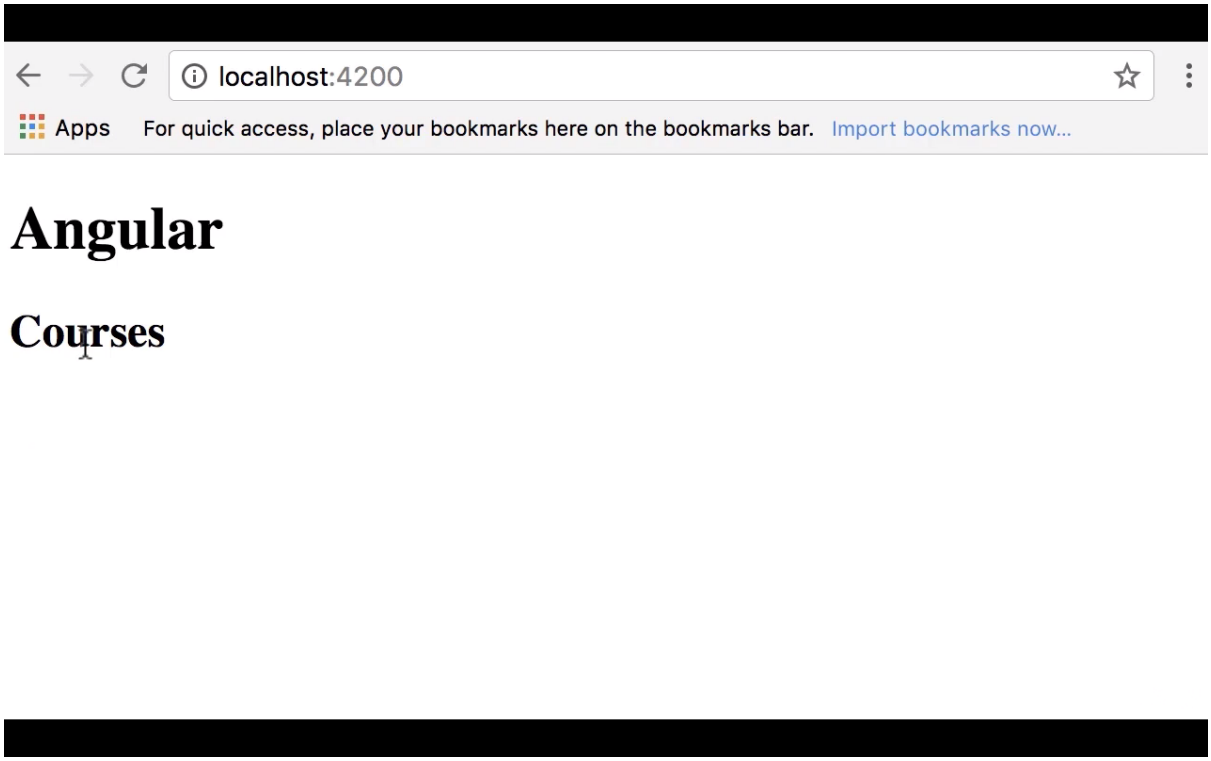

courses.component.ts
app.module.ts
import { CoursesComponent } from './courses.component';
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
@NgModule({
declarations: [
AppComponent,
CoursesComponent
],
imports: [
BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})
Auto-import Plugin

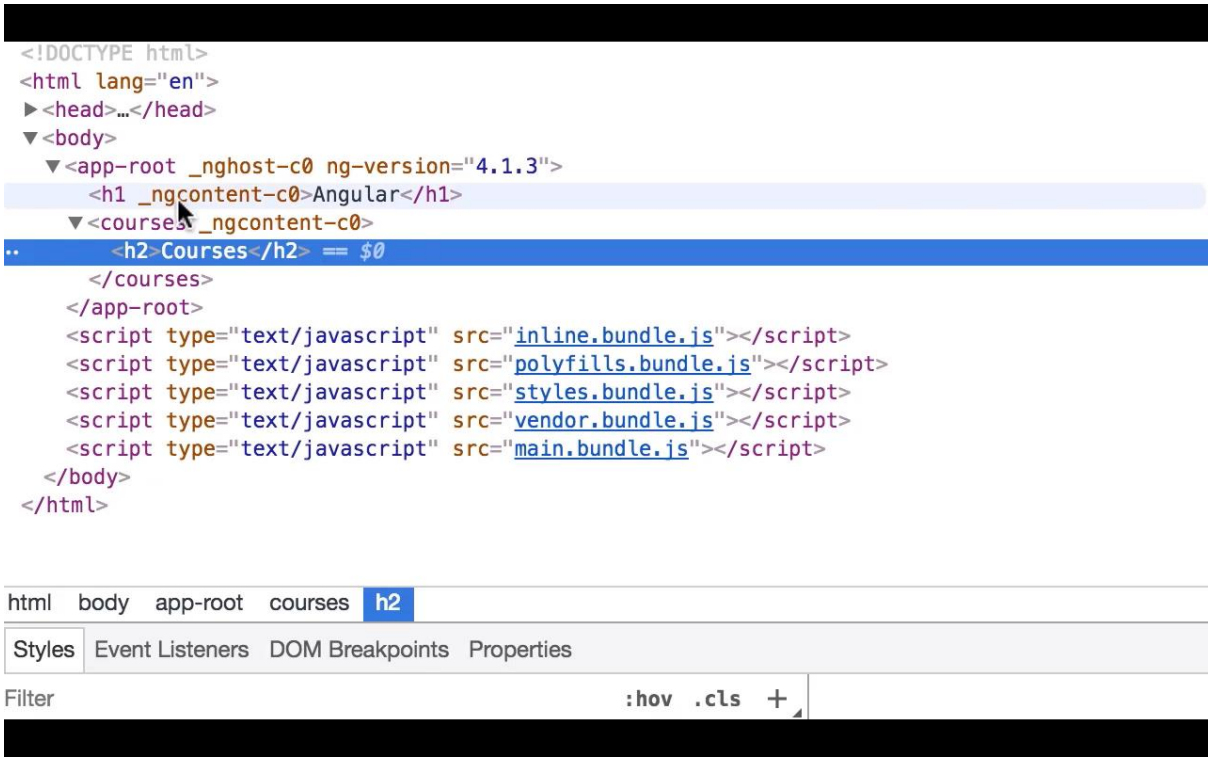

EXTENSIONS
auto import
Auto Import 1.2.2
Automatically finds, parses a...
steoates
Auto Import - ES6 & TS 1.3.3
Automatically finds, parses a...
Martin Oppitz
Install
Auto Close Tag 0.4.2
Automatically add HTML/XML...
Jun Han
Install
Auto Rename Tag 0.0.12
Auto rename paired HTML/X...
Jun Han
Install
Auto-Open Markdown Prev...
app.module.ts
import { CoursesComponent } from './courses
import { BrowserModule } from '@angular/pla
import { NgModule } from '@angular/core';
import { AppComponent } from './app.compone
@NgModule({
declarations: [
AppComponent,
CoursesComponent
],
imports: [
BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})

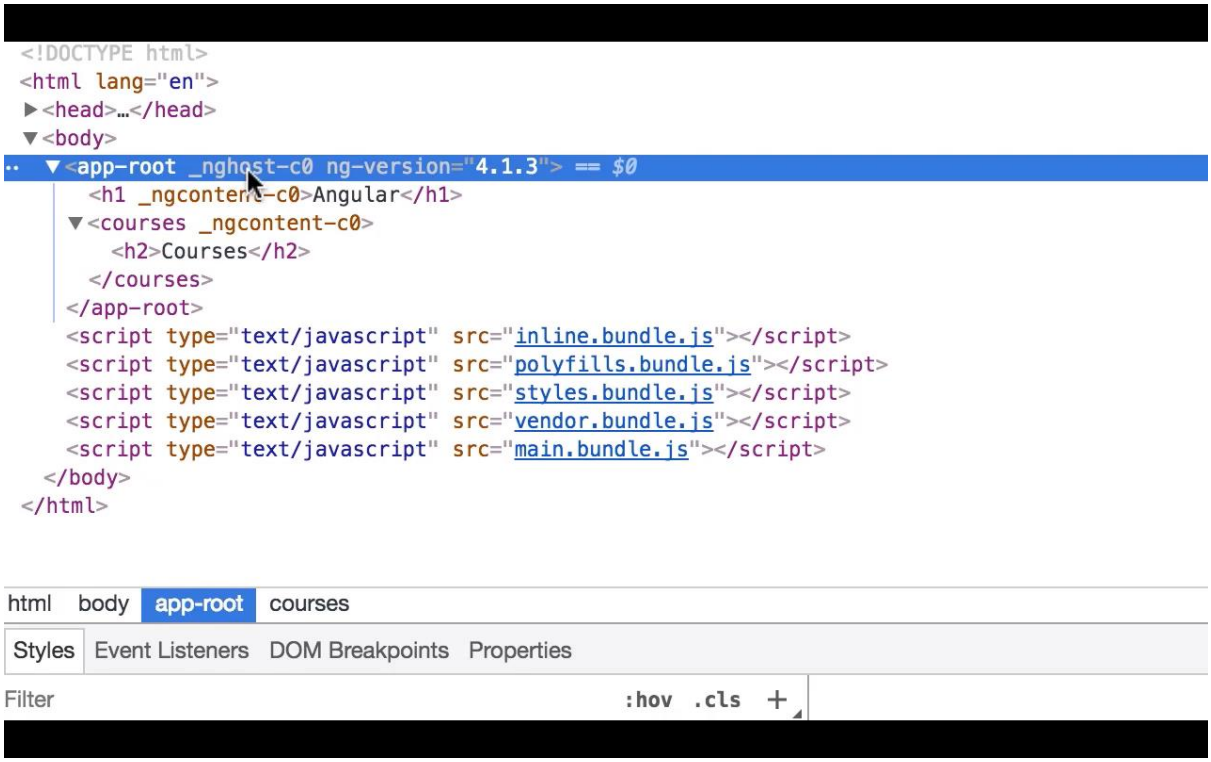

courses.component.ts
1
2 import { Component } from '@angular/core';
3
4 @Component({
5 selector: 'courses',
6 template: '<h2>Courses</h2>'
7 })
8 export class CoursesComponent {
9 }

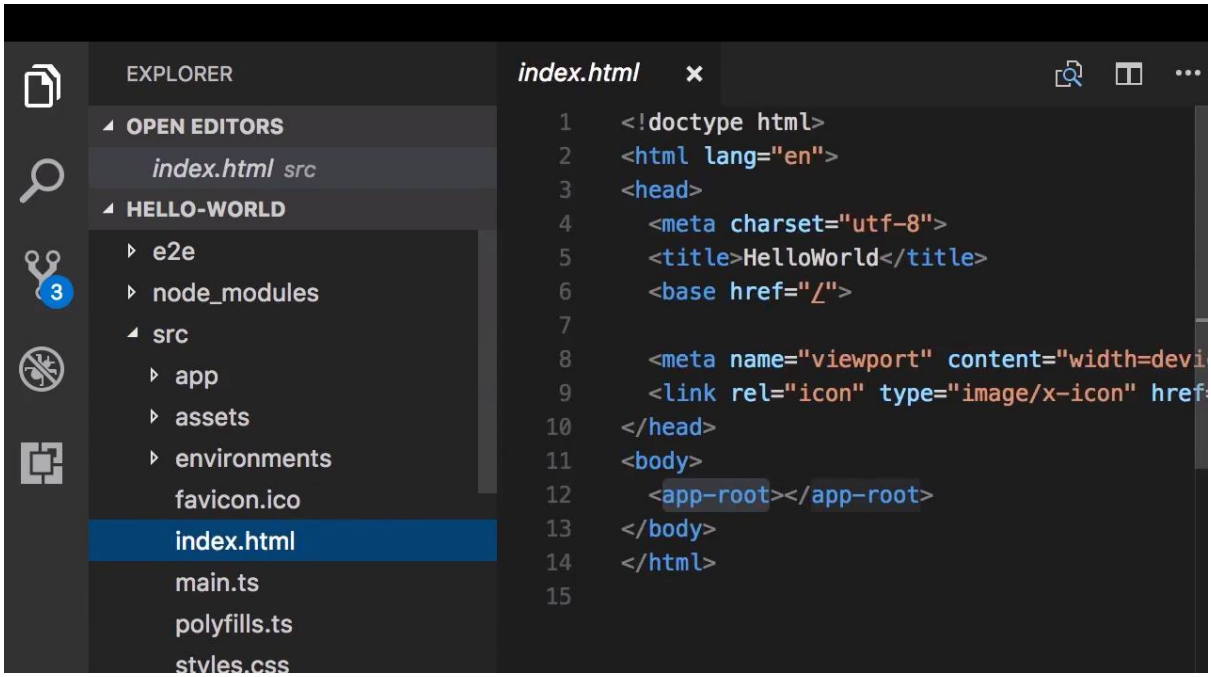

courses.component.ts
1
2 import { Component } from '@angular/core';
3
4 @Component({
5 selector: 'courses',
6 template: '<h2>Courses</h2>'
7 })
8 export class CoursesComponent {
9 }

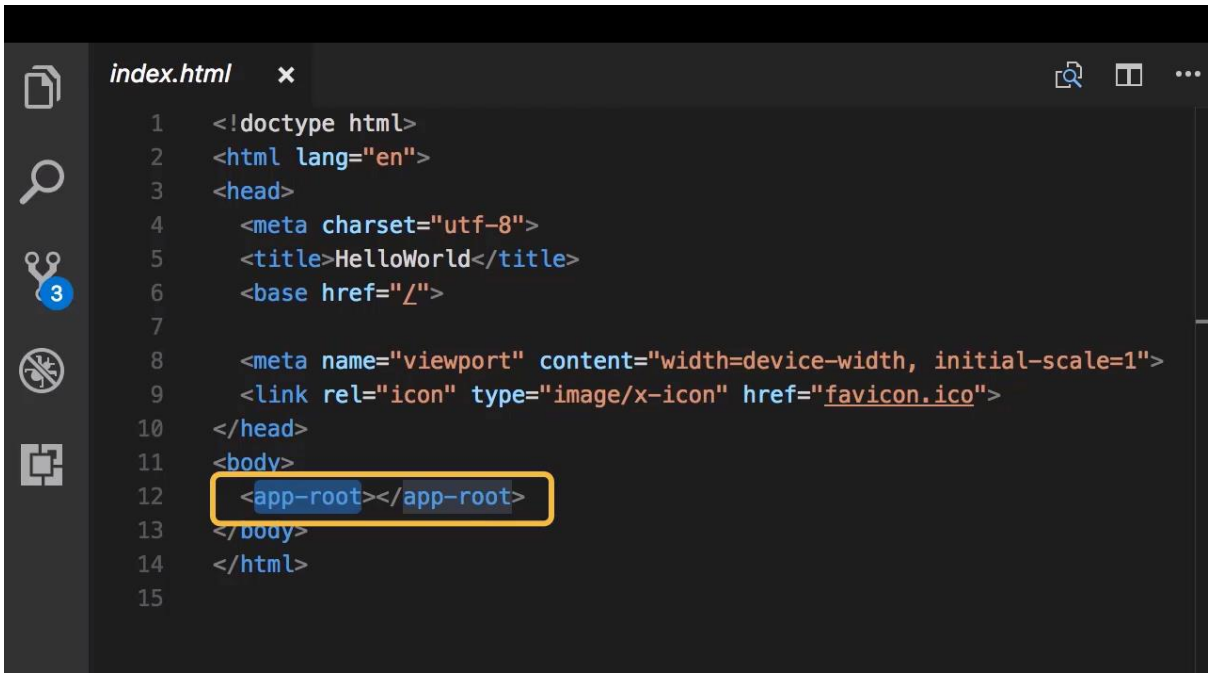

app.component.html
1 <h1>Angular</h1>
2 <courses></courses>
2

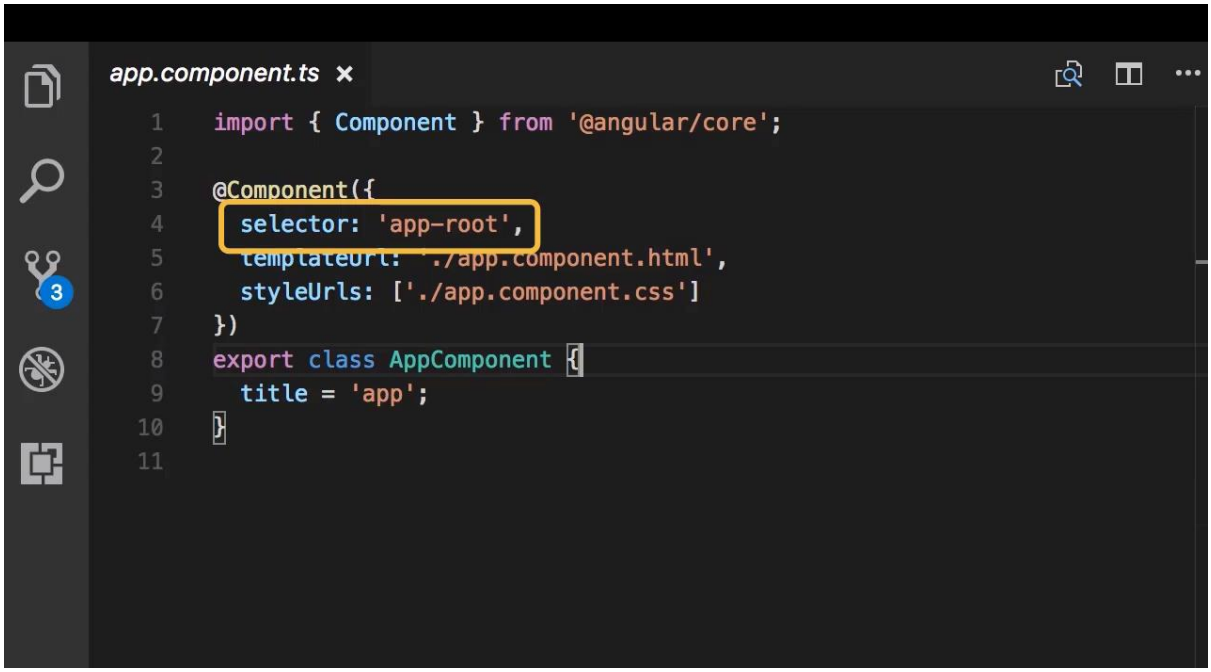

localhost:4200
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Angular
Courses


<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body>
▼<app-root _nghost-c0 ng-version="4.1.3">
<h1 _ngcontent-c0>Angular</h1>
▼<courses _ngcontent-c0>
<h2>Courses</h2> == $0
</courses>
</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="styles.bundle.js"></script>
<script type="text/javascript" src="vendor.bundle.js"></script>
<script type="text/javascript" src="main.bundle.js"></script>
</body>
</html>
html body app-root courses h2
Styles Event Listeners DOM Breakpoints Properties
Filter :hov .cls +

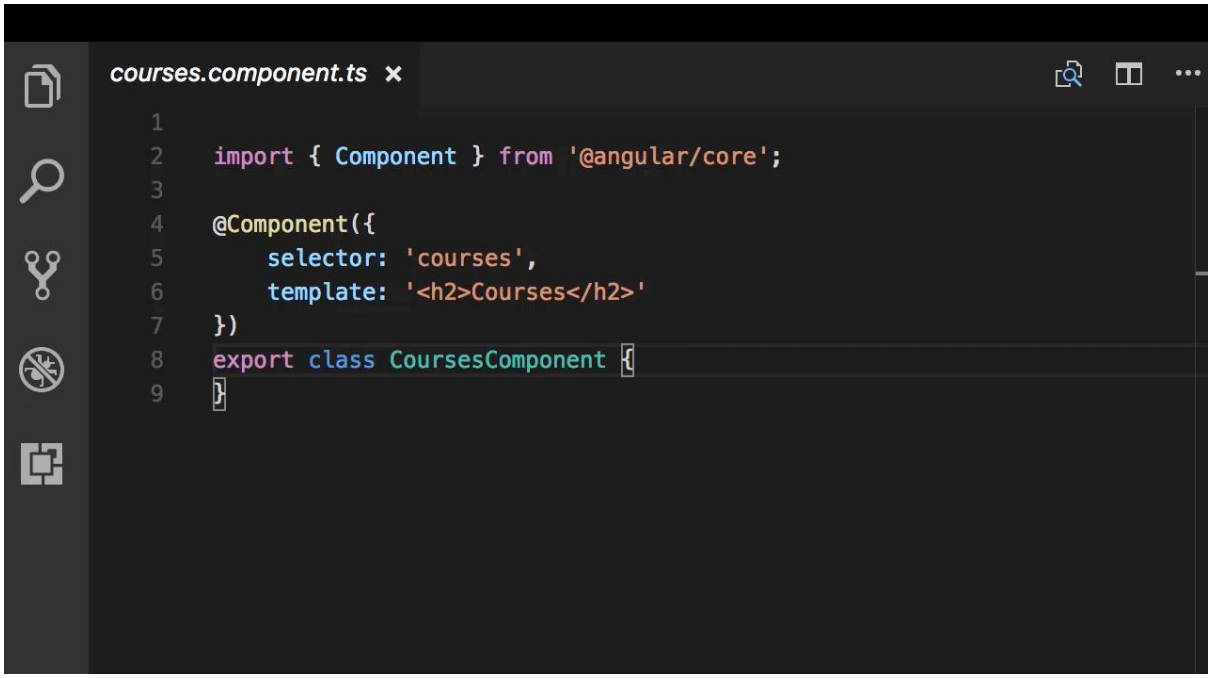

<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body>
▼<app-root _nghost-c0 ng-version="4.1.3"> == $0
<h1 _ngcontent-c0>Angular</h1>
▼<courses _ngcontent-c0>
<h2>Courses</h2>
</courses>
</app-root>
<script type="text/javascript" src="inline.bundle.js"></script>
<script type="text/javascript" src="polyfills.bundle.js"></script>
<script type="text/javascript" src="styles.bundle.js"></script>
<script type="text/javascript" src="vendor.bundle.js"></script>
<script type="text/javascript" src="main.bundle.js"></script>
</body>
</html>
html body app-root courses
Styles Event Listeners DOM Breakpoints Properties
Filter :hov .cls +

