Harmony IT Solution

Angular

Tahaluf Training Center 2022

1 Routing.

2 Default Route.

3 Create and Validate Form.

Objective

The Objective of this lecture

- Understand the routing in the angular and how to navigate from one component to another.
- Get to know about the reactive form.
- How to deal with the form control and apply the validation for each input.

Routing

Overview of Routing

Routing allows you to move from one part of the application to another part or one View to another View.

Overview of Routing

Since the app component is the first component that is loaded when the project
Has run and if you want to go to another page we will use routing in the app component.

Add this tag in the app.component.html:

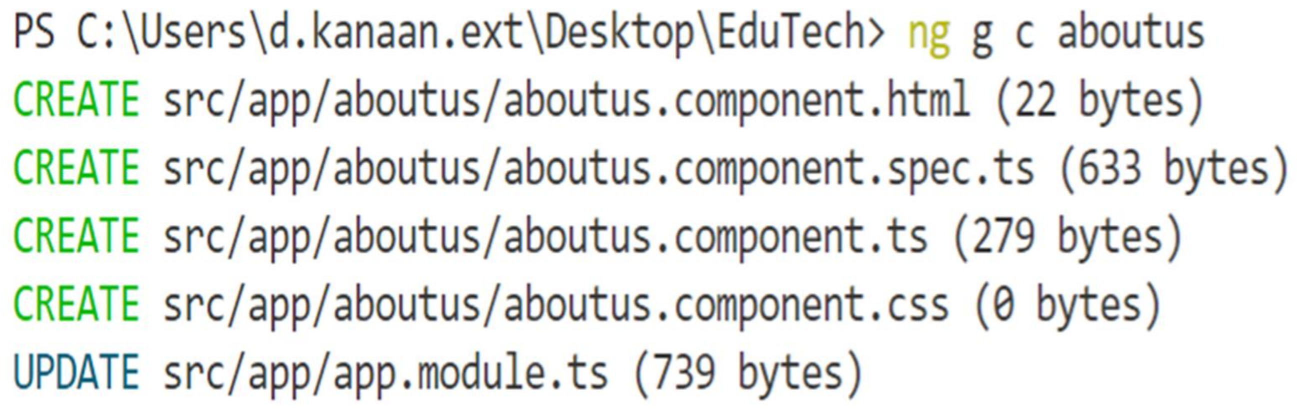
```
<router-outlet></router-outlet>
```

Example of Routing

Generate a new component called about us and another component called contact us in the app module.

Example Solution

Generate about us component in the app module:



```
PS C:\Users\d.kanaan.ext\Desktop\EduTech> ng g c aboutus
CREATE src/app/aboutus/aboutus.component.html (22 bytes)
CREATE src/app/aboutus/aboutus.component.spec.ts (633 bytes)
CREATE src/app/aboutus/aboutus.component.ts (279 bytes)
CREATE src/app/aboutus/aboutus.component.css (0 bytes)
UPDATE src/app/app.module.ts (739 bytes)
```

Overview of Routing

The first page will load is app components and if you want to go to the about us component in the same module **(App Module).**

In-app**-routing.module.ts,** There is an array called **routes**
this array is used to add the route for all components.

Overview of Routing

Each route is an object and each route consists of a path
and component

path: where you want to go for example /aboutus but you must write the page name without using /.

component: component name which you want to display.

Example of Routing

```
const routes: Routes = [
  {
      path: 'aboutus',
      component: AboutusComponent
  }
];
```

NOTE: Once you write the component name it will import this component like this:

```
import { AboutusComponent } from './aboutus/aboutus.component';
```

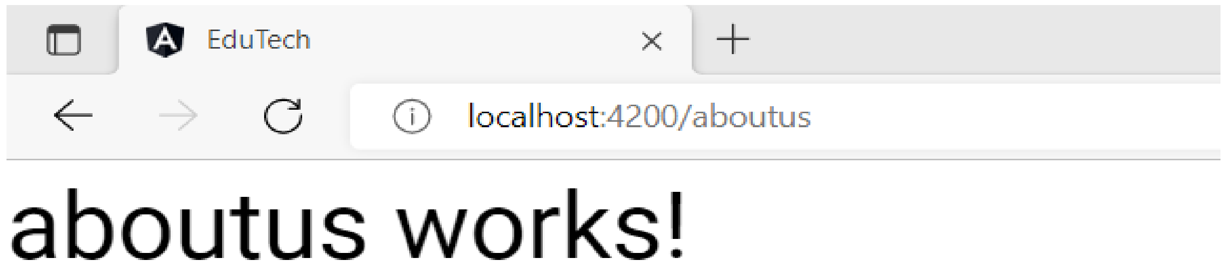
Example of Routing

To add more than one route separate them with a comma.

```
const routes: Routes = [
  {
      path: 'aboutus',
      component: AboutusComponent
  },
  {
      path: 'contactus',
      component: ContactusComponent
  }
];
```

