Angular Learn :

1. Angular Setup:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | |  | |  |
|  |  | |  | | |  |  | | --- | --- | | Angular Commands | description | | Dotnet run | Run dot net app | | Node js file |  | | Npm install | Npm install | | Node -v | Check node version | | Npm-v | Checj npm version | | ng version - for all check versions |  | | npm install -g @angular/cli |  | | ng serve --open |  | | Ng serve |  | | Ng build |  | | ng generate component component-name    ng g c component-name | Create new component | | ng g c PageNotFound --skip-tests | page not found | |
|  | * 1. Vscode download   2. You need node.js to install   3. Check node version , please close vscode whle installing and after that run as admin   4. Check npm version as well   5. Install angular cli from vs code ( npm install -g @angular/cli)   6. When create amd pushing new angular proj to git ignore this nodemuodules files   7. When cloning install npm for node modules  1. **Angular Commands:** | |  | |
| Angular Commands | description | |
| Dotnet run | Run dot net app | |
| Node js file |  | |
| Npm install | Npm install | |
| Node -v | Check node version | |
| Npm-v | Checj npm version | |
| ng version - for all check versions |  | |
| npm install -g @angular/cli |  | |
| ng serve --open |  | |
| Ng serve |  | |
| Ng build |  | |
| ng generate component component-name    ng g c component-name | Create new component | |
| ng g c PageNotFound --skip-tests | page not found | |
| ng new angular-tailwind | Install tailwind css | |
| Ng new angular17\_without\_standalone – standalone false | With out standalone below | |
| ng generate service services/Post | Create service | |
| ng generate environments | Create environment file to store keys like apikey | |
| ng generate interface interfaces/MovieDetail | Create interfaces | |
| Ctrl+shift+alt+p | Convert object to interface | |

1. Important Points:

**Routes:**

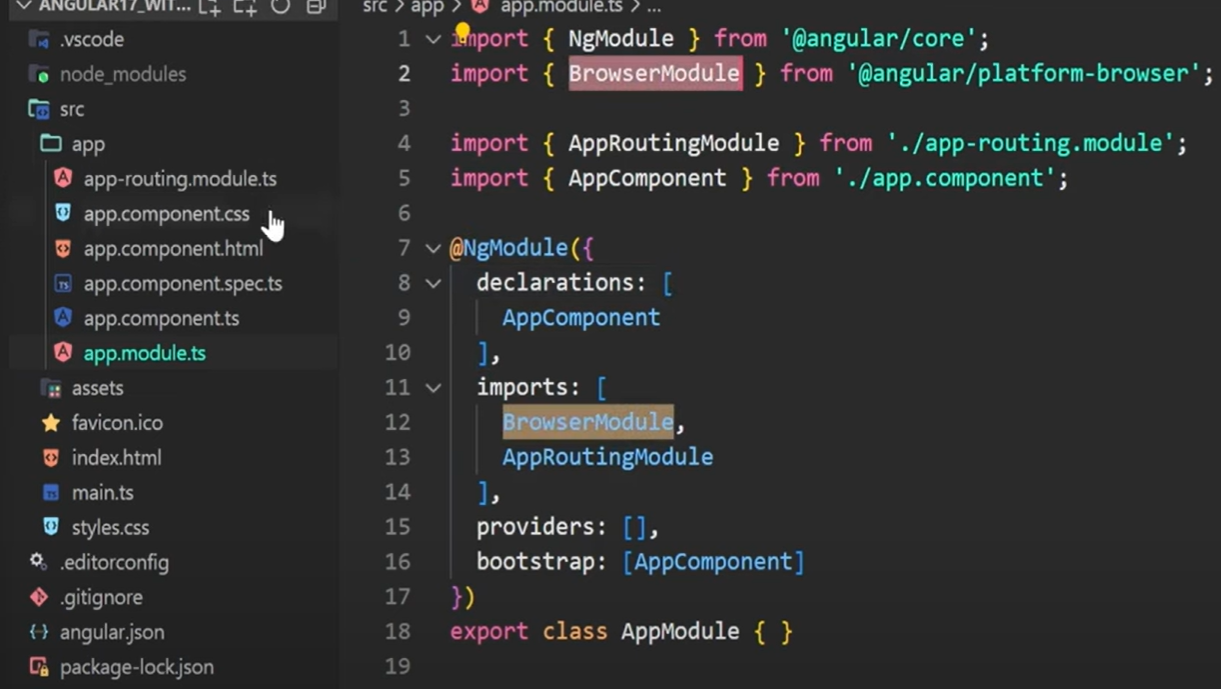
* + - App.component.html – is the first file for all navbar and routerlink
* <app-navbar></app-navbar>
* <router-outlet></router-outlet>
  + - Navbar .html
    - Next all pages – home , about, contact