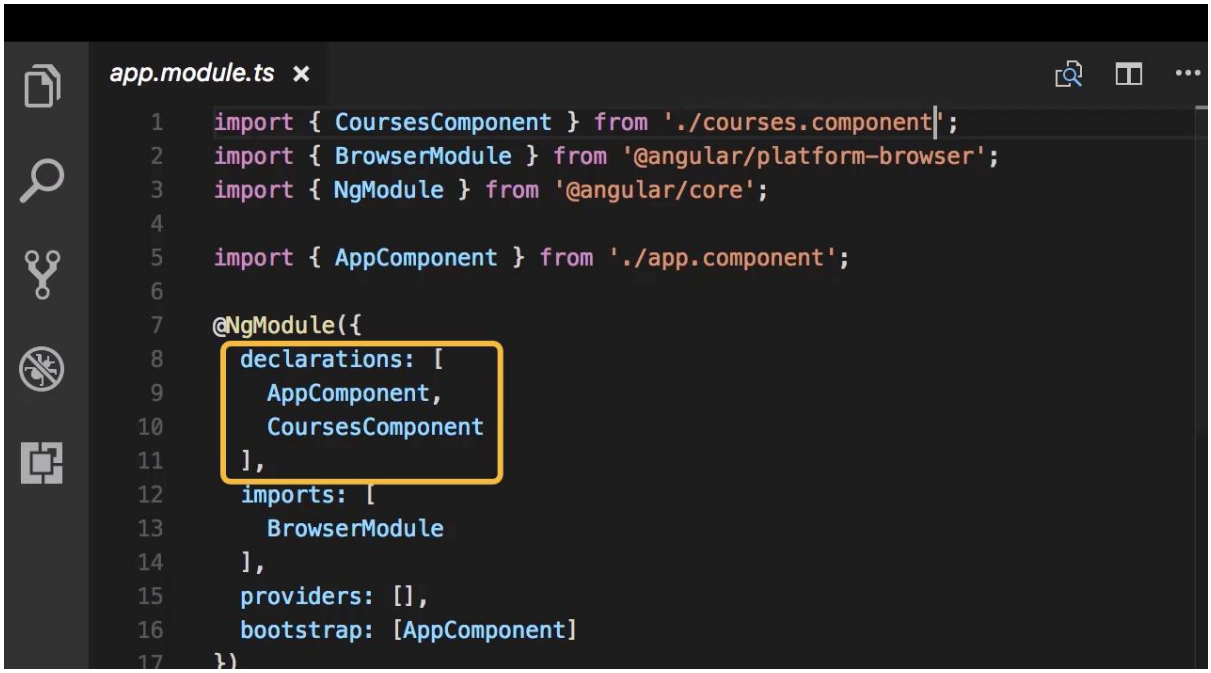

EXPLORER
OPEN EDITORS
index.html src
HELLO-WORLD
e2e
node_modules
src
app
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css
index.html
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>HelloWorld</title>
6 <base href="/">
7
8 <meta name="viewport" content="width=devi
9 <link rel="icon" type="image/x-icon" href
10 </head>
11 <body>
12 <app-root></app-root>
13 </body>
14 </html>
15

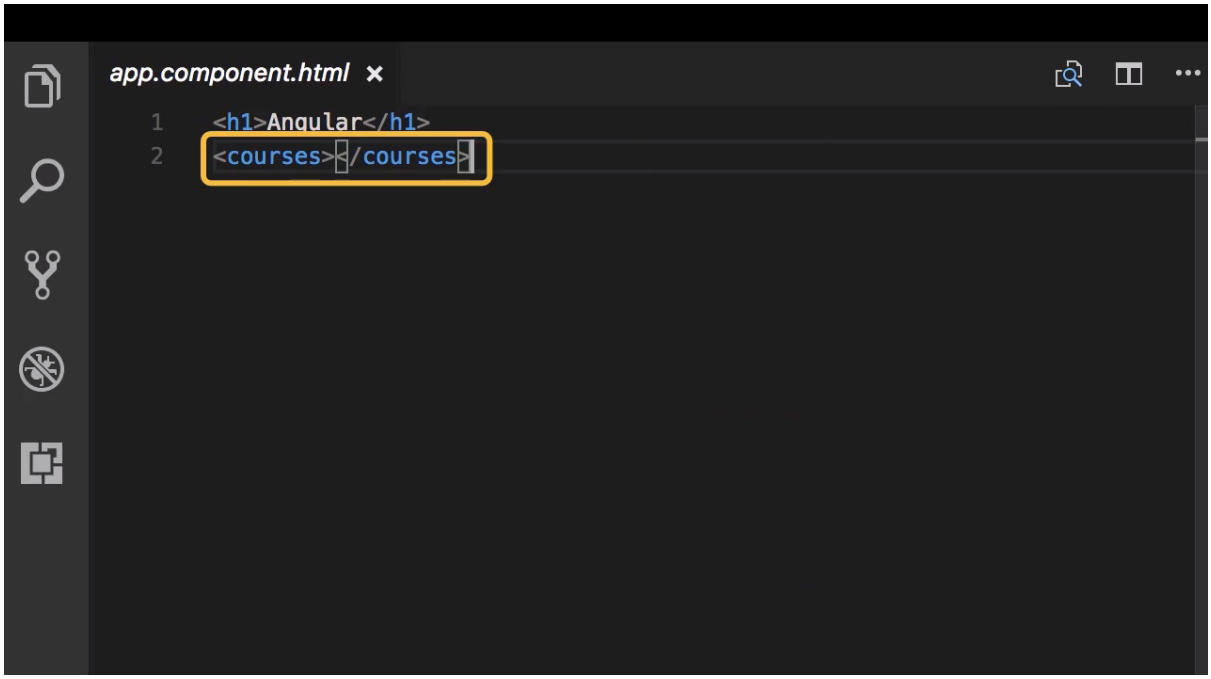

index.html
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>HelloWorld</title>
6 <base href="/">
7
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <link rel="icon" type="image/x-icon" href="favicon.ico">
10 </head>
11 <body>
12 <app-root></app-root>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


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

# Creating Components Using Angular CLI

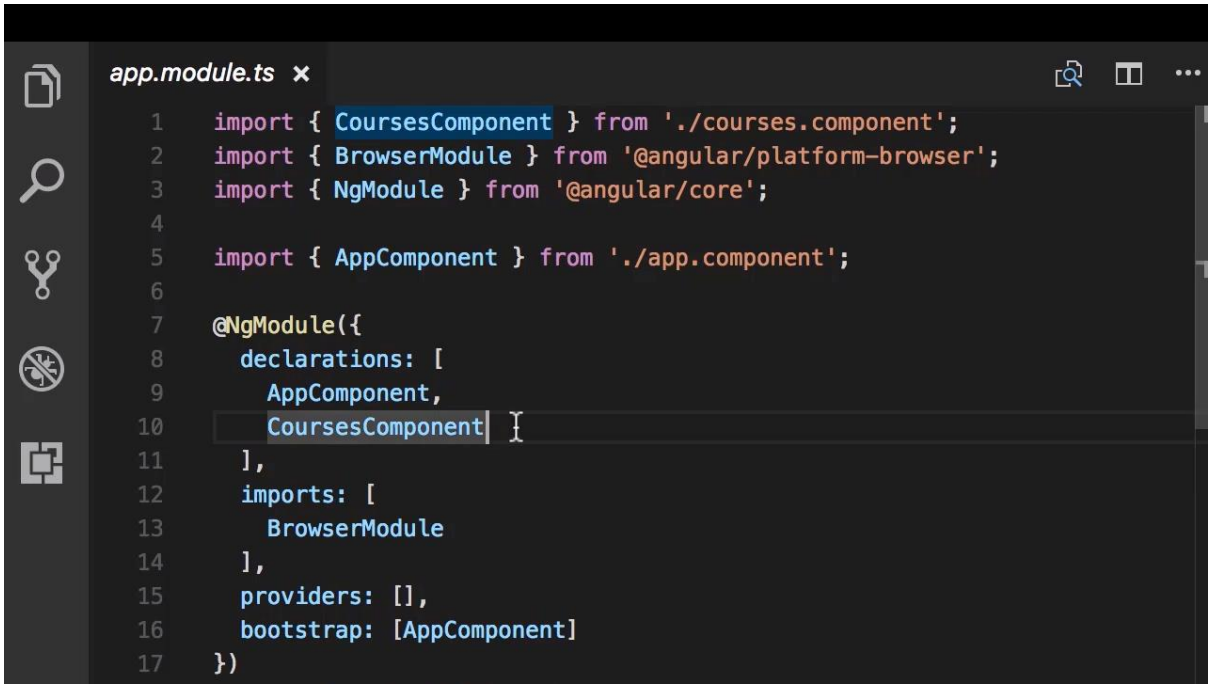


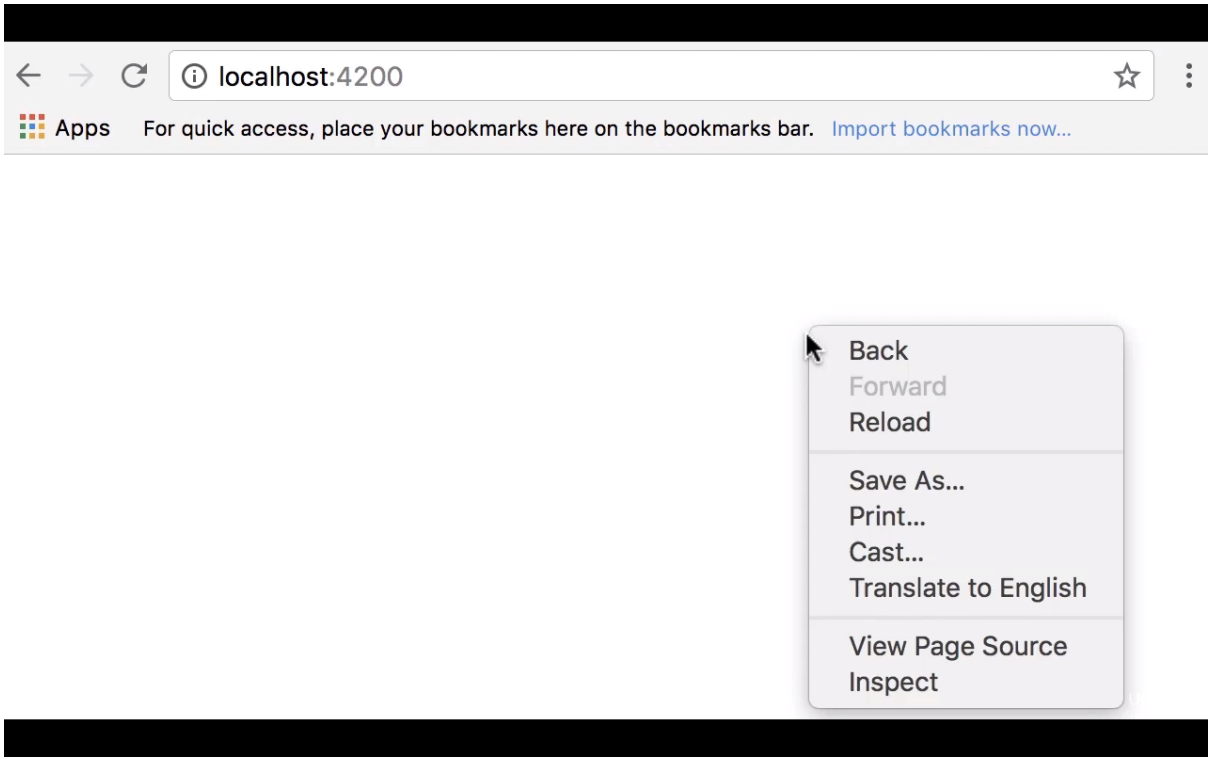


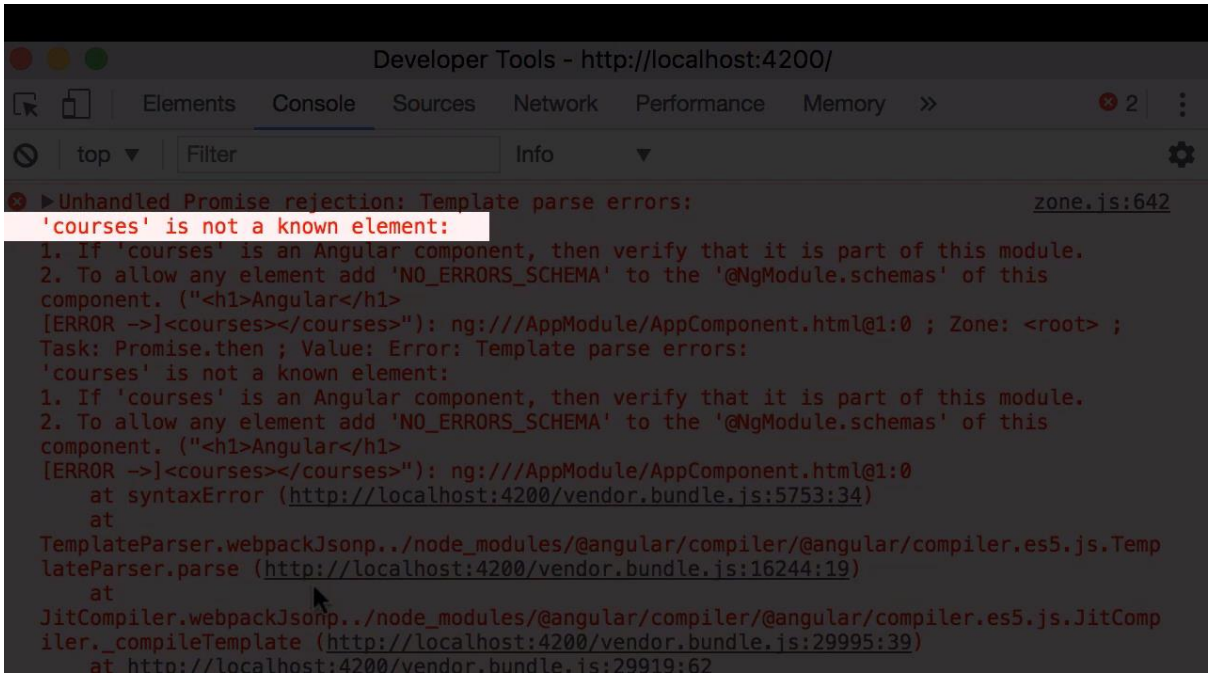




Two problems with this approach:

1. Its tedious
2. If we forget the registration step , the app is going to break.

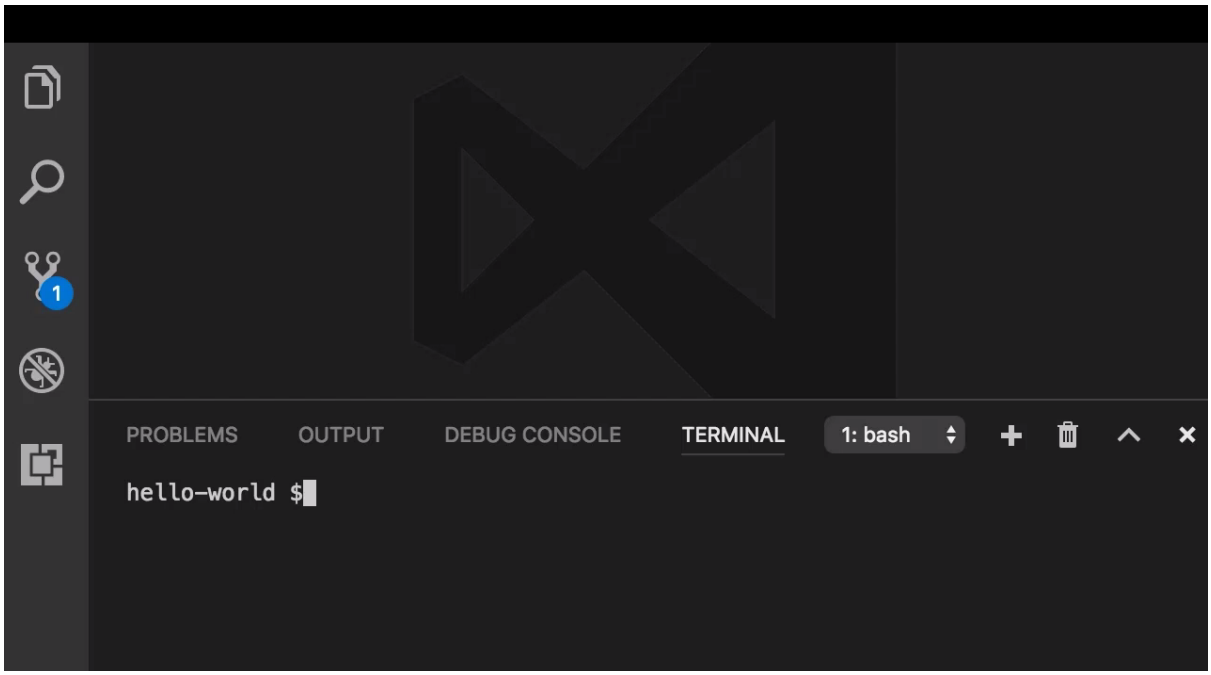

app.module.ts
1 import { CoursesComponent } from './courses.component';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { NgModule } from '@angular/core';
4
5 import { AppComponent } from './app.component';
6
7 @NgModule({
8 declarations: [
9 AppComponent,
10 CoursesComponent
11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

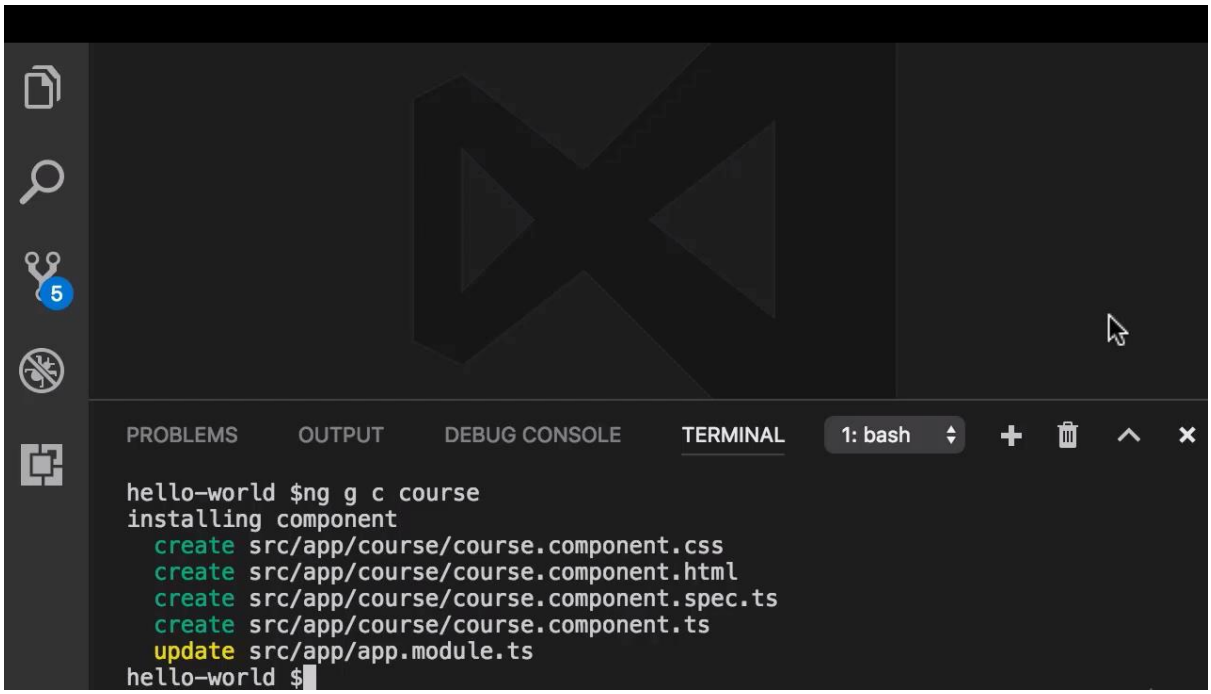

localhost:4200
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Back
Forward
Reload
Save As...
Print...
Cast...
Translate to English
View Page Source
Inspect


