

# Angular

Tahaluf Training Center 2021



## Chapter 3

**1** Moving from one page to another

**2** Create and Validate form



## Moving from one page to another

Since App components are the first components which be loaded when the project has run and you want to go to another page you must use routing here.



## Moving from one page to another

**First** You must use this tag in the app.component.html

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

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



## Moving from one page to another

Generate about us component in app module :

```
PS C:\Users\User\Desktop\portal> 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 (479 bytes)  
PS C:\Users\User\Desktop\portal> |
```



## Moving from one page to another

Generate contact us component in app module:

```
PS C:\Users\User\Desktop\portal> ng g c contactus  
CREATE src/app/contactus/contactus.component.html (24 bytes)  
CREATE src/app/contactus/contactus.component.spec.ts (647 bytes)  
CREATE src/app/contactus/contactus.component.ts (287 bytes)  
CREATE src/app/contactus/contactus.component.css (0 bytes)  
UPDATE src/app/app.module.ts (573 bytes)  
PS C:\Users\User\Desktop\portal>
```



## Moving from one page to another

The first page will load is app components and if you want to go to about us page 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.



## Moving from one page to another

Each rout is an object and each rout consists of a path and component Where :

**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.





## Moving from one page to another

```
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';
```



## Moving from one page to another

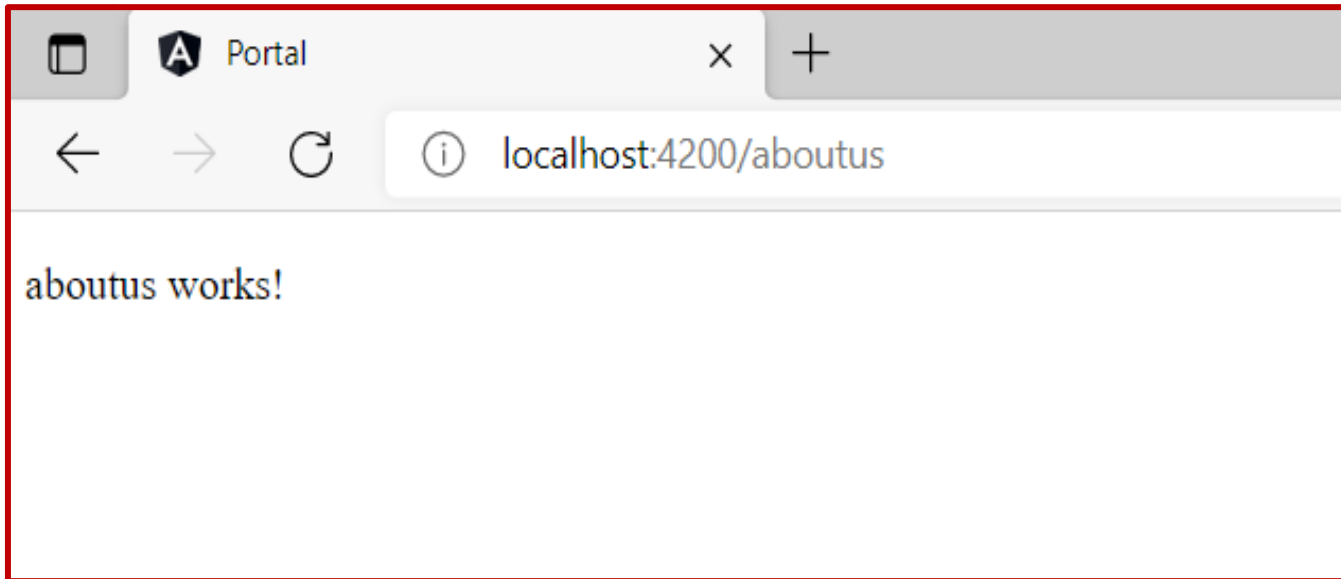
To add more than one route Separate them with a comma.