1. **Few points:**

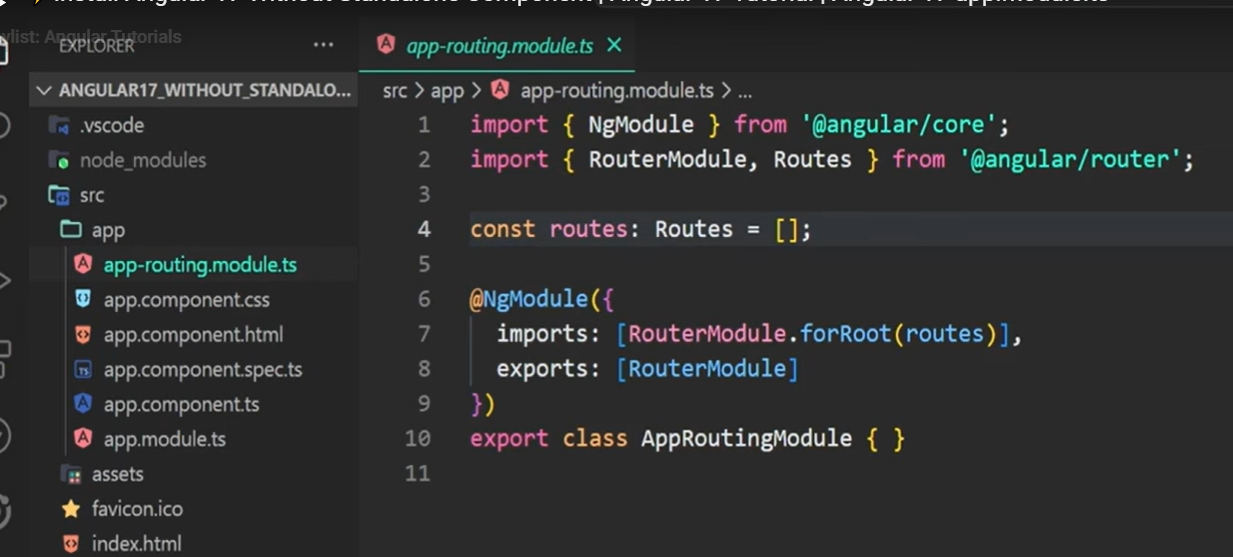
* Angualr 17 without standalone component means no app.module.ts. Below see standalone
* @Component({
* selector: 'app-root',
* standalone: true,
* imports: [NavbarComponent, RouterOutlet],
* templateUrl: './app.component.html',
* styleUrl: './app.component.css'
* })
* App.Route.ts has replaced App.Module.ts

**a. With out standalone :**

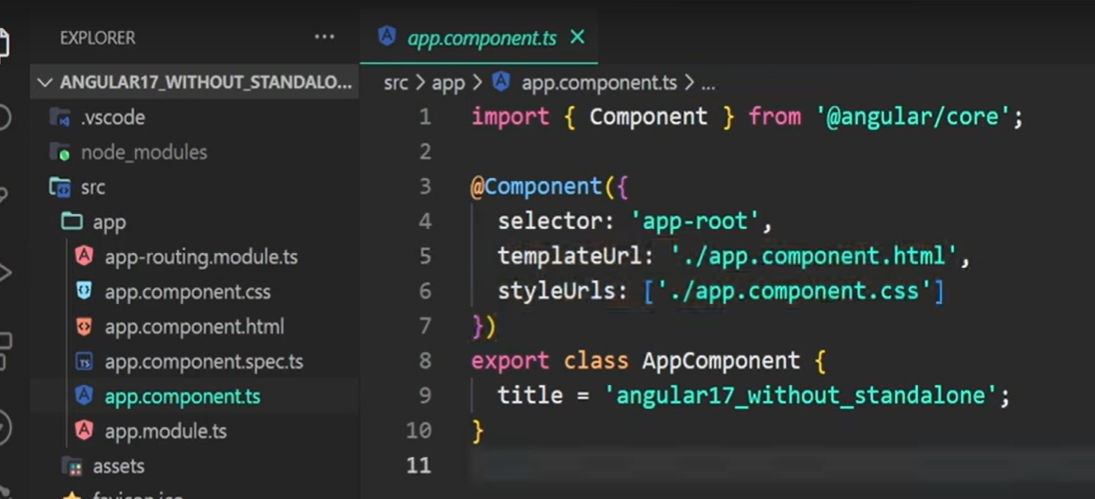
* Ng new angular17\_without\_standalone – standalone false : you will see app.module.ts



App.routimg.module.ts



**App.component.ts**



**b.Adding Child Routes :**

**Company Page : /about/company**

1. Child – company component: about/company
2. Create new component company
3. Add button in about html:
4. <div class="flex-justify-center">
5. <button routerLink="company" class="flex mx-auto text-white bg-yellow-500 border-0 py-2 px-8 focus:outline-none hover:bg-indigo-600 rounded text-lg">Company</button>
6. </div>
7. In company.ts make sure import router link
8. @Component({
9. selector: 'app-about',
10. standalone: true,
11. imports: [RouterLink],
12. templateUrl: './about.component.html',
13. styleUrl: './about.component.css'
14. In app.route .ts please add child route under about :

****

* Same for Employees Page as child page for about.

**c.Defer Block : video**

* Deferrable views , also known as @defer blocks, are powerful toll that can be used to reduce the initial bundle size of your application or defer heavy components that may not ever loaded until later time.
* Inorder for dependencies with @defer block that need meet two conditions:
  + - * + They must be standalone. Non standalone dependieces cannot de deferred and will still be earky loaded even inside of @defer blocks
        + They must not be directly referenced from the same file outside of @defer blocks this include ViewChild queries.
        + **Eager Loading**

**Eager loading** is the default approach in Angular. In eager loading, all the necessary modules and their components are loaded when the application starts.

* **Lazy Loading**

**Lazy loading** is a technique where Angular loads modules asynchronously when a specific route is activated.

Resources:

1. Routing : phpnodetuts - main

<https://www.youtube.com/watch?v=lIb_gnleUns>

1. Questpond:

<https://www.youtube.com/watch?v=-9VcW7MBDs8>