Developer Tools - http://localhost:4200/
Elements
Console
Sources
Network
Performance
Memory
2
top
Filter
Info
▶Unhandled Promise rejection: Template parse errors:
zone.js:642
'courses' is not a known element:
1. If 'courses' is an Angular component, then verify that it is part of this module.
2. To allow any element add 'NO_ERRORS_SCHEMA' to the '@NgModule.schemas' of this component. ("<h1>Angular</h1>
[ERROR ->]<courses></courses>"): ng:///AppModule/AppComponent.html@1:0 ; Zone: <root> ;
Task: Promise.then ; Value: Error: Template parse errors:
'courses' is not a known element:
1. If 'courses' is an Angular component, then verify that it is part of this module.
2. To allow any element add 'NO_ERRORS_SCHEMA' to the '@NgModule.schemas' of this component. ("<h1>Angular</h1>
[ERROR ->]<courses></courses>"): ng:///AppModule/AppComponent.html@1:0
at syntaxError (http://localhost:4200/vendor.bundle.js:5753:34)
at TemplateParser.webpackJsonp../node_modules/@angular/compiler/@angular/compiler.es5.js.TemplateParser.parse (http://localhost:4200/vendor.bundle.js:16244:19)
at JitCompiler.webpackJsonp../node_modules/@angular/compiler/@angular/compiler.es5.js.JitCompiler._compileTemplate (http://localhost:4200/vendor.bundle.js:29995:39)
at http://localhost:4200/vendor.bundle.js:29919:62

Now let us see a quicker and more reliable way to create a angular component

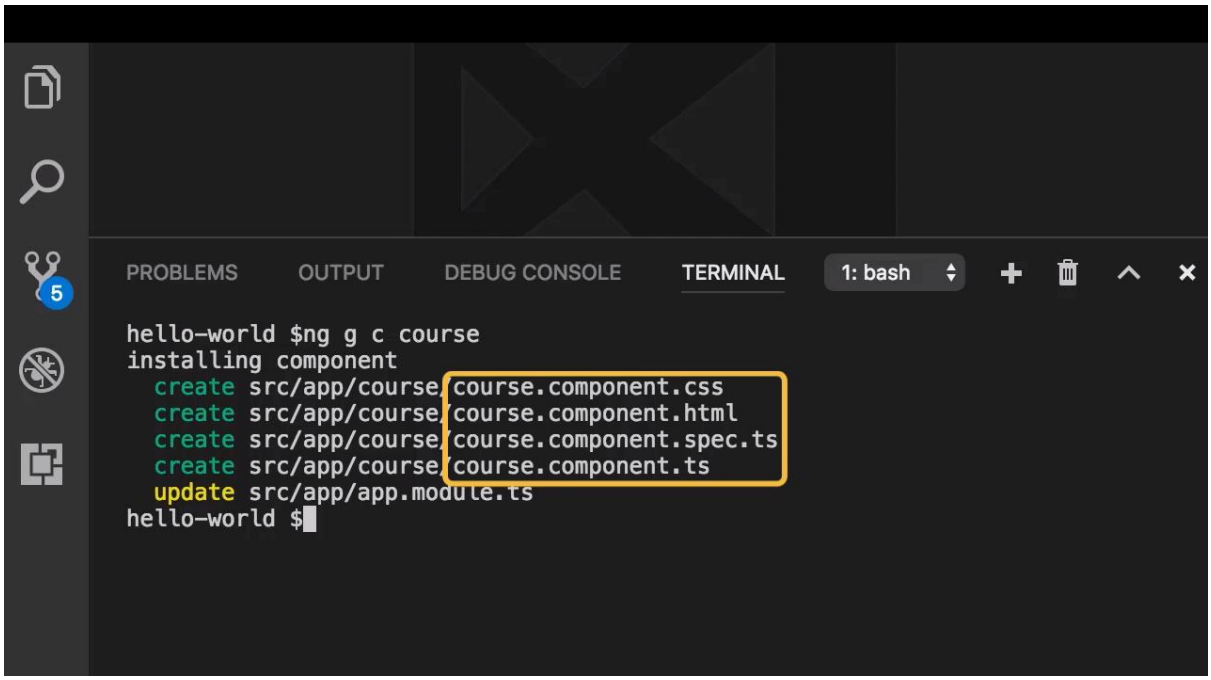
- Goto visual studio
- Press ctrl + backtick ( before the number 1 ) to run node inside vs