Example of Routing

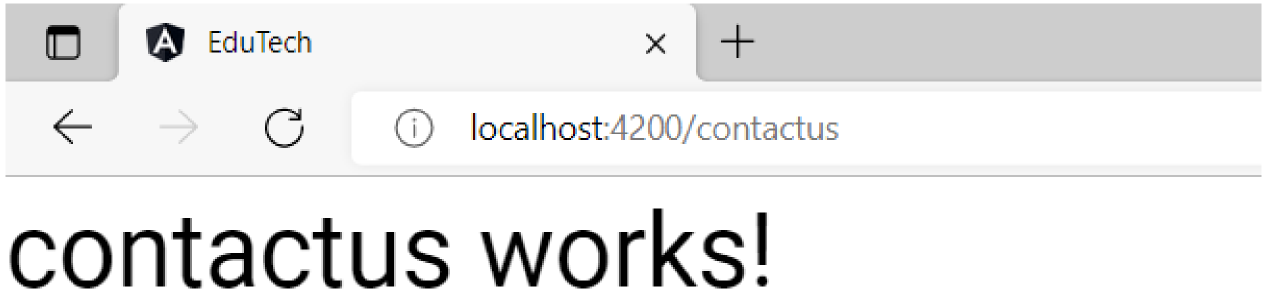
Because about us component in the app module use /aboutus in the URL.



aboutus works!

Example of Routing

Contact us component.



```
localhost:4200/contactus
```

contactus works!

Default Route

Overview of Routing

The default route is redirected -for example- to the home path.

This means that, when you navigate to the root of your application /, you are redirected to the home path (/home).

The path is empty, indicating the default route.

Example of Routing

The default route of the project is the about us component.

```
const routes: Routes = [
  {
      path: '',
      component: AboutusComponent
  },
]
```

In this way when the project was run the about us component will load.

Exercise

Create a route for auth module to login and register components.

Example of Routing

If you want to make the login page the first page to be loaded.

In-app-routing.module.ts

```
const routes: Routes = [
  {
      path: '',
      loadChildren: () => import('./auth/auth.module').
          then((m) => m.AuthModule)
  },]
```

Example of Routing

And In auth-routing.module.ts :

```
const routes: Routes = [
  {
      path: '',
      component: LoginComponent
  },
  {
      path: 'register',
      component: RegisterComponent
  }
];
```

Shared Module

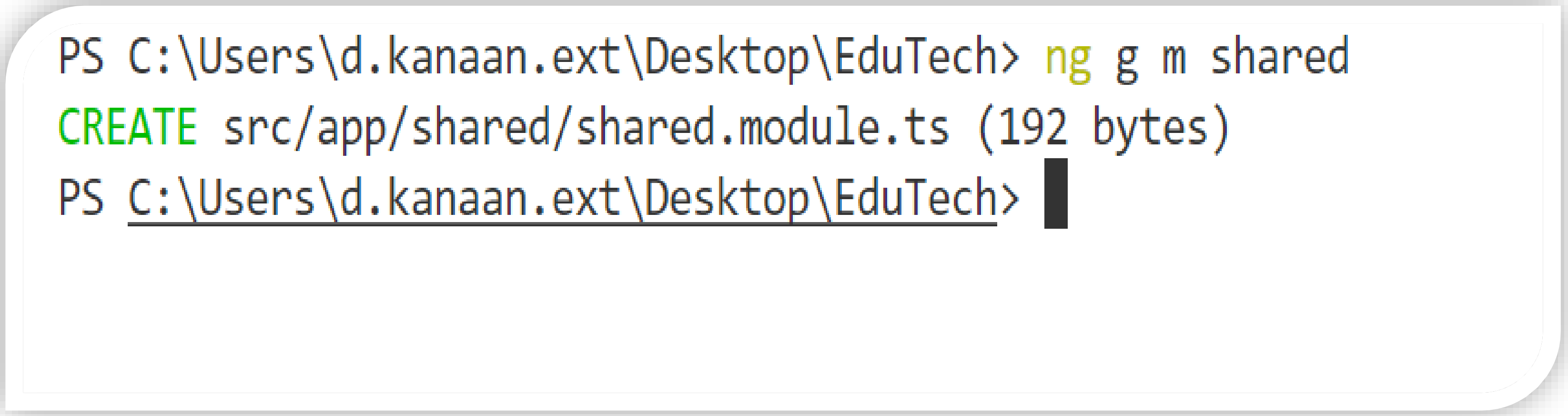
Overview of Shared Module

Your code can be organized and streamlined by creating shared modules.

Directives, pipes, and components that are commonly used can be put into this module, which you can then import wherever you need them in your application.