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



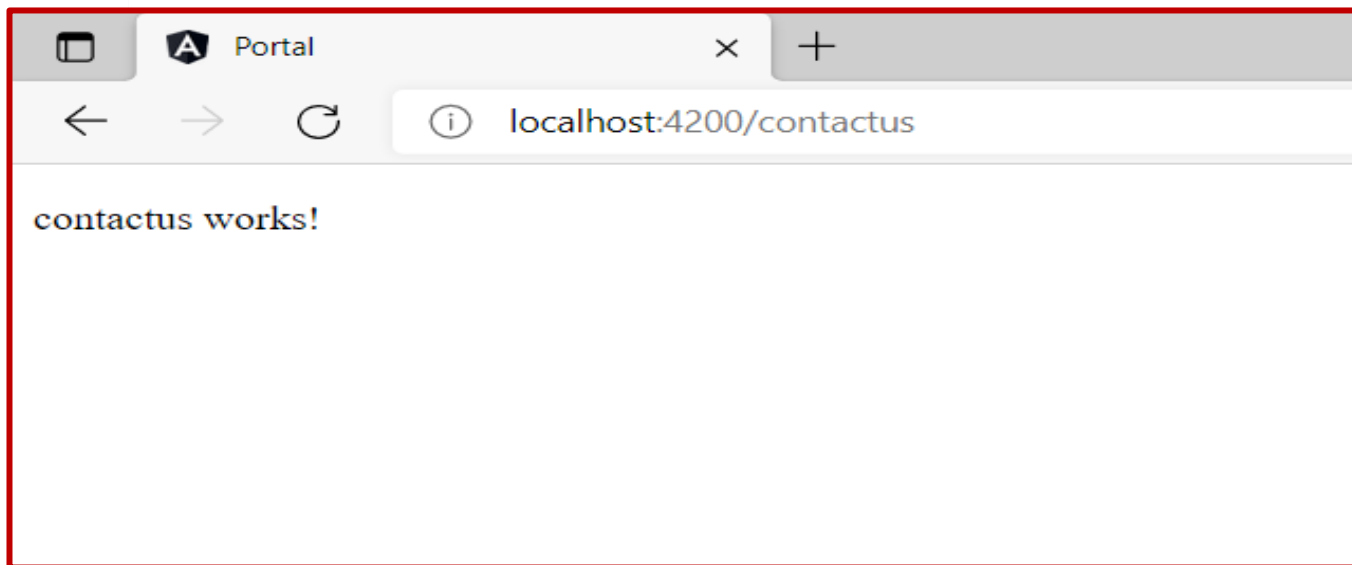
## Moving from one page to another

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



# Moving from one page to another

/ contactus



## Moving from one page to another

Default route: path of the default route is empty string.

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

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



## Moving from one page to another

### Exercise:

Create Route for auth module to login and register component.



## Moving from one page to another

### Exercise Solution:

In the auth-routing.module.ts will add routing for login and register .

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



## Moving from one page to another

Now If you want to use /login, it won't work because the auth module doesn't load.

And to route the auth module , go to app-routing.module.ts and add this code :

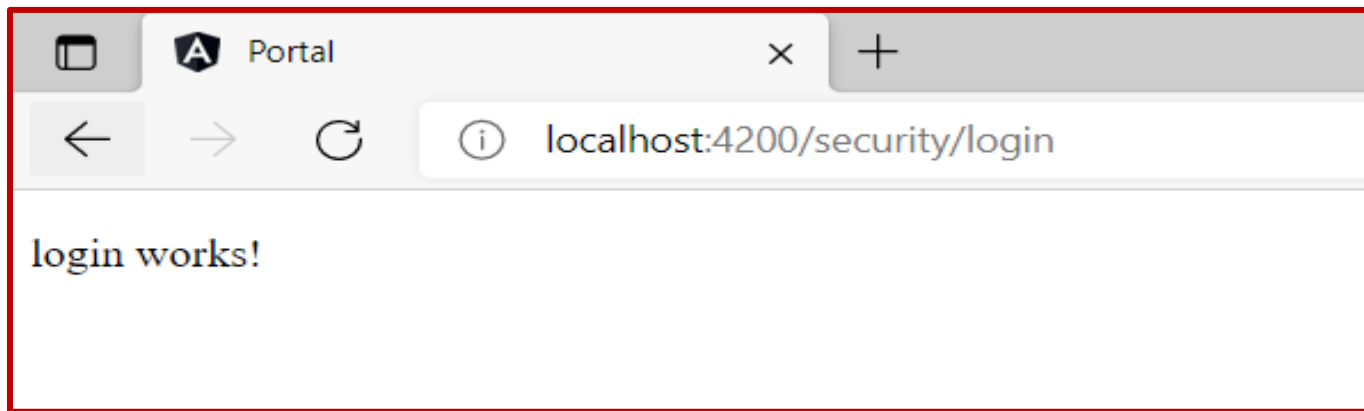
```
{  
  path: 'security',  
    loadChildren : () =>  
import('./auth/auth.module').  
then((m) => m.AuthModule)  
}  
];
```





## Moving from one page to another

To go to login page or register page you must use **/security/login** or **/security/register** in URL.