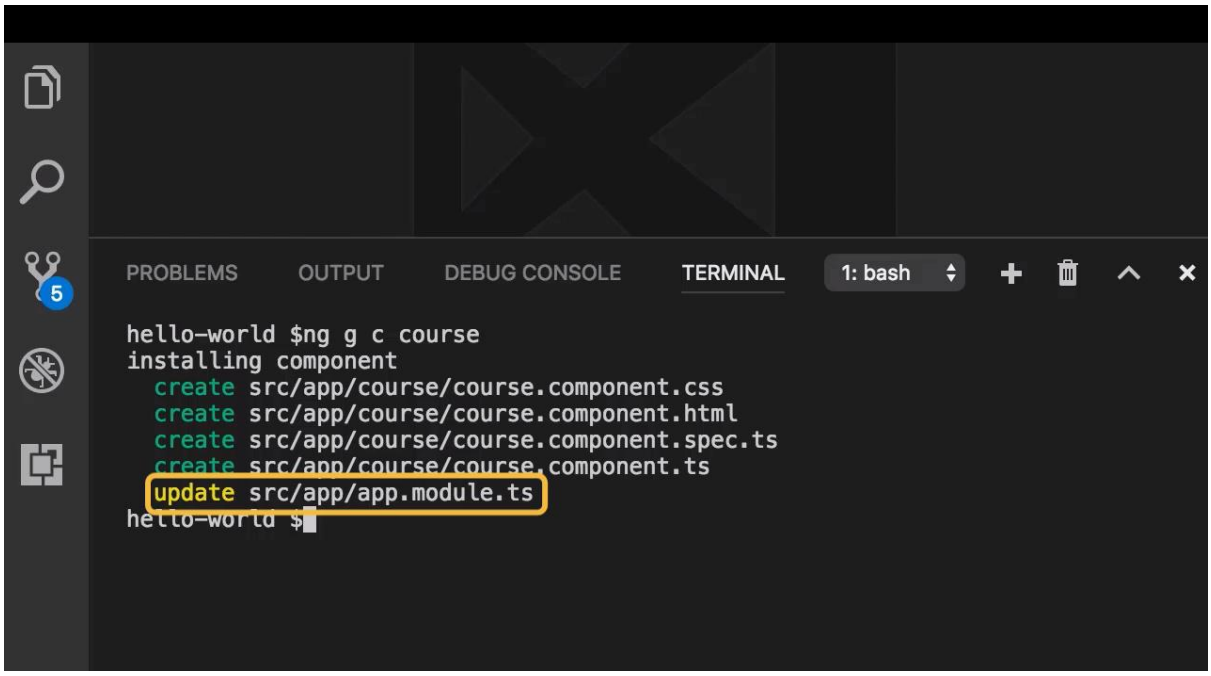


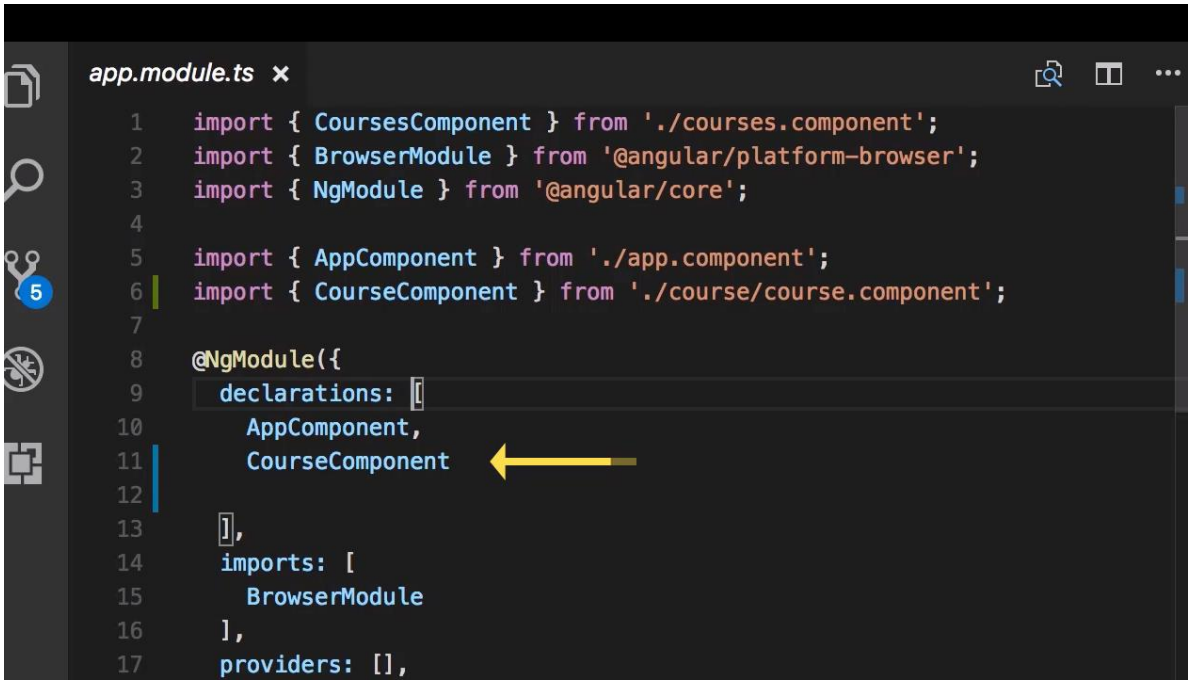



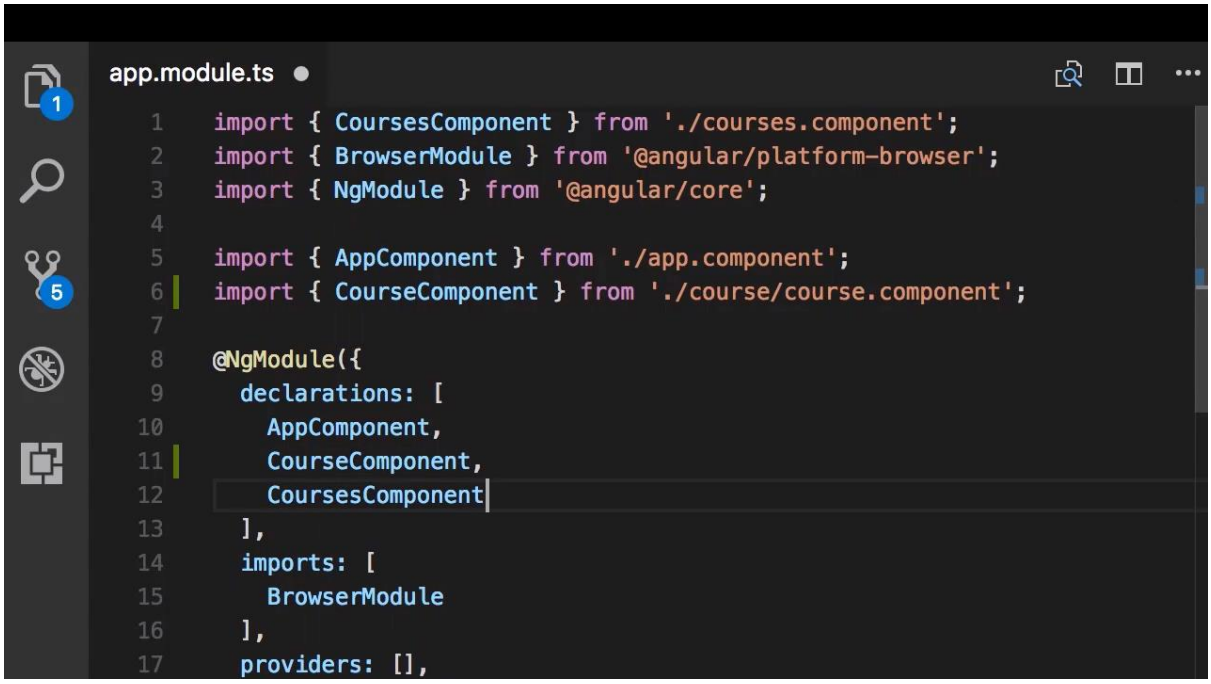

g stands for generate

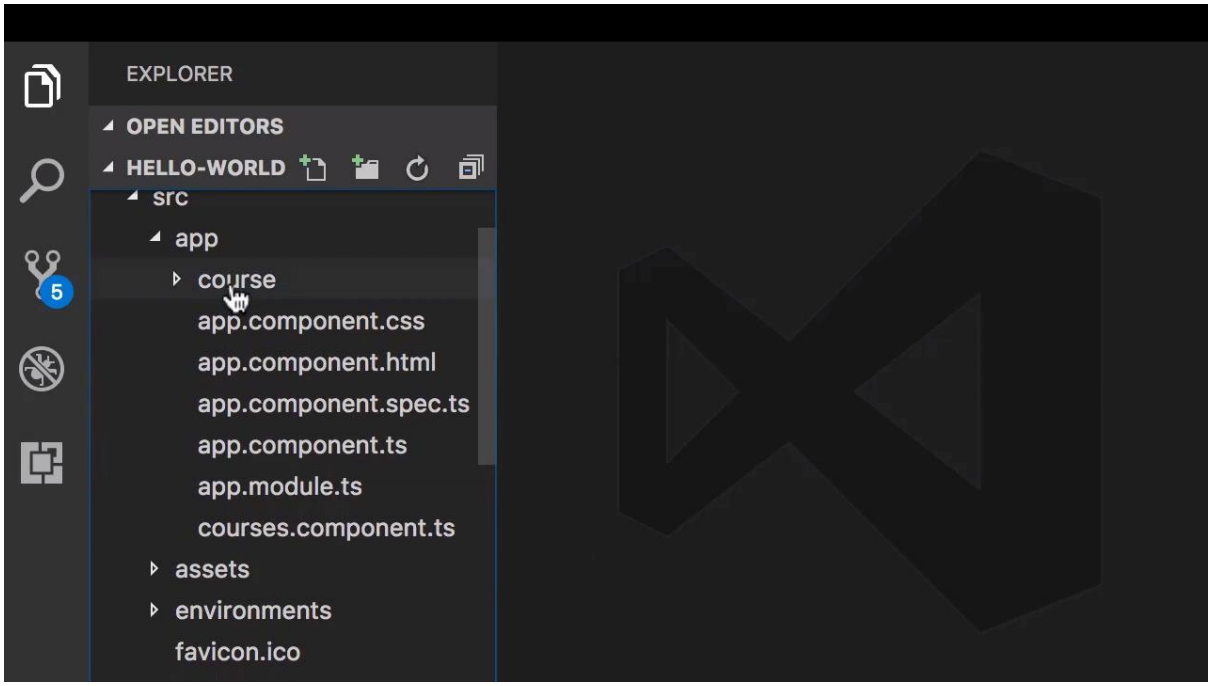
c stands for component

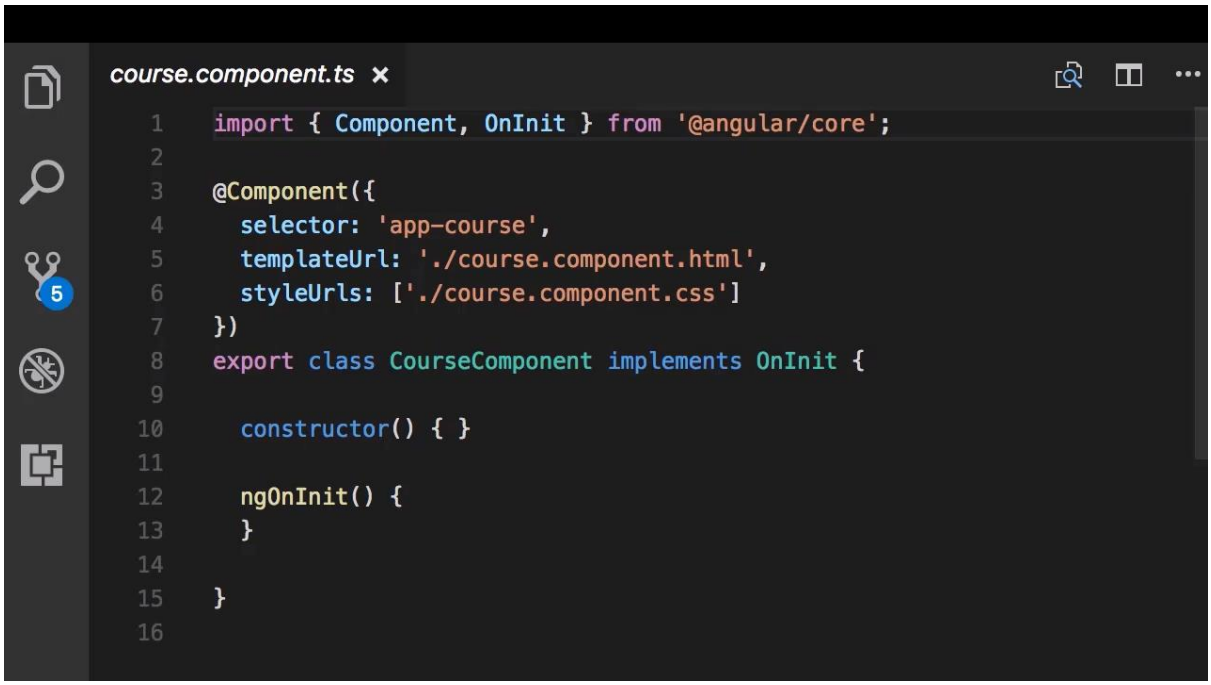


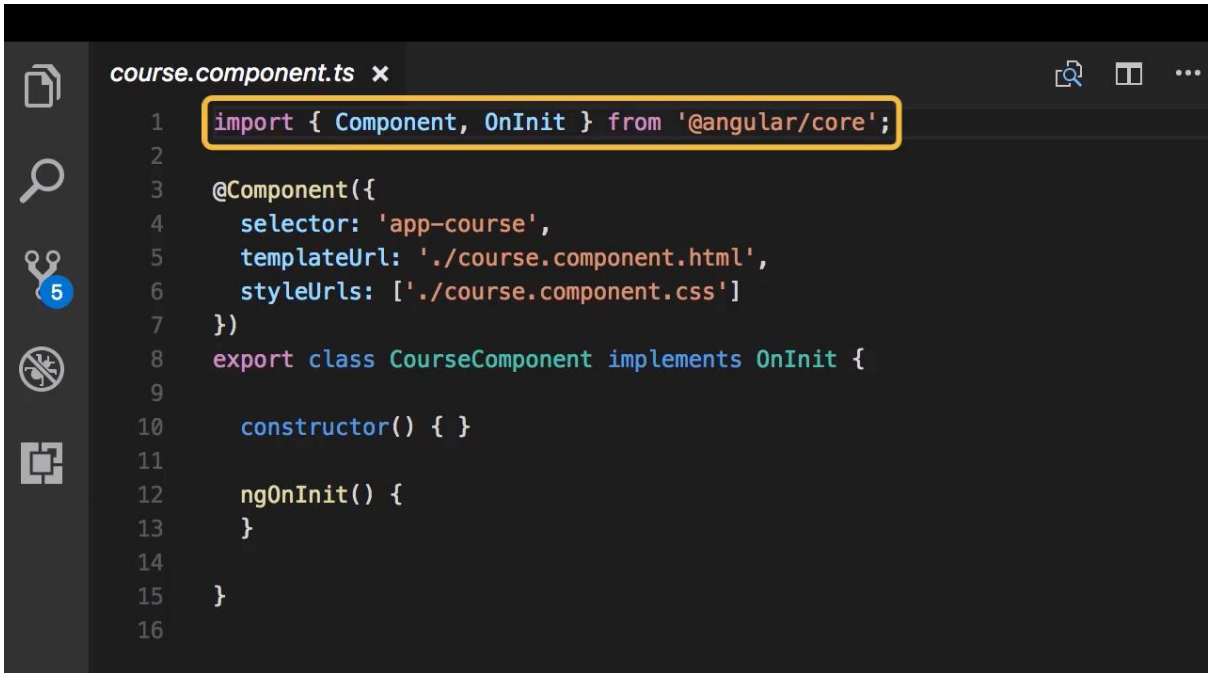



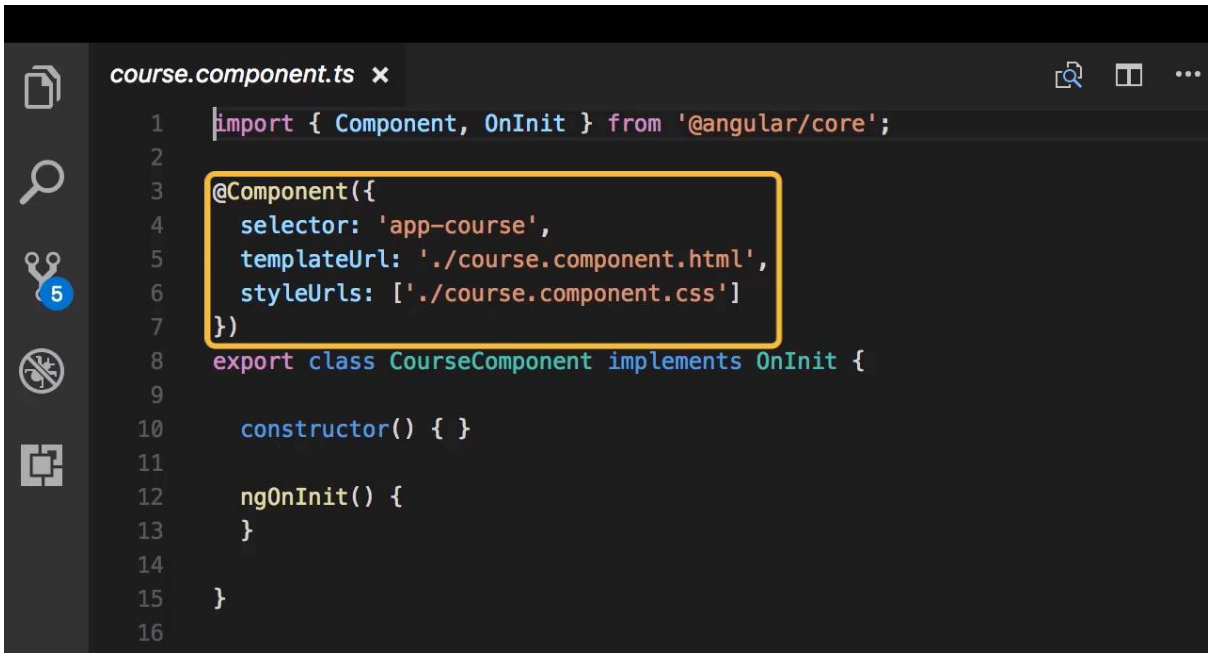

app.module.ts
import { CoursesComponent } from './courses.component';
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { CourseComponent } from './course/course.component';
@NgModule({
declarations: [
AppComponent,
CourseComponent
],
imports: [
BrowserModule
],
providers: [],


app.module.ts
import { CoursesComponent } from './courses.component';
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { CourseComponent } from './course/course.component';
@NgModule({
declarations: [
AppComponent,
CourseComponent,
CoursesComponent
],
imports: [
BrowserModule
],
providers: [],


EXPLORER
OPEN EDITORS
HELLO-WORLD
src
app
course
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts
courses.component.ts
assets
environments
favicon.ico


course.component.ts
import { Component, OnInit } from '@angular/core';
@Component({
selector: 'app-course',
templateUrl: './course.component.html',
styleUrls: ['./course.component.css']
})
export class CourseComponent implements OnInit {
constructor() { }
ngOnInit() {
}
}


course.component.ts
import { Component, OnInit } from '@angular/core';
@Component({
selector: 'app-course',
templateUrl: './course.component.html',
styleUrls: ['./course.component.css']
})
export class CourseComponent implements OnInit {
constructor() { }
ngOnInit() {
}
}


course.component.ts
import { Component, OnInit } from '@angular/core';
@Component({
selector: 'app-course',
templateUrl: './course.component.html',
styleUrls: ['./course.component.css']
})
export class CourseComponent implements OnInit {
constructor() { }
ngOnInit() {
}
}

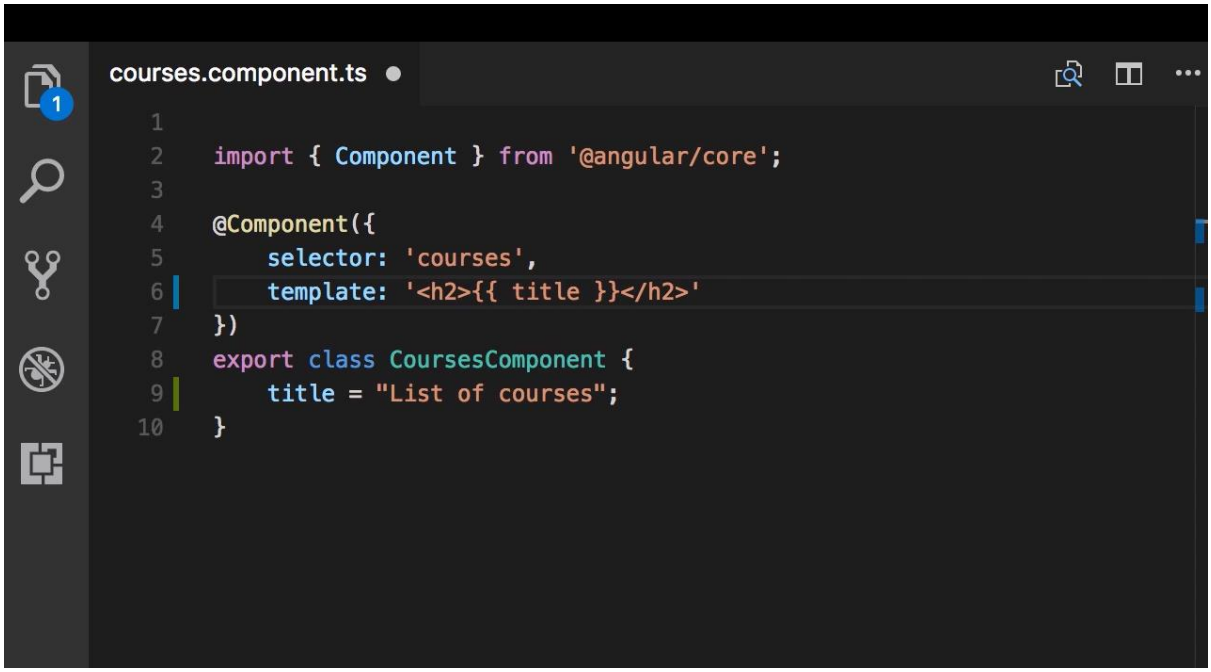
# Templates

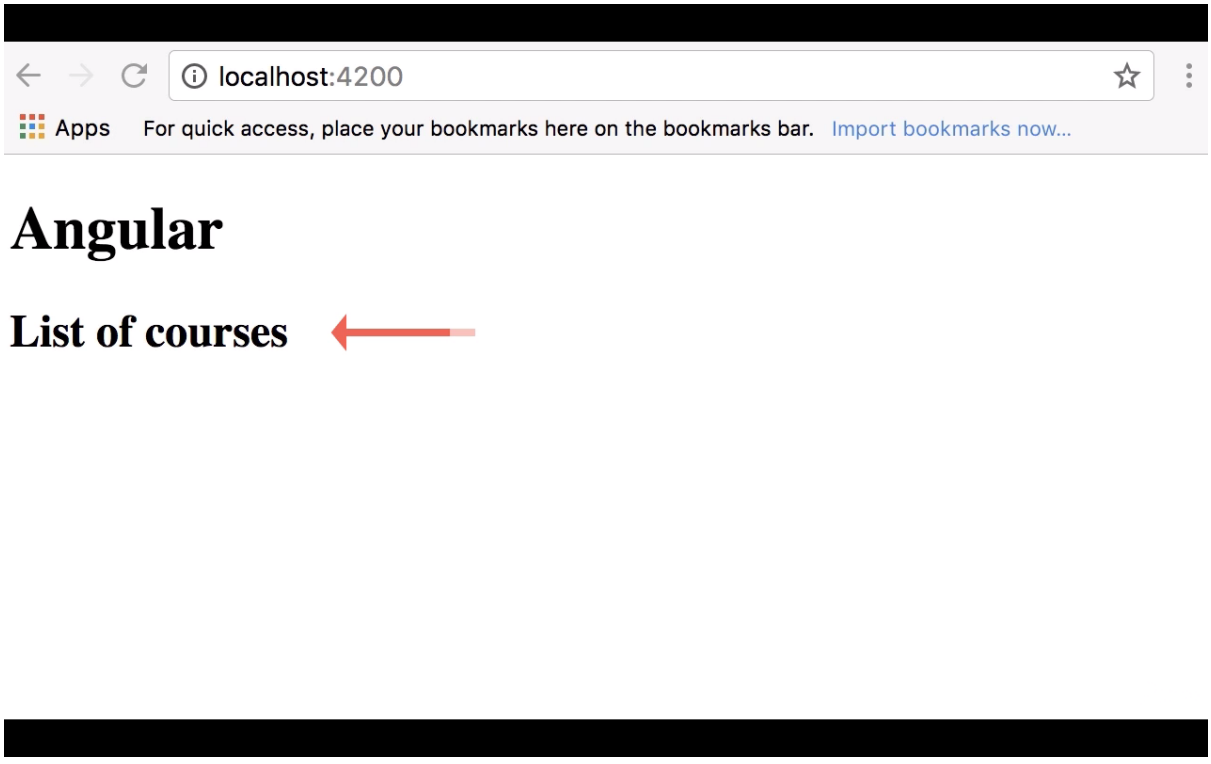
## Templates

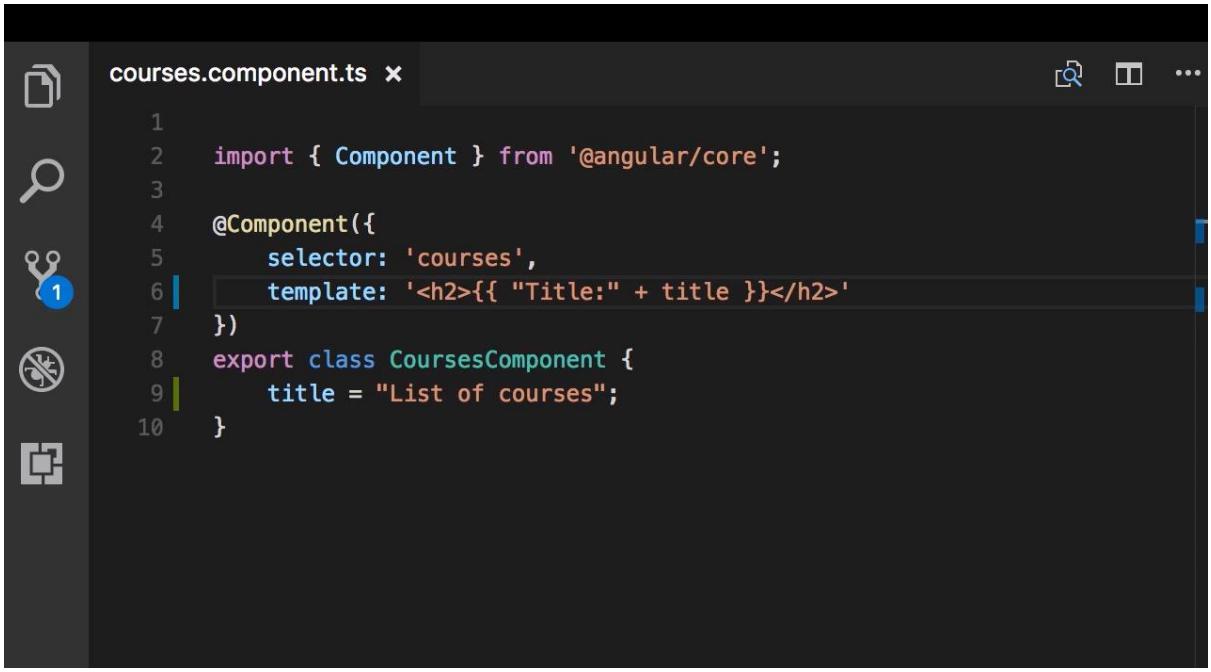
courses.component.ts

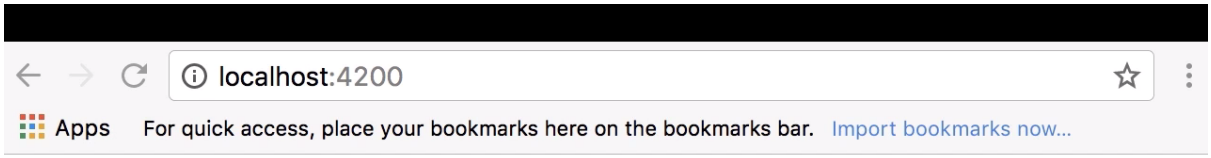
```
import { Component } from '@angular/core';

@Component({
    selector: 'courses',
    template: '<h2>Courses</h2>'
})
export class CoursesComponent {
}
```


courses.component.ts
1
import { Component } from '@angular/core';
@Component({
selector: 'courses',
template: '<h2>{{ title }}</h2>'
})
export class CoursesComponent {
title = "List of courses";
}

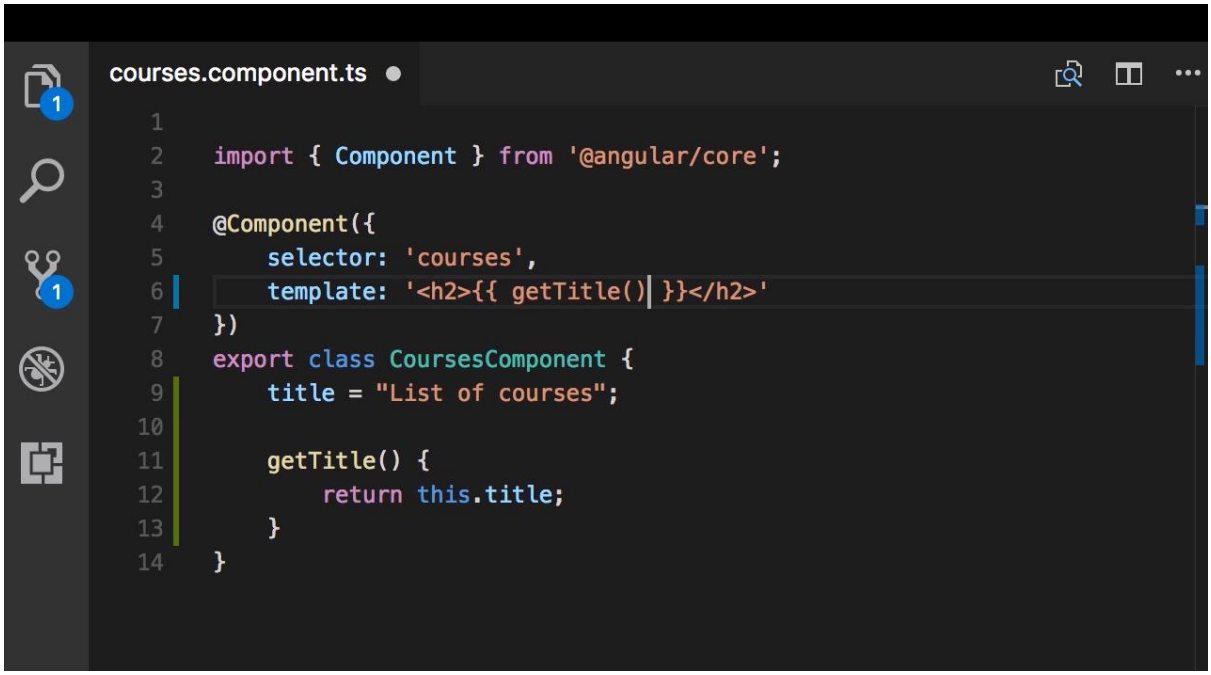

localhost:4200
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Angular
List of courses