Generate A Shared Module

In order to generate a shared module, the same command to generate any other module will be used.



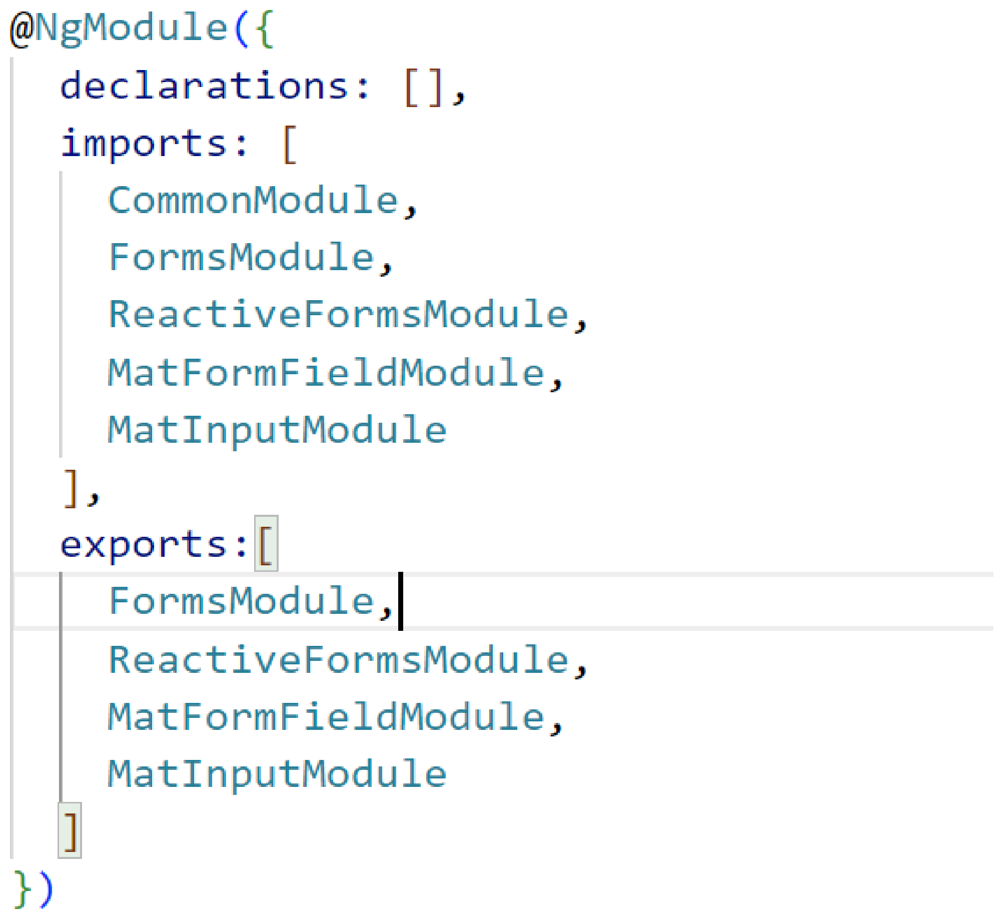
```
PS C:\Users\d.kanaan.ext\Desktop\EduTech> ng g m shared
CREATE src/app/shared/shared.module.ts (192 bytes)
PS C:\Users\d.kanaan.ext\Desktop\EduTech>
```

Consider the following:

The CommonModule is imported because the module's component needs common directives.

The module declares and exports utility pipes, directives, and components.

It re-exports all modules, components,
pipes, and directives that are declared and imported.



```
@NgModule({
  declarations: [],
  imports: [
    CommonModule,
    FormsModule,
    ReactiveFormsModule,
    MatFormFieldModule,
    MatInputModule
  ],
  exports:[
    FormsModule,
    ReactiveFormsModule,
    MatFormFieldModule,
    MatInputModule
  ]
})
```

Notes:

You can create a template that you will use multiple times, such as navbars, footers, or sidebars, in the shared module.

Exercises

Create a navbar and footer components in the shared module and transfer the template from the app module to the shared module.

Include the shared module

In order to include the navbar and footer component in the login component, you need first import the shared module in the **auth module**, then use the selector of the navbar and footer component in the login component.

References

[1] Angular, "Angular," *Angular.io*, 2019. https://angular.io/

[2] "Complete Angular Tutorial For Beginners," *TekTutorialsHub*. https://www.tektutorialshub.com/angular-tutorial/

[3]"npm | build amazing things," *Npmjs.com*, 2019. https://www.npmjs.com/

[4]"Angular Tutorial for Beginners | Simplilearn," *Simplilearn.com*. https://www.simplilearn.com/tutorials/angular-tutorial (accessed Aug. 19, 2022).

Harmony IT Solution