Angular

Tahaluf Training Center 2021

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Chapter 4

1. Spinner
2. How to navigate from pages
3. Create and validate form using form group

Make the button disabled if the user enter invalid username or password.

```
<button [disabled]="!emailFormControl.valid
||!passwordFormControl.valid"  mat-raised-button
style="flex:1;width: 100% ;margin-top: 50px;"color-
text="primary">Login</button>
```

Spinner

Add the spinner in login page .

- **First** , you must Import the library in **auth.module.ts** :

```
import {MatProgressSpinnerModule} from
'@angular/material/progress-spinner';
```

- Define the MatProgressSpinnerModule in import

In login.component.html:

```
<mat-progress-spinner class="example-margin"
[color] = "color"[mode] = "mode"[value] = "value" >
  </mat - progress - spinner >
```

Because the color, mode and value are a variable you must define in ts :

In login.component.ts:

```
color: ThemePalette = 'primary';
mode: ProgressSpinnerMode = 'indeterminate';
value = 50;
```

Spinner

- To use ThemePalette , ProgressSpinnerMode import in typescript file :

```
import { ThemePalette } from
'@angular/material/core’;
import { ProgressSpinnerMode } from
'@angular/material/progress-spinner';
```

If the user enter the valid username and password show the loader and if not , show Login :

In login.component.html:

```
 <button [disabled]="!emailFormControl.valid
||!passwordFormControl.valid" mat-raised-button
(click)="submit()"style="flex:1;width: 100% ;margin-
top: 50px;" color="primary">
<ng-container *ngIf="ShowLoader">
<mat-progress-spinner class="example-margin"
[color]="color" [mode]="mode" [value]="value">
</mat-progress-spinner></ng-container>
<ng-container *ngIf="!ShowLoader">Login
</ng-container></button>
```

In login.component.ts:

```
submit(){
    //Go to Loader
    this.ShowLoader = true;
    setTimeout(() => {
        this.ShowLoader = false;
    }
    ), 2000
}
```

Spinner

Another way to add the spinner .
using this library : ng-x spinner.

- **First** you must install this package :

npm ng-x spinner

- Then Import NgxSpinnerModule in in the root module(**AppModule**):

Spinner

- ```
  import { NgxSpinnerModule } from "ngx-spinner";
  ```

- And in import section add NgxSpinnerModule ;

- then : add NgxSpinnerService service wherever you want to use the ngx-spinner (**In Login. components .ts**)
  ```

Spinner

- ```
  import { NgxSpinnerService } from "ngx-spinner";
  ```

- Define an object from this service in the constructure

  ```
  private spinner: NgxSpinnerService
  ```
  ```

Spinner

- If you want to show the spinner use :

```
this.spinner.show()
and to hide the spinner:
this.spinner.hide();
```

Spinner

- Add in app.component .html

```
<ngx-spinner bdColor="rgba(0, 0, 0, 0.8)"
size="medium" color="#fff" type="square-
jelly-box" [fullScreen] = "true" > <p
style="color: white" > Loading... </p></ngx -
spinner >
```

Chapter 4

1. Spinner
2. How to navigate from pages
3. Create and validate form using form group

How to navigate from pages

- To go from login page to register page you must use routing
- Add in login.component .html

```
 </div >
 <p (click)="goToRegisterPage()" style="cursor:
pointer; text-align: center;font-size: 1.7em;margin-
top: 40px; "> Create new account</p>
    </div >
```

How to navigate from pages

In login.component .ts

In constructer parameter

```
private router:Router
```

Add this function

```
goToRegisterPage(){
    this.router.navigate(['register'])
}
```

Chapter 4

1. Spinner
2. How to navigate from pages
3. Create and validate form using form group

Create and validate form using form group

- We will use form group .
- Define an instance of form group.
- **In register.component.ts** : and inside this instance will declared instance of form group

Create and validate form using form group

In register . component . ts:

```
export class RegisterComponent implements OnInit {

registerForm: FormGroup = new FormGroup({
fullName: new FormControl('', [Validators.required]),
email: new FormControl('', [Validators.required,
Validators.email]),
address: new FormControl('', [Validators.required]),
phoneNumber: new FormControl(''),
password: new FormControl('', [Validators.required,
Validators.minLength(8)])
    })
```

Create and validate form using form group

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

```
<form [formGroup] = "registerForm"(submit) = "submit()" >
    <mat-form-field class="example-full-width">
        <mat-label>FullName</mat-label>
        <input matInput formControlName="fullName"
style="font-size: x-large;" placeholder="Full Name ">
<mat-error
*ngIf="registerForm.controls.fullName.hasError('required')
">
fullName is  <strong>required</strong>
</mat-error>
    </mat-form-field>
```

Create and validate form using form group

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

```
<mat-form-field class="example-full-width">
    <mat-label>Email</mat-label>
    <input matInput formControlName="email"
style="font-size: x-large;" placeholder="Ex.
pat@example.com">
        <mat-error
*ngIf="registerForm.controls.email.hasError('