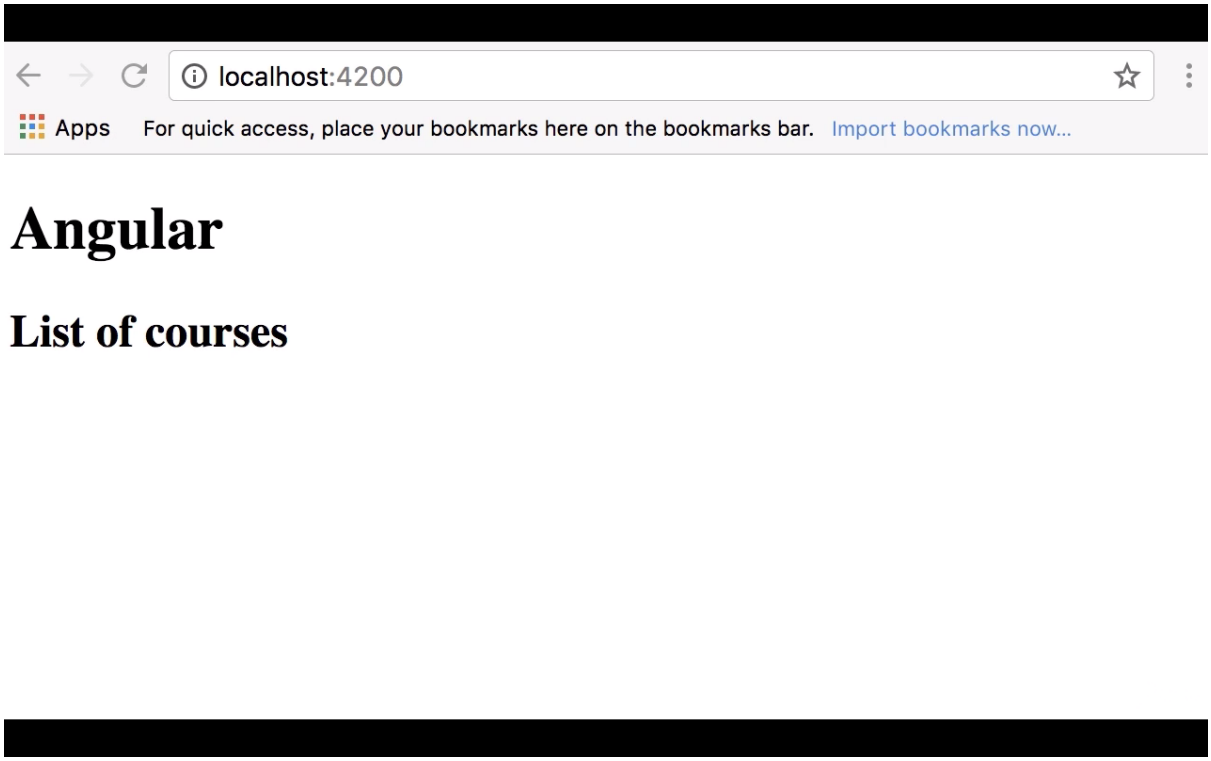


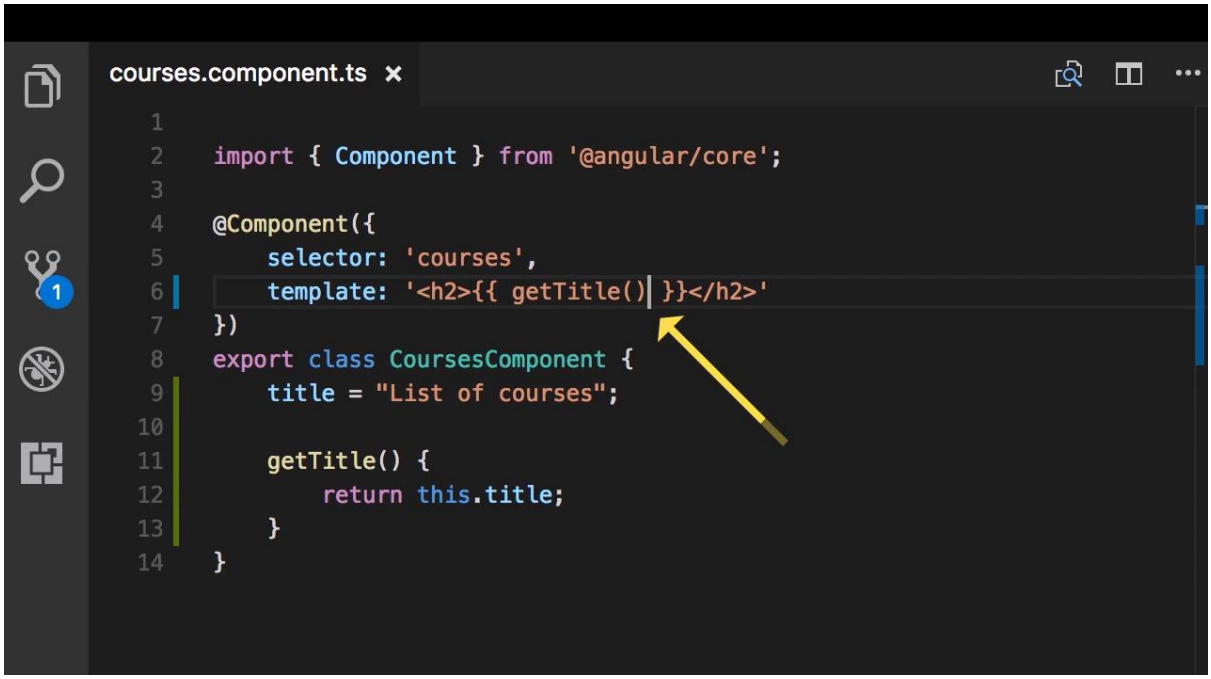



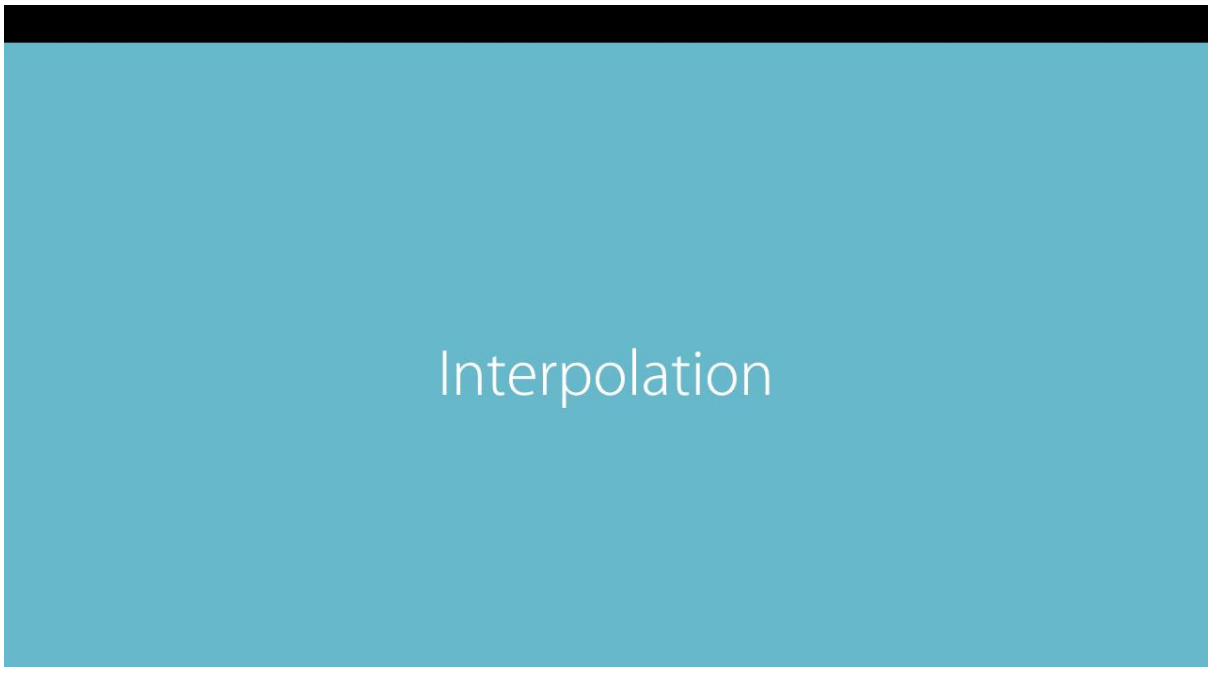

# Angular

## Title:List of courses

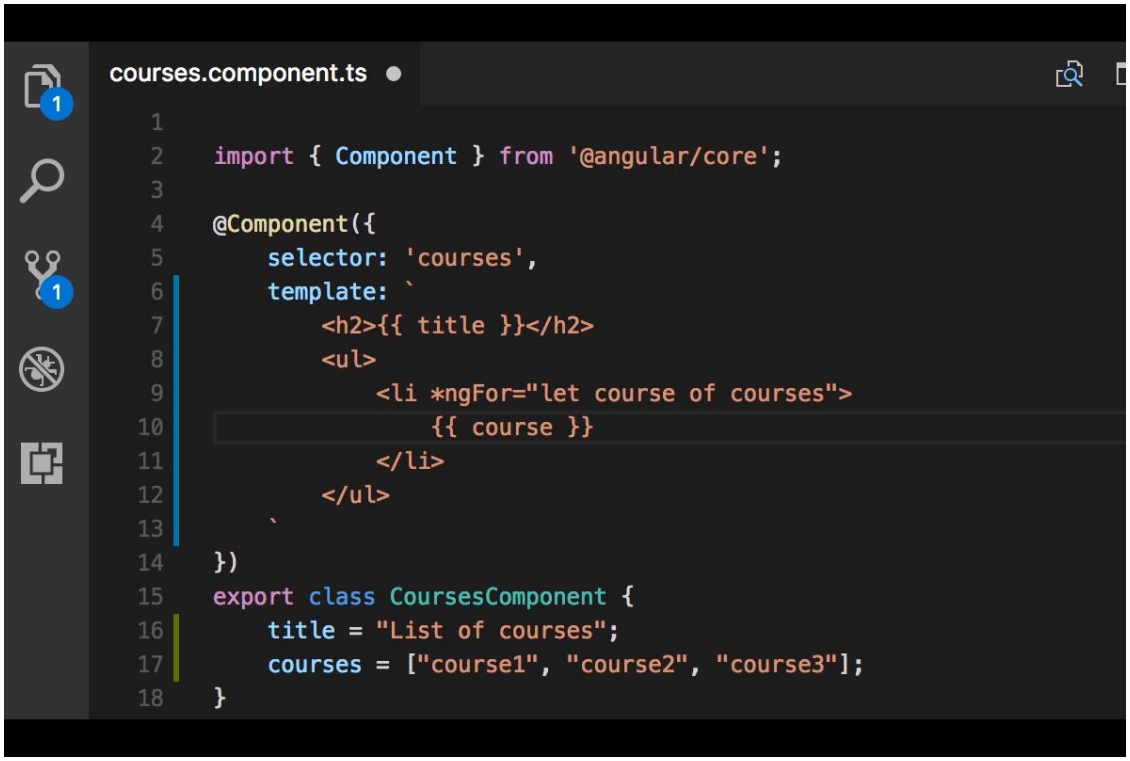

courses.component.ts
import { Component } from '@angular/core';
@Component({
selector: 'courses',
template: '<h2>{{ getTitle() }}</h2>'
})
export class CoursesComponent {
title = "List of courses";
getTitle() {
return this.title;
}
}

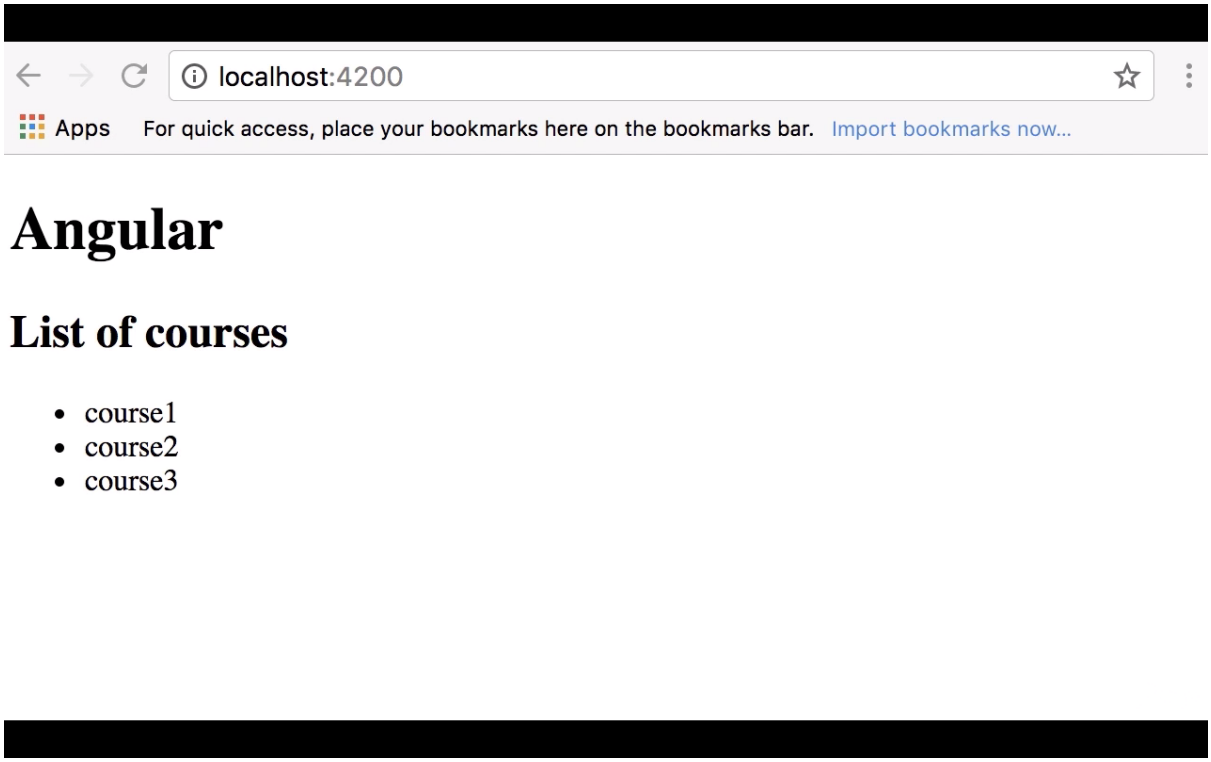

localhost:4200
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Angular
List of courses


courses.component.ts
1
2 import { Component } from '@angular/core';
3
4 @Component({
5 selector: 'courses',
6 template: '<h2>{{ getTitle() }}</h2>'
7 })
8 export class CoursesComponent {
9 title = "List of courses";
10
11 getTitle() {
12 return this.title;
13 }
14 }


Interpolation

# Directives


courses.component.ts
import { Component } from '@angular/core';
@Component({
selector: 'courses',
template: `
<h2>{{ title }}</h2>
<ul>
<li *ngFor="let course of courses">
{{ course }}
</li>
</ul>
`
})
export class CoursesComponent {
title = "List of courses";
courses = ["course1", "course2", "course3"];
}

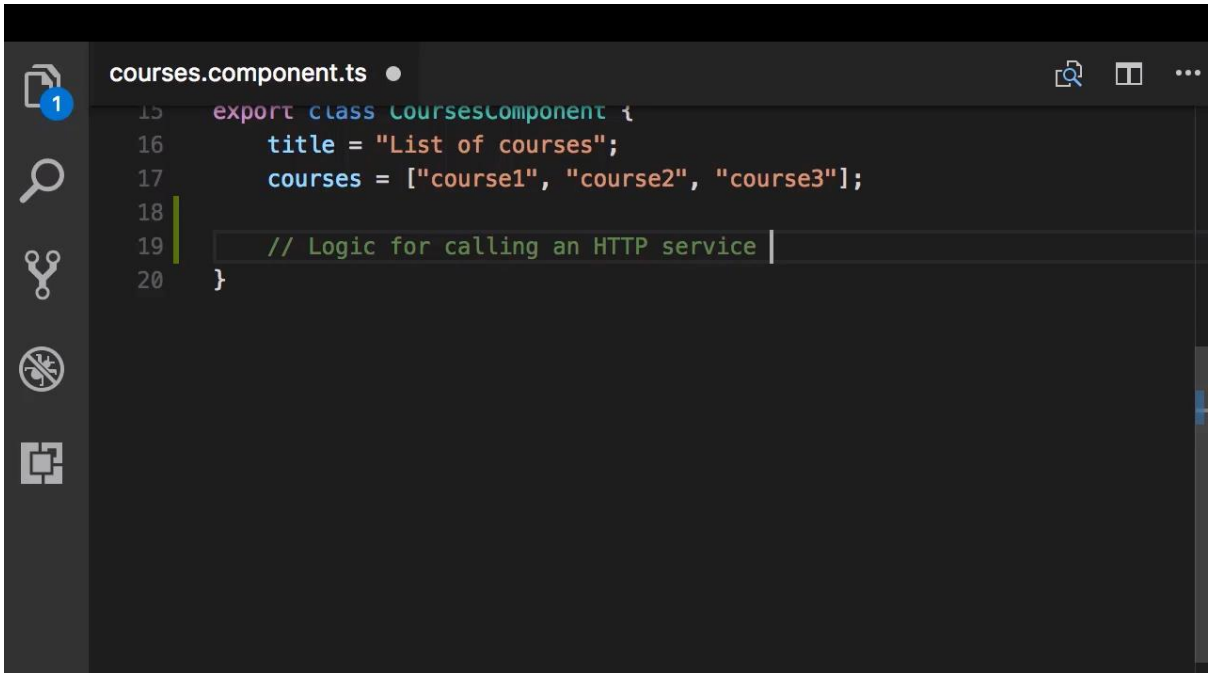

localhost:4200
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Angular
List of courses
course1
course2
course3

# Services

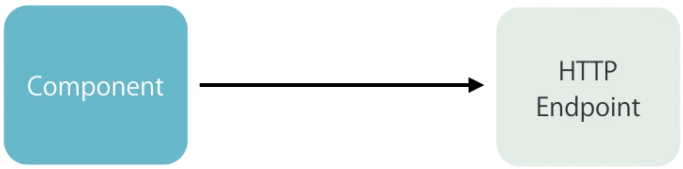
## Services

courses.component.ts

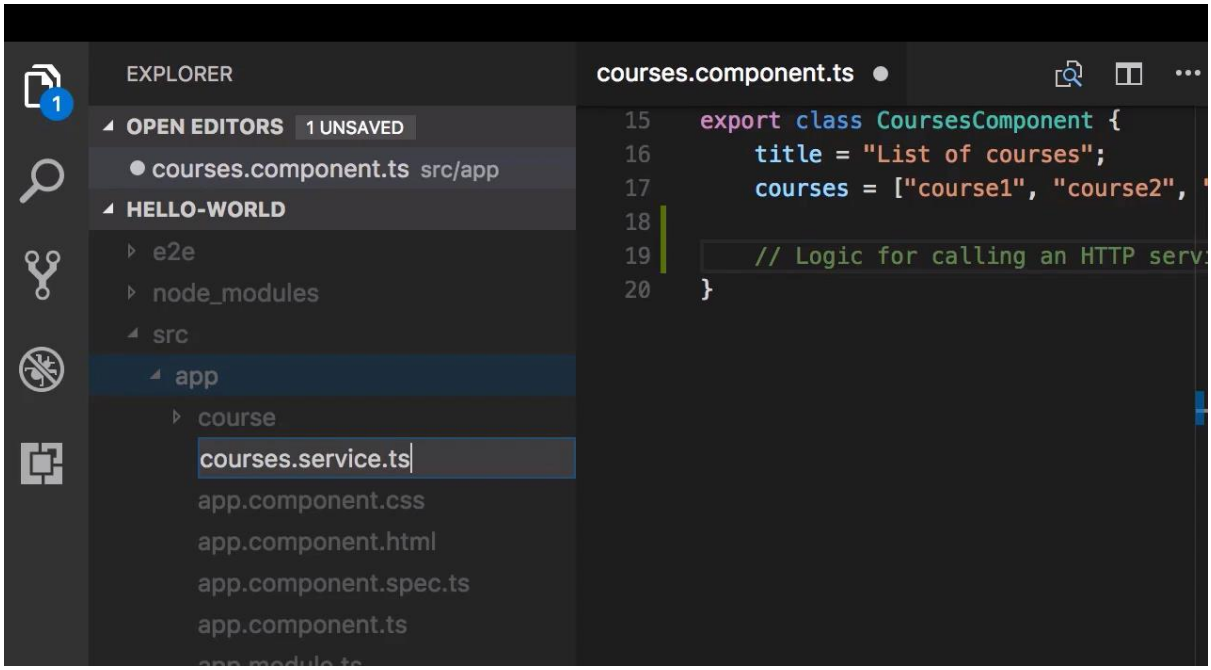
```
@Component({
    selector: 'courses',
    template: `
        <h2>{{ title }}</h2>
        <ul>
            <li *ngFor="let course of courses">
                {{ course }}
            </li>
        </ul>
    `
})
export class CoursesComponent {
    title = "List of courses";
    courses = ["course1", "course2", "course3"];
}
```