## Moving from one page to another

But 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)  
  },  
];
```



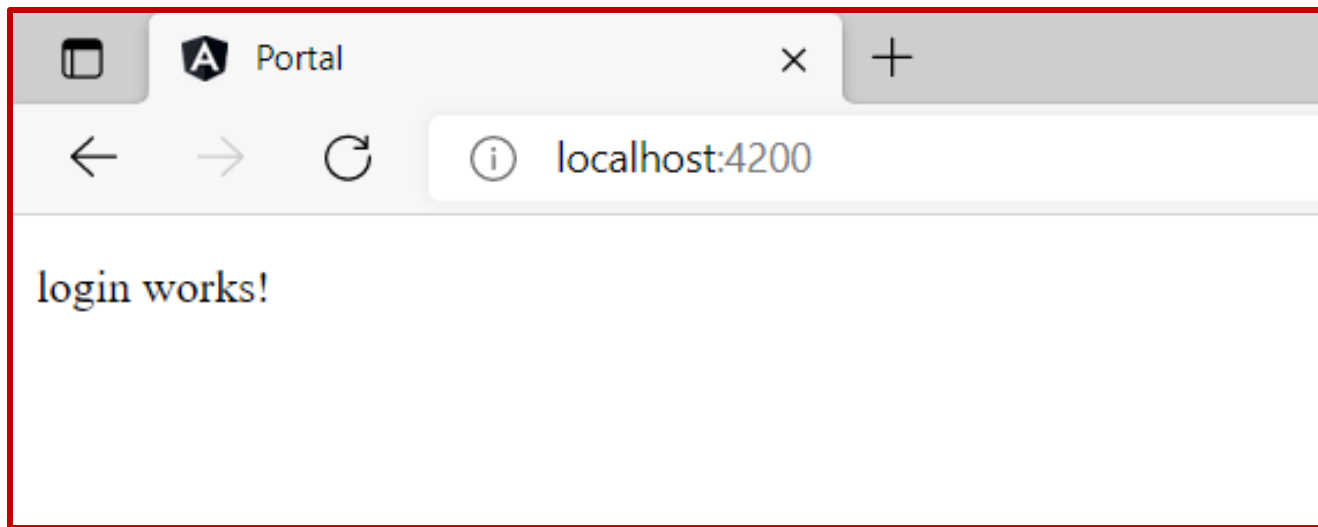
## Moving from one page to another

And In auth-routing.module.ts :

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



## Moving from one page to another



## Chapter 3

1 Moving from one page to another

2 **Create and Validate form**



## Create and Validate form

We will use material angular to install packages :

<https://material.angular.io/>

To use material angular you must install this package :

`ng add @angular/material`



## Create and Validate form

```
PS C:\Users\User\Desktop\portal> ng add @angular/material
i Using package manager: npm
✓ Found compatible package version: @angular/material@12.1.0.
✓ Package information loaded.

The package @angular/material@12.1.0 will be installed and executed.
Would you like to proceed? Yes
✓ Package successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Indigo
/Pink      [ Preview: https://material.angular.io?theme=indigo-pink
]
```



## Create and Validate form

We will work on Login component.

To use form – filed you must add in auth.module.ts, And add MatFormFieldModule in import section.

```
import {MatFormFieldModule} from '@angular/material/  
form-field';
```





## Create and Validate form

To use the input in auth.module.ts, Add MatInputModule in import section.

```
import {MatInputModule} from '@angular/material  
/input';
```



## Create and Validate form

**Form Control:** Form input instance with multiple functionality  
,ex: validation , flag , and data binding.



## Create and Validate form

### Defined form control :

#### Syntax :

// Form Control object receives two parameter  
Variable = new FormControl(**form state** , **Validation Option** )



## Create and Validate form

### In login.component.html

```
<div class ="container">
<div class ="loginCard">
<mat-form-field class ="example-full-width"  >
<mat-label>Email</mat-label>
<input type ="email"  matInput style="font-size: x-
large;"  [formControl]="emailControl"  placeholder=
"Ex. pat@example.com">
</mat-form-field>
</div>
</div>
```



## Create and Validate form

### Add Password :

```
<mat-form-field class="example-full-width" >  
<mat-label>Password</mat-label>  
<input type="password" matInput style="font-size:  
x-large;" [formControl]="passwordControl"  
placeholder="Your Password">  
</mat-form-field>
```



## Create and Validate form

In login.component.ts

```
export class LoginComponent implements OnInit {  
  emailControl = new FormControl('', [Validators.required,  
    Validators.email]);  
  passwordControl=newFormControl('',[Validators.required,  
    Validators.  
    minLength(8)])
```

But you must import **FormsModuls** and  
**ReactiveFormModule** in auth.module.ts in import section :



## Create and Validate form

How to work **\*nglf** ?

In login.component .ts

```
showMessage = false;
```

```
toggleShowMessage()  
{  
    this.showMessage = !this.showMessage;  
}
```



## Create and Validate form

In login.component .html

```
<h4 *ngIf="showMessage">This is message to show</h4>

<!--
  ifshowMessage==true then 'This is message to show'
else '' -->
<button (click) ="toggleShowMessage()">
  toggle</button>
```





## Create and Validate form

Add error message for email:

```
<mat-error *ngIf="emailControl.hasError('email') &&  
!emailControl.hasError('required')">
```

Please enter a valid email address

```
</mat-error>
```

```
<mat-error
```

```
*ngIf="emailControl.hasError('required')">
```

Email is <strong>required</strong>

```
</mat-error>
```



## Create and Validate form

Add error message for Password:

```
<mat-error *ngIf  
="passwordControl.hasError('minlength') && !password  
Control.hasError('required')">  
Password must be more than 8 char  
</mat-error>  
<mat-error *ngIf  
="passwordControl.hasError('required')">  
    Password is <strong>required</strong>  
</mat-error>
```



## Create and Validate form

To get value from form control:

In login.component .html

```
<button type="submit" (click)="submit()">  
  submit</button>
```



## Create and Validate form

In login.component .ts

```
submit() {  
  //Go to Loader const  
  emailValue=this.emailControl.value;  
  const  
  passwordValue=this.passwordControl.value;  
  console  
  .log(emailValue,passwordValue);  
}
```