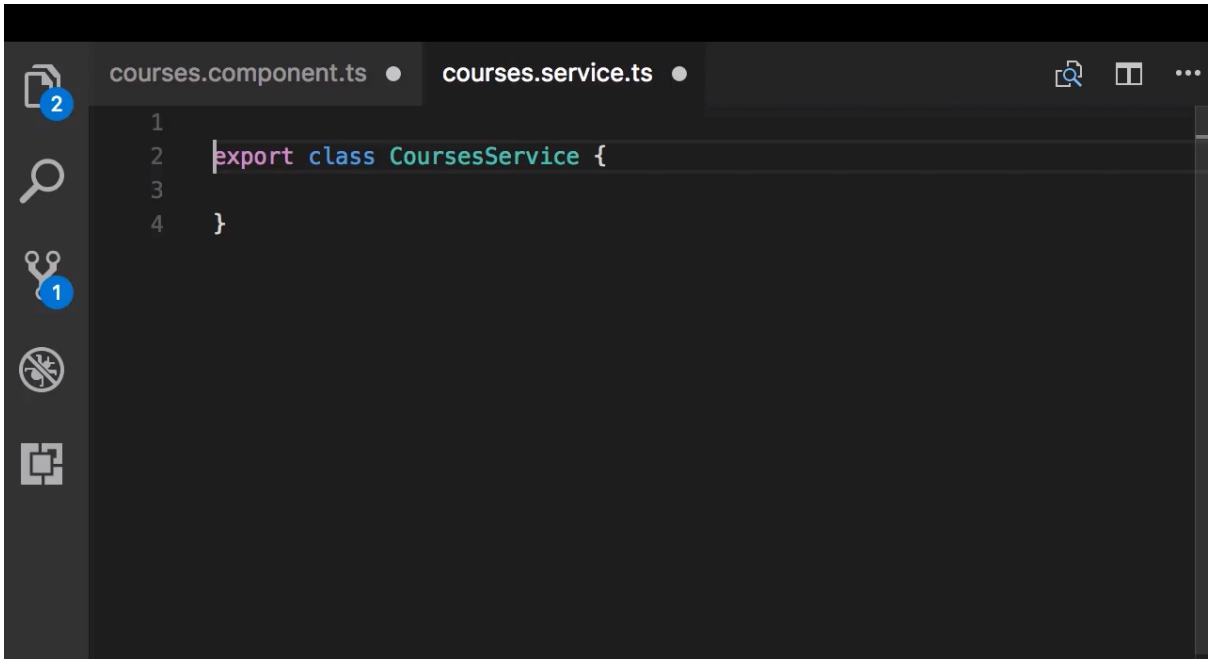

Tight coupling …unit testing is difficult

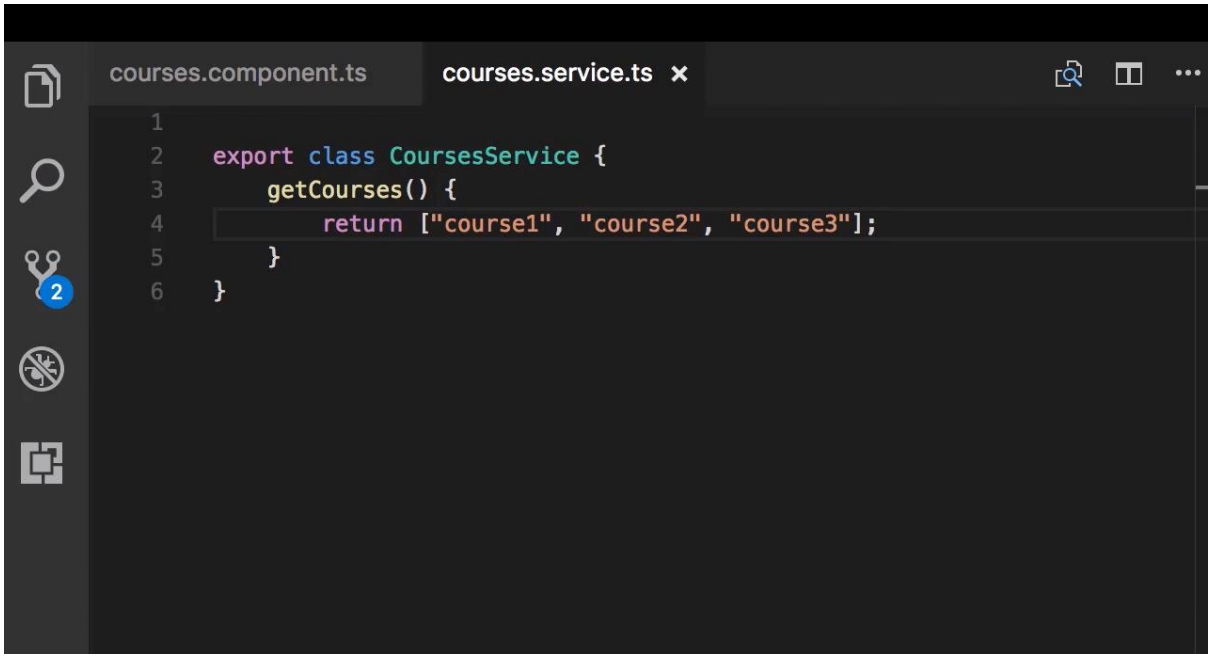



A component should not any logic other than the presentation logic



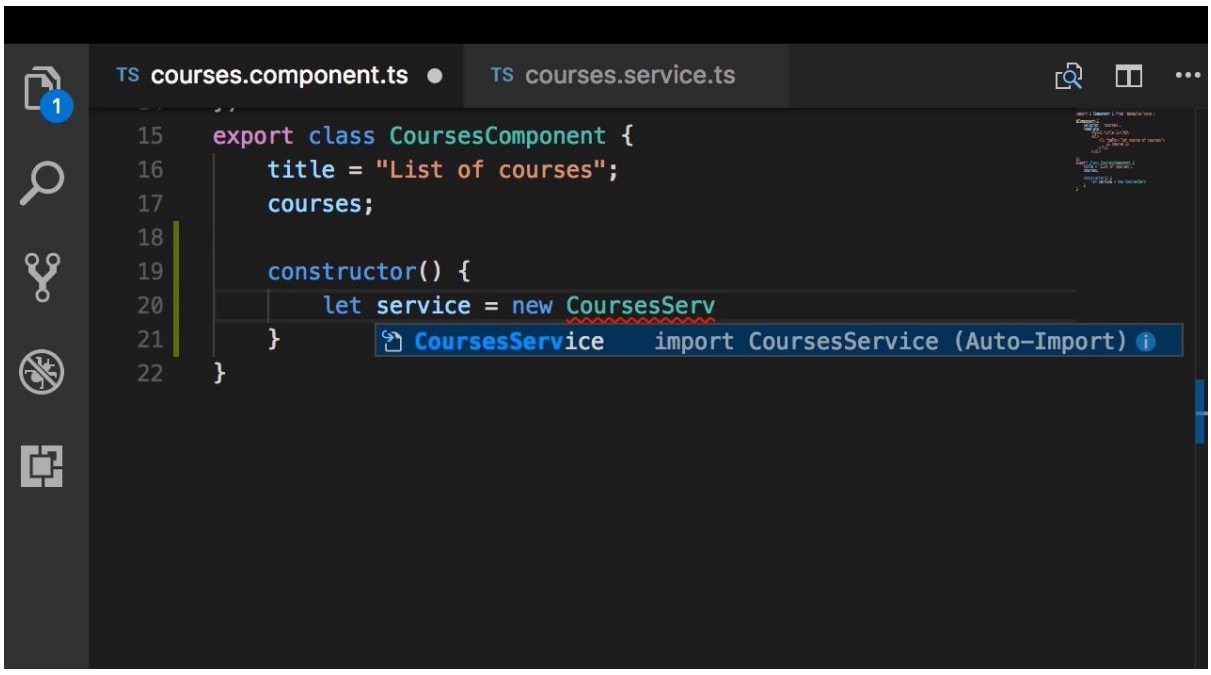

A service is a plain ts class with no decorator like in the case of a component class

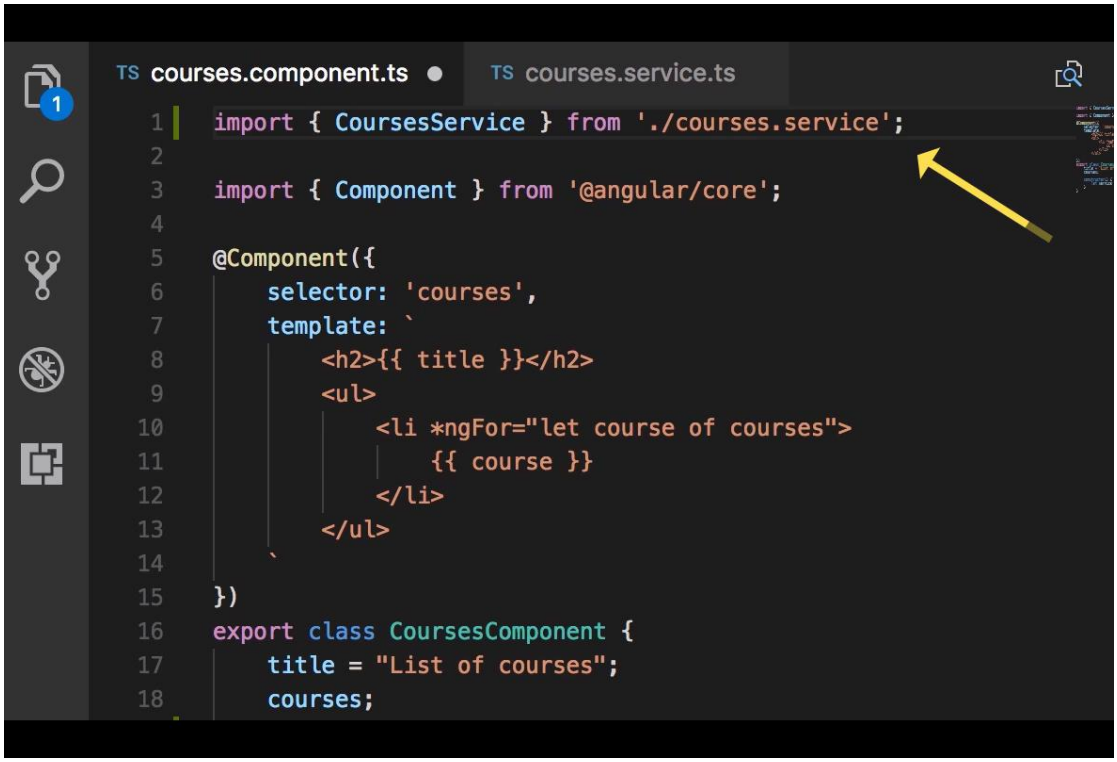


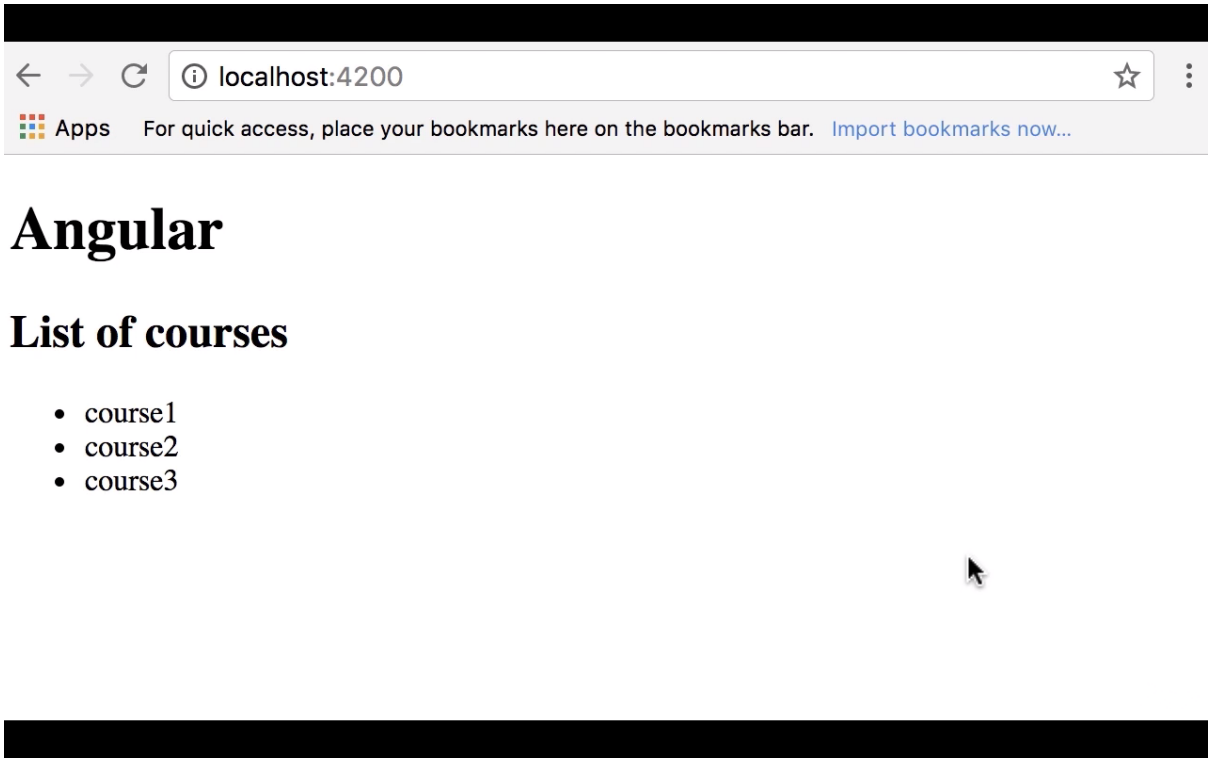

courses.component.ts
courses.service.ts
export class CoursesService {
getCourses() {
return ["course1", "course2", "course3"];
}
}

# Dependency Injection

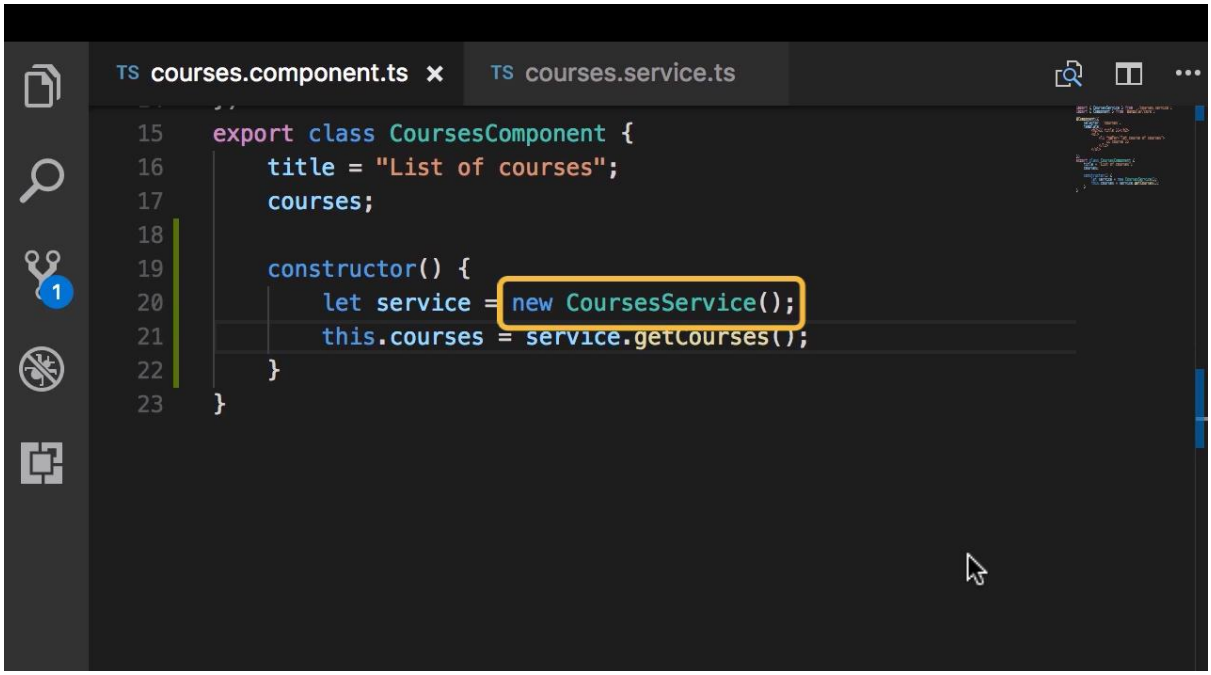
Dependency Injection



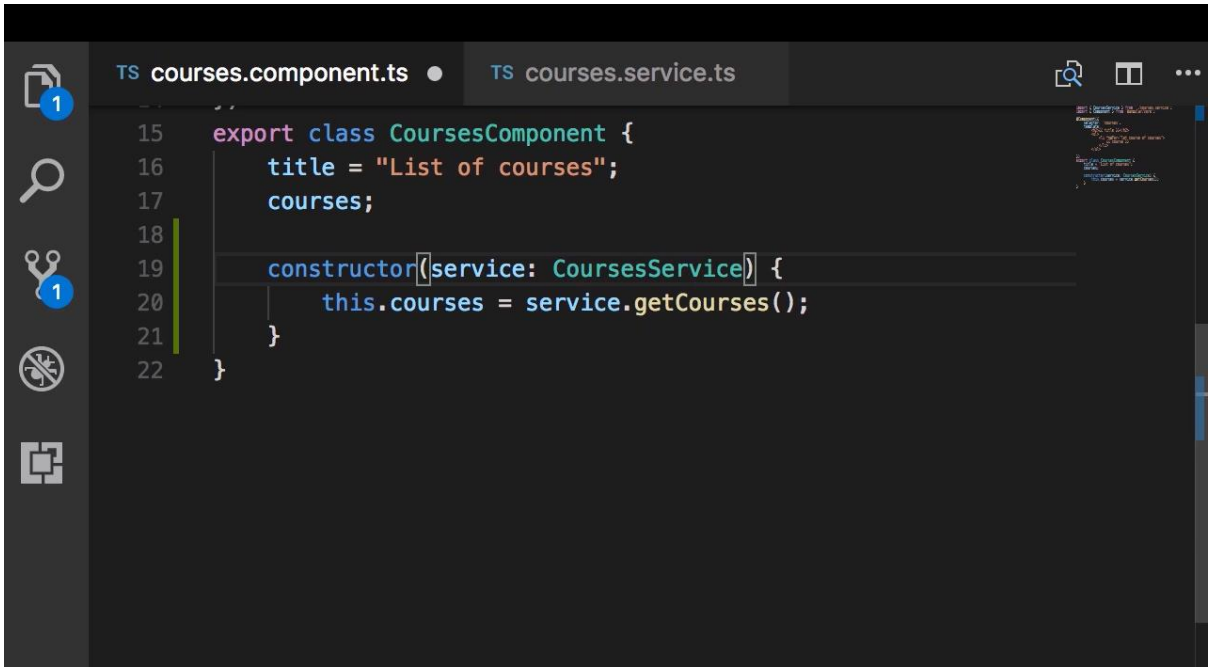


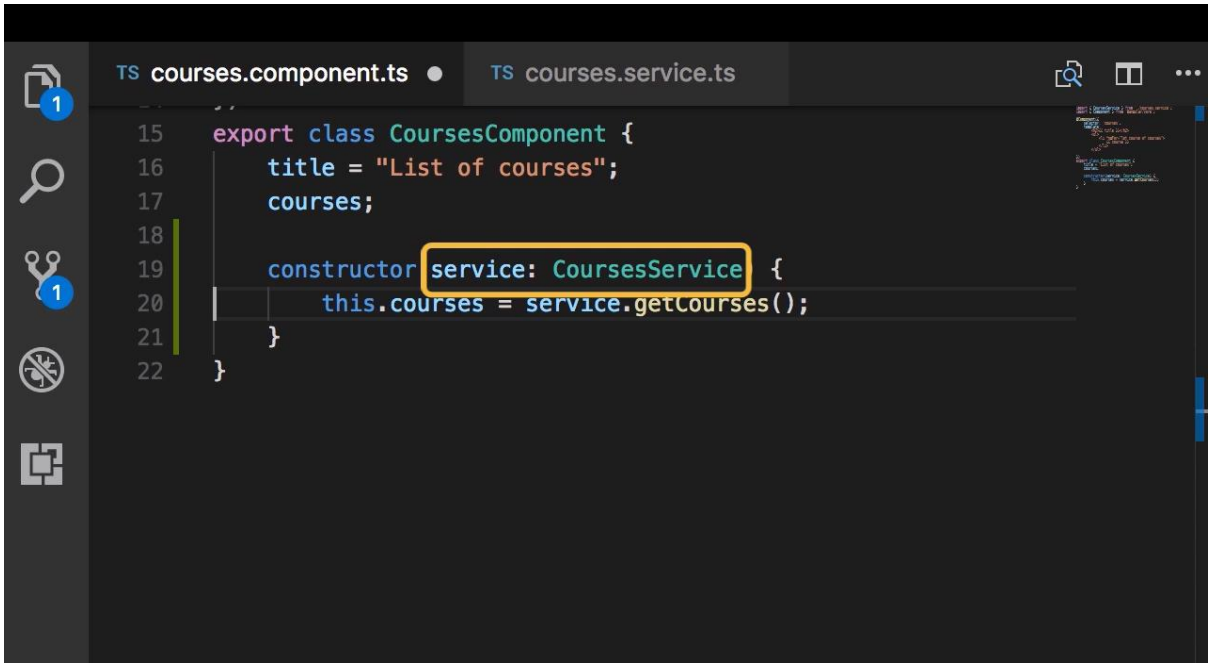

Tight coupling

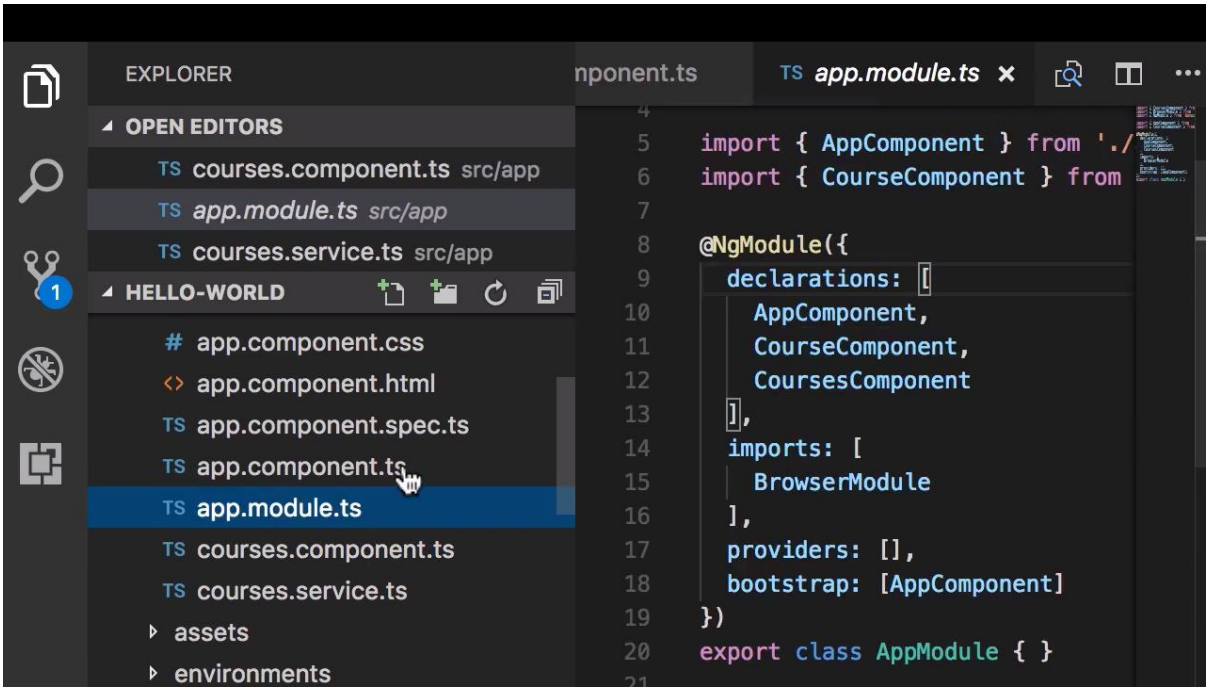


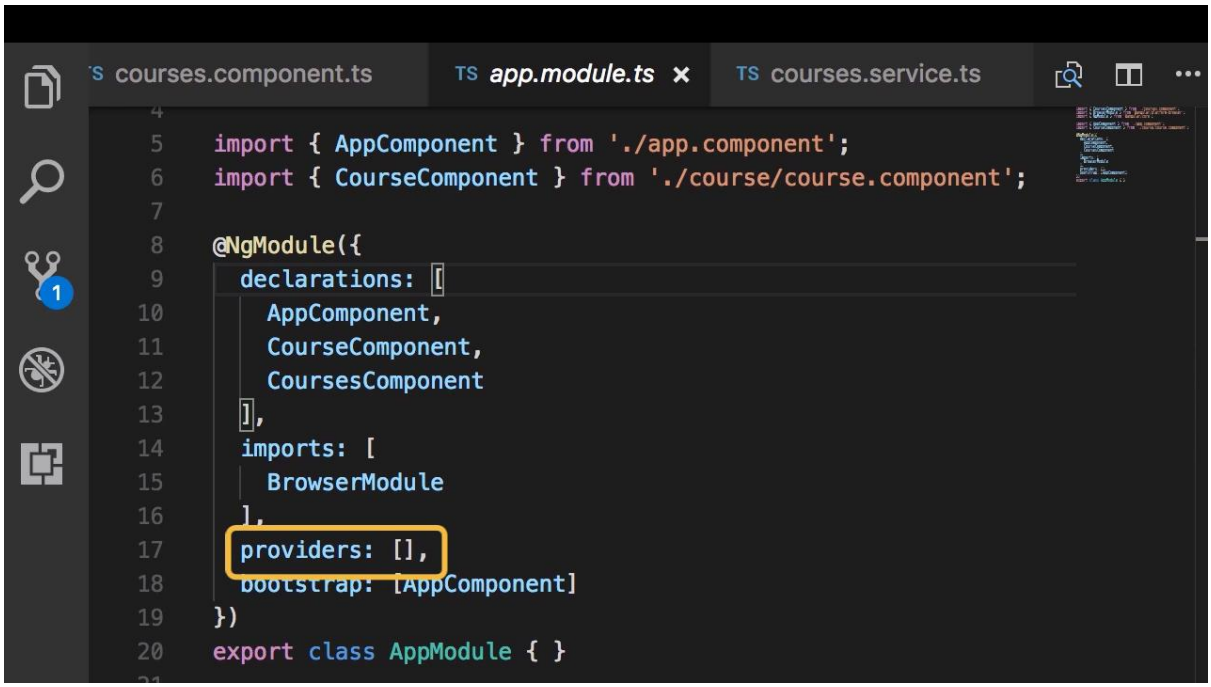

In this case angular injects the CourseService object by calling the constructor. There is no tight coupling

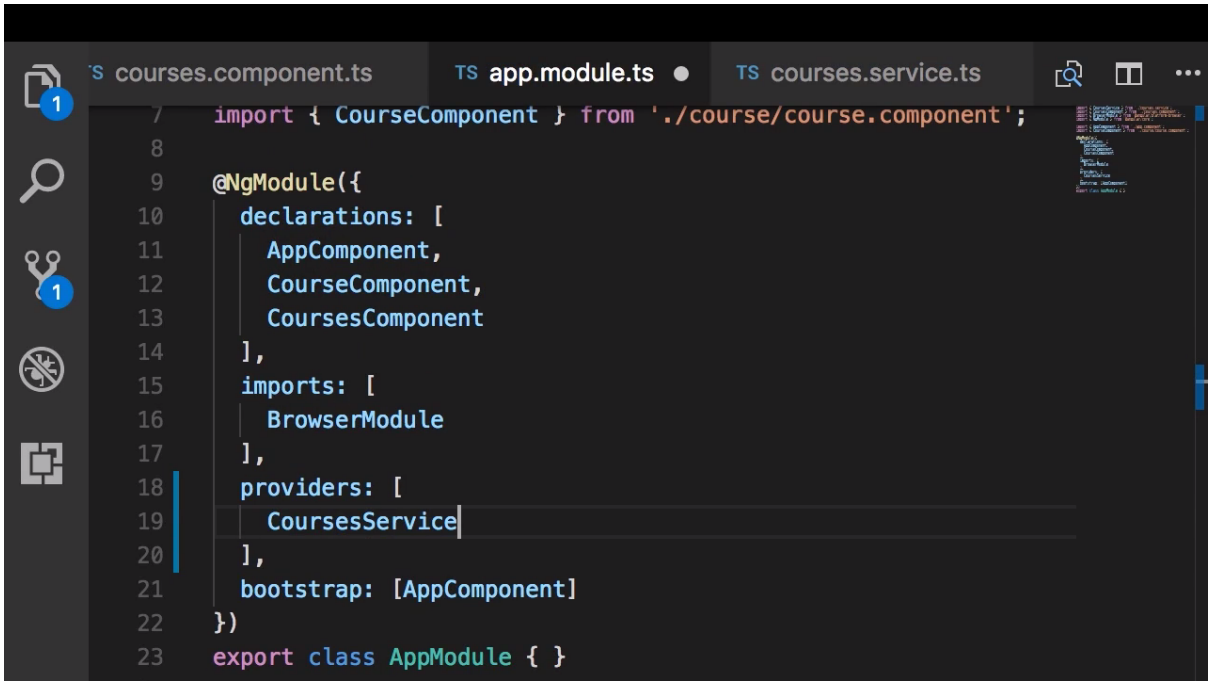


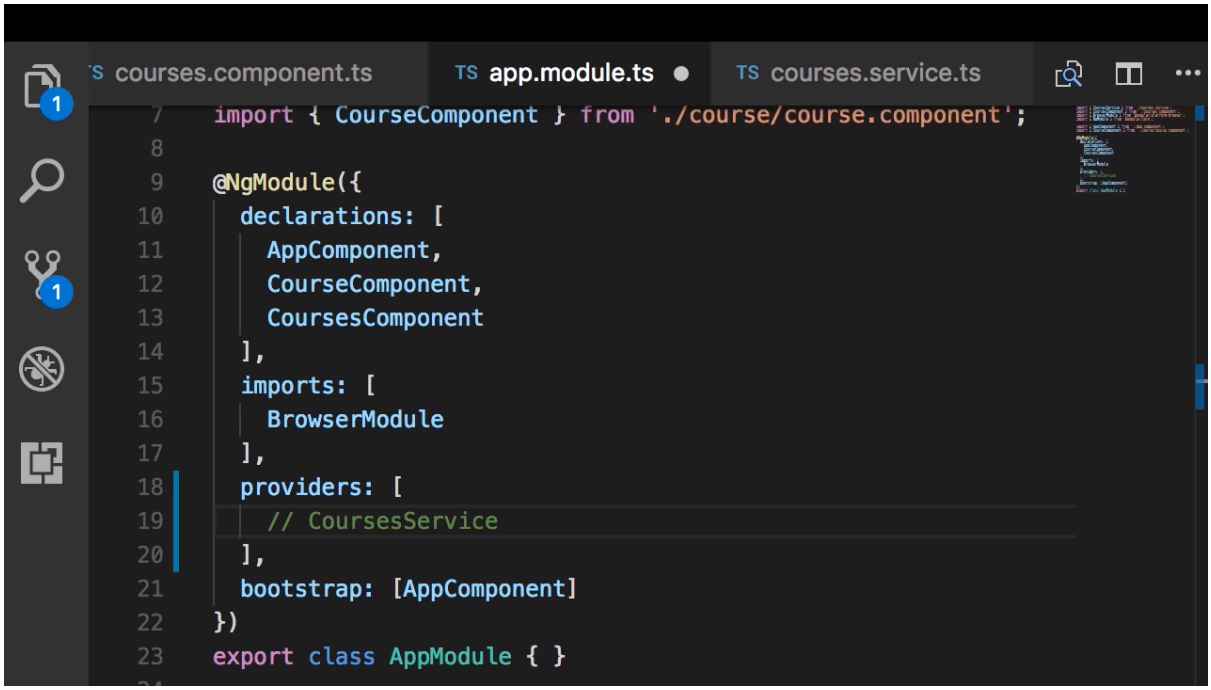
Now we need to register the dependencies for a give component . In this Case the CoursesComponent in order to perform its responsibilities , needs to have a copy of the courses. So angular injects the service that has the implementation of getting courses into the constructor CourseCopmponent constructor. So we need to now register the Dependency with the dependency injection framework of angular.

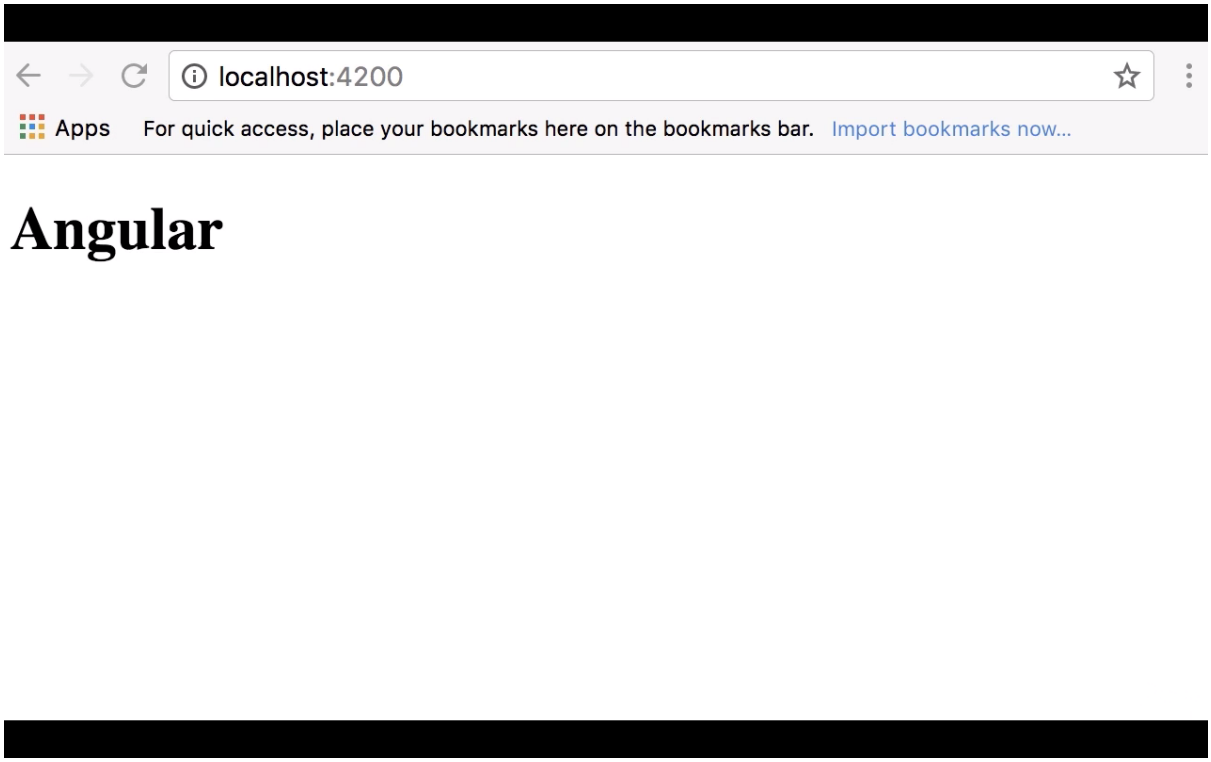


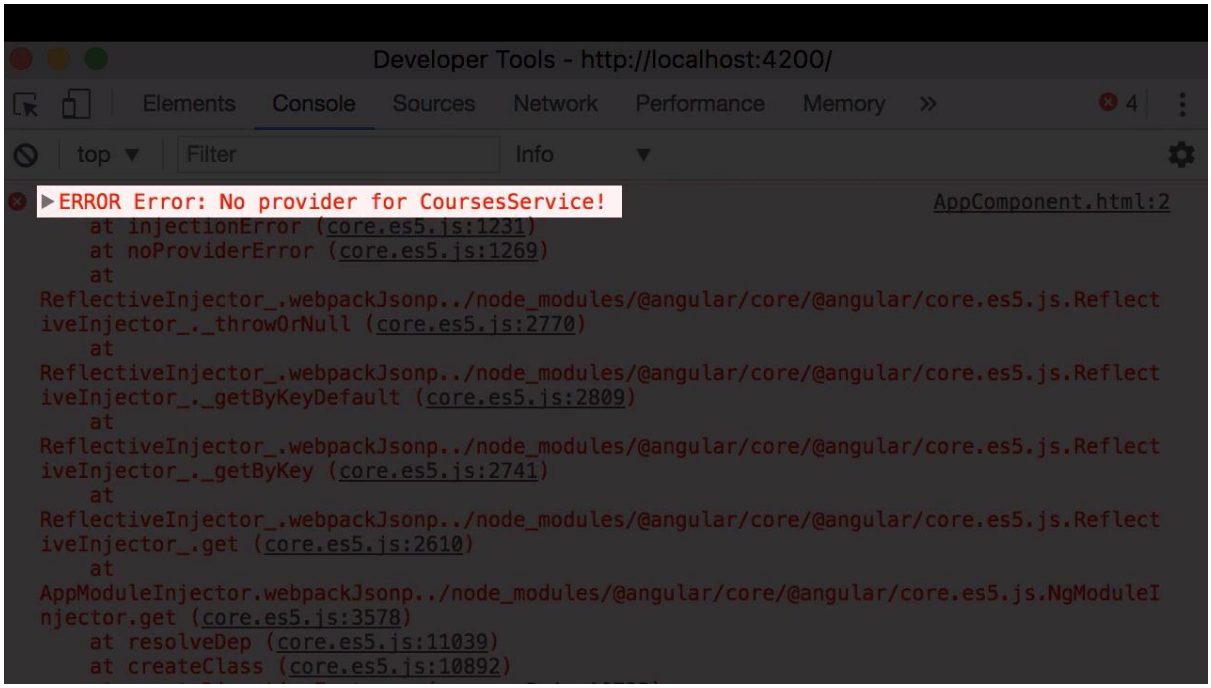



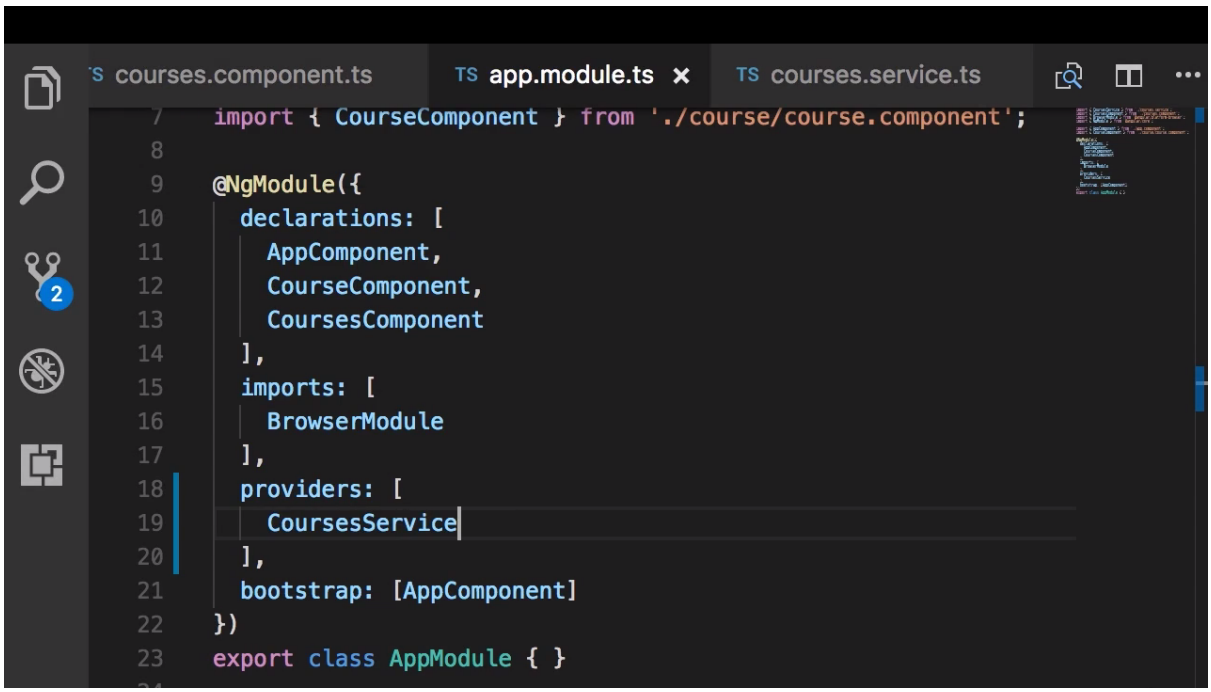

courses.component.ts
app.module.ts
courses.service.ts
import { AppComponent } from './app.component';
import { CourseComponent } from './course/course.component';
@NgModule({
declarations: [
AppComponent,
CourseComponent,
CoursesComponent
],
imports: [
BrowserModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

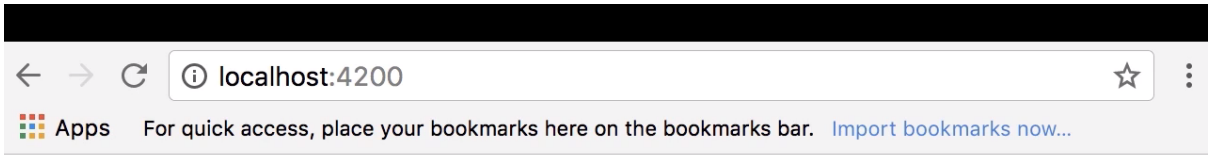

courses.component.ts
app.module.ts
courses.service.ts
import { CourseComponent } from './course/course.component';
@NgModule({
declarations: [
AppComponent,
CourseComponent,
CoursesComponent
],
imports: [
BrowserModule
],
providers: [
CoursesService
],
bootstrap: [AppComponent]
})
export class AppModule { }


courses.component.ts
app.module.ts
courses.service.ts
import { CourseComponent } from './course/course.component';
@NgModule({
declarations: [
AppComponent,
CourseComponent,
CoursesComponent
],
imports: [
BrowserModule
],
providers: [
// CoursesService
],
bootstrap: [AppComponent]
})
export class AppModule { }


localhost:4200
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Angular


Developer Tools - http://localhost:4200/
Elements
Console
Sources
Network
Performance
Memory
top
Filter
Info
ERROR Error: No provider for CoursesService!
AppComponent.html:2
at injectionError (core.es5.js:1231)
at noProviderError (core.es5.js:1269)
at ReflectiveInjector_.webpackJsonp../node_modules/@angular/core/@angular/core.es5.js.ReflectiveInjector_._throwOrNull (core.es5.js:2770)
at ReflectiveInjector_.webpackJsonp../node_modules/@angular/core/@angular/core.es5.js.ReflectiveInjector_._getByKeyDefault (core.es5.js:2809)
at ReflectiveInjector_.webpackJsonp../node_modules/@angular/core/@angular/core.es5.js.ReflectiveInjector_._getByKey (core.es5.js:2741)
at ReflectiveInjector_.webpackJsonp../node_modules/@angular/core/@angular/core.es5.js.ReflectiveInjector_.get (core.es5.js:2610)
at AppModuleInjector.webpackJsonp../node_modules/@angular/core/@angular/core.es5.js.NgModuleInjector.get (core.es5.js:3578)
at resolveDep (core.es5.js:11039)
at createClass (core.es5.js:10892)

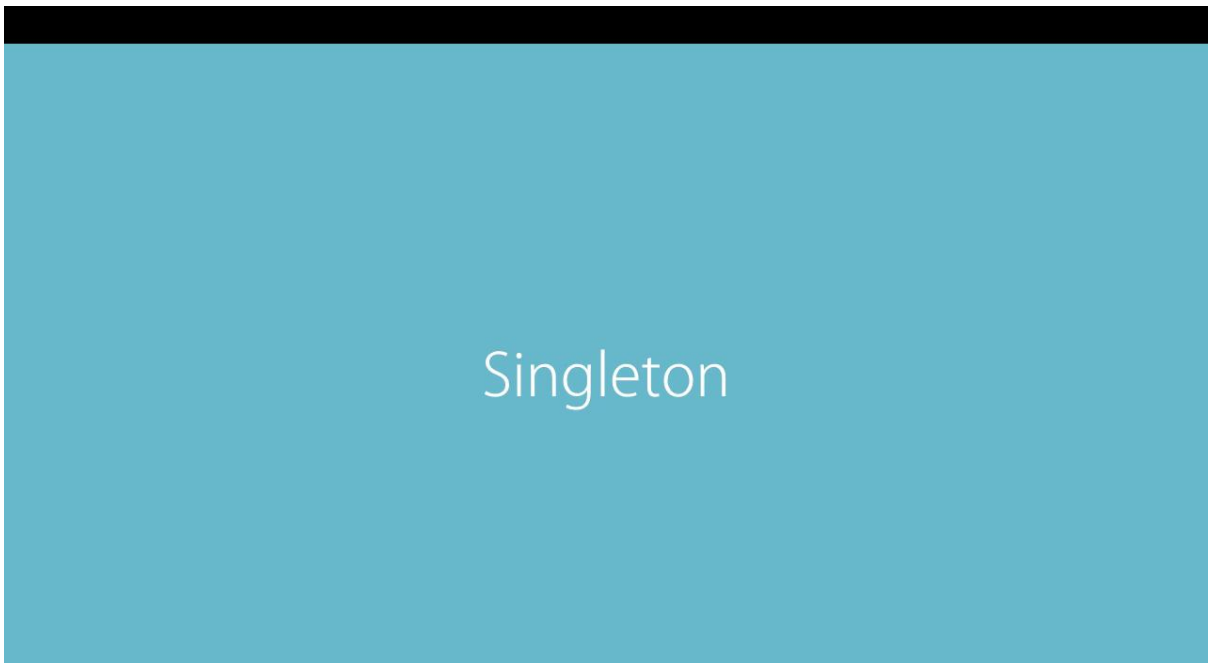

courses.component.ts
app.module.ts
courses.service.ts
import { CourseComponent } from './course/course.component';
@NgModule({
declarations: [
AppComponent,
CourseComponent,
CoursesComponent
],
imports: [
BrowserModule
],
providers: [
CoursesService
],
bootstrap: [AppComponent]
})
export class AppModule { }



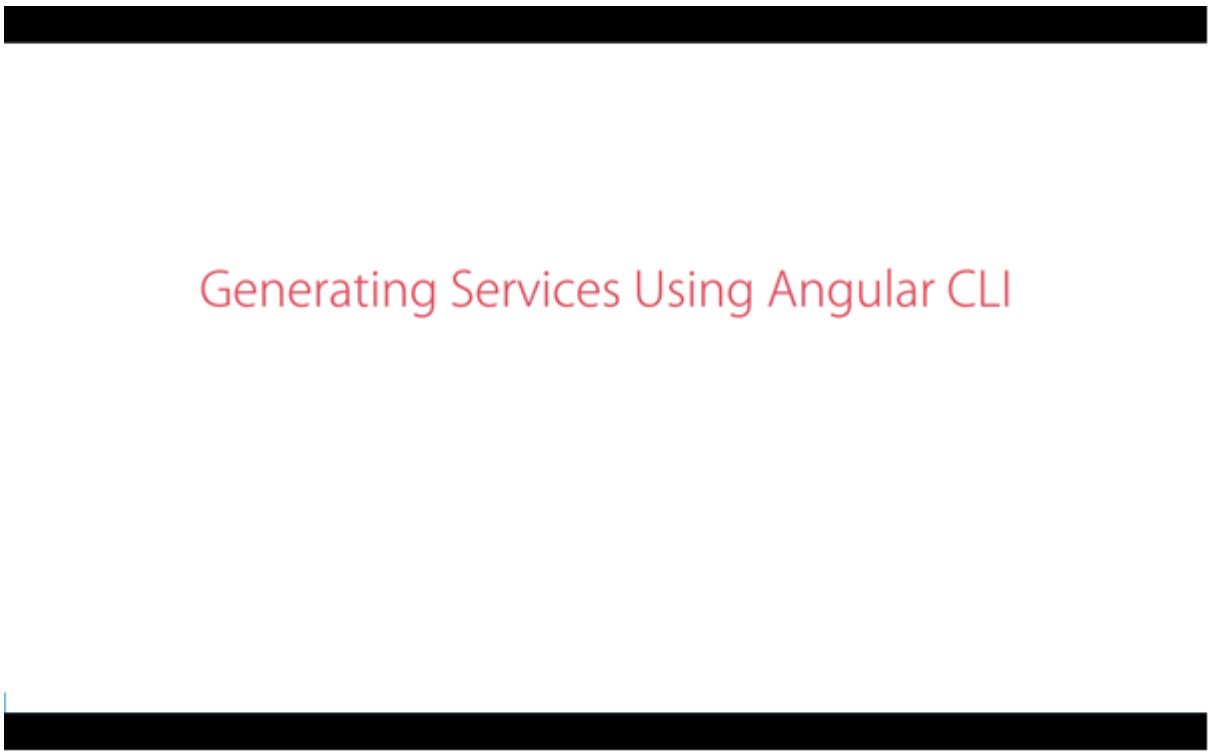

# Angular

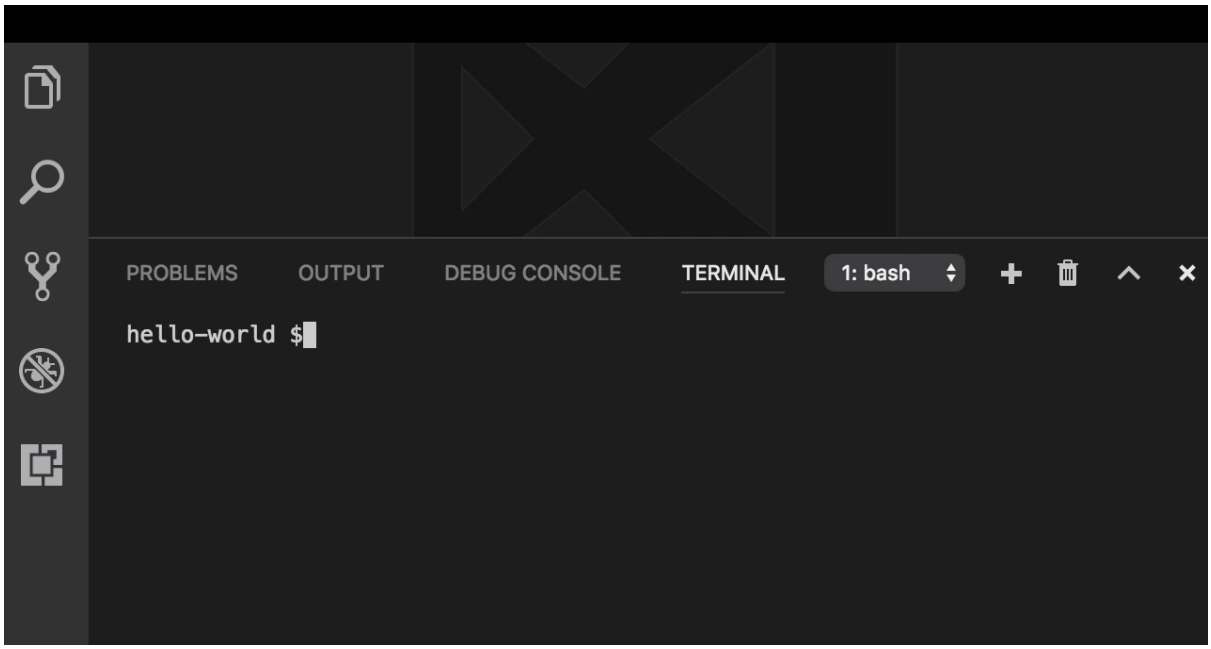
## List of courses

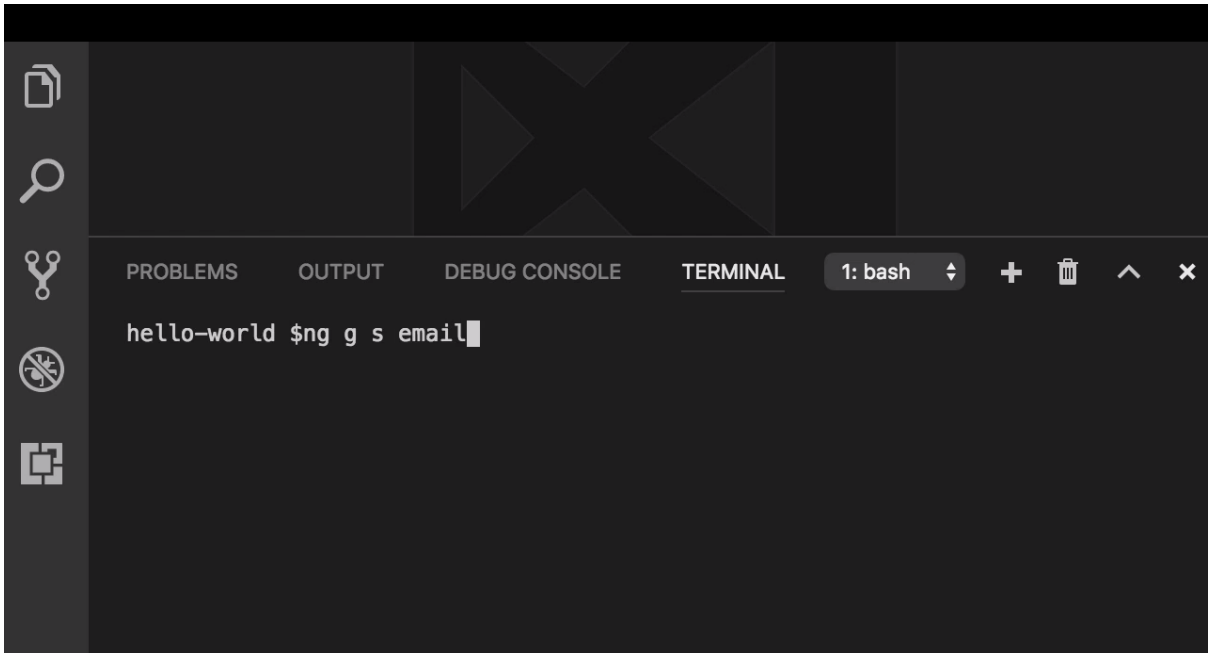
- course1
- course2
- course3

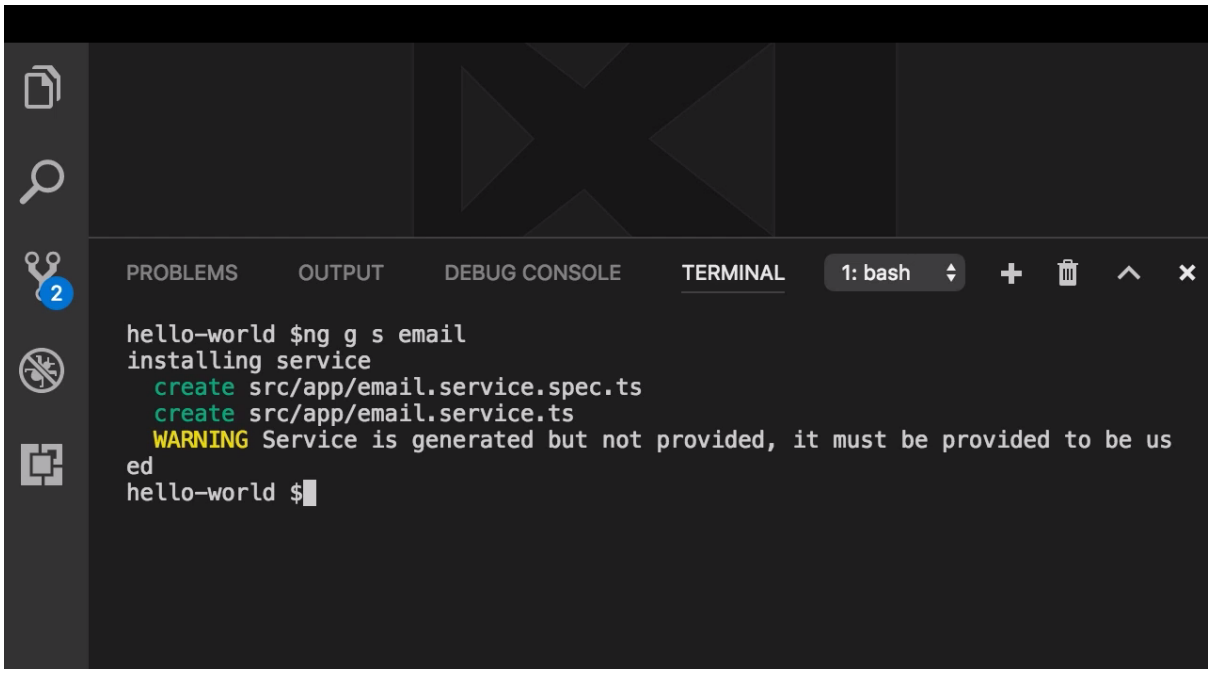


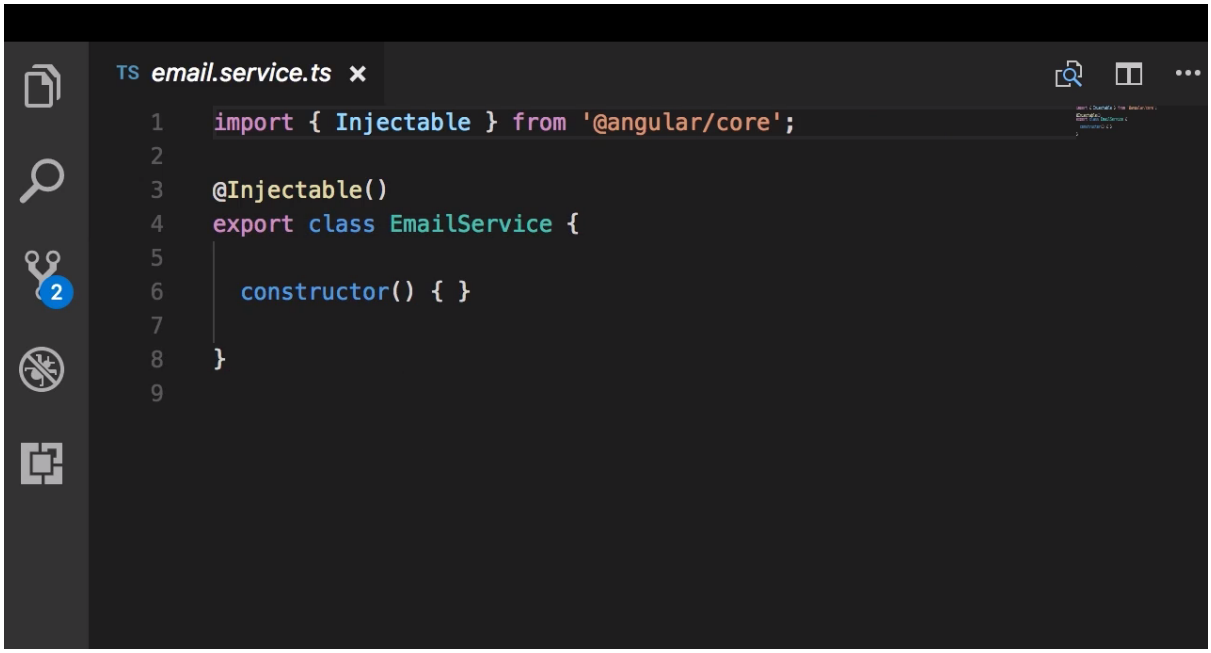
# Generating Services Using Angular CLI



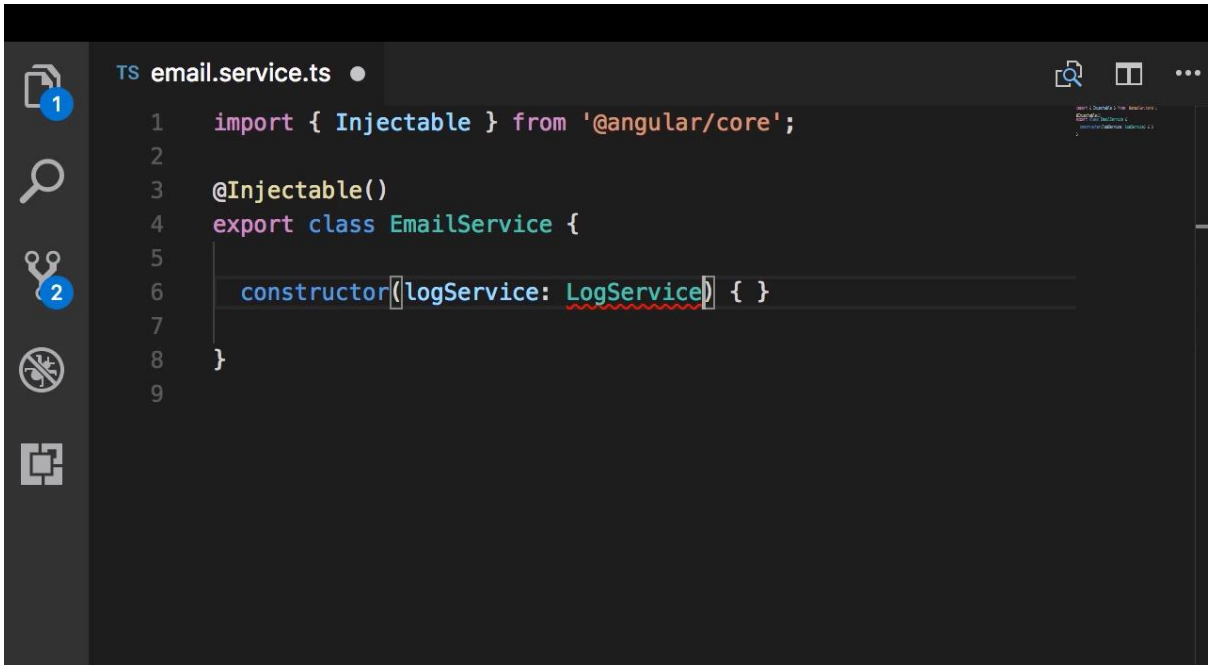



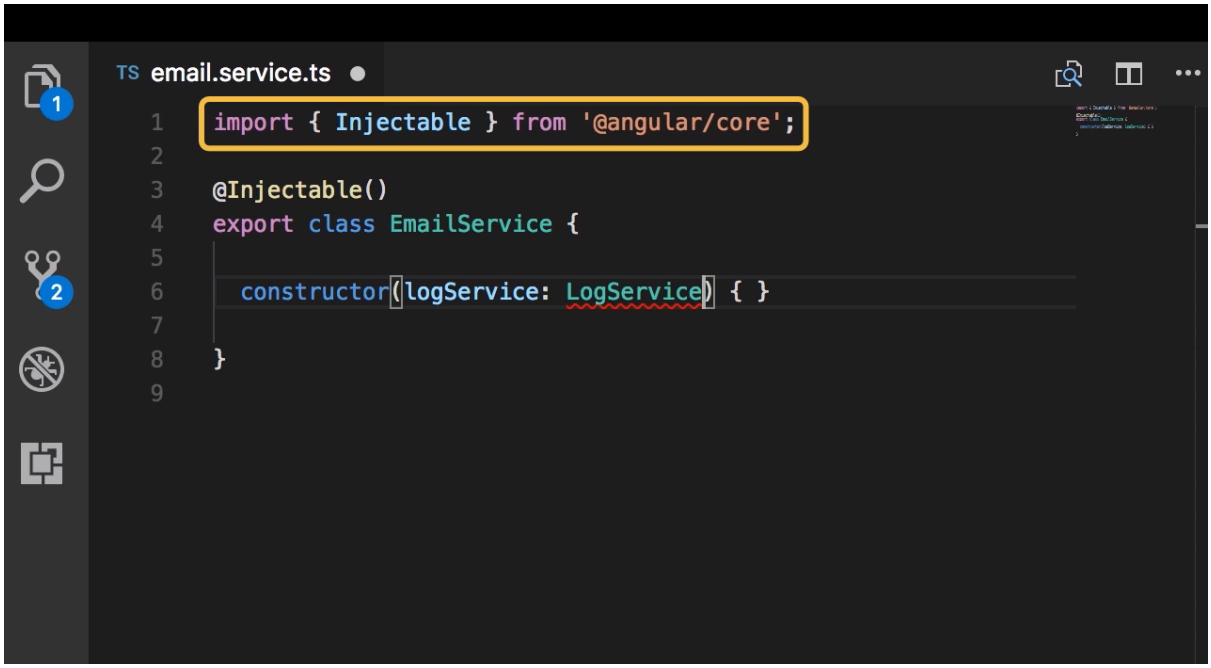

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: bash
hello-world $ng g s email


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: bash
hello-world $ng g s email
installing service
create src/app/email.service.spec.ts
create src/app/email.service.ts
WARNING Service is generated but not provided, it must be provided to be us
ed
hello-world $


email.service.ts
import { Injectable } from '@angular/core';
@Injectable()
export class EmailService {
constructor() { }
}

Injectable


email.service.ts
import { Injectable } from '@angular/core';
@Injectable()
export class EmailService {
constructor(logService: LogService) { }
}


TS email.service.ts
import { Injectable } from '@angular/core';
@Injectable()
export class EmailService {
constructor(logService: LogService) { }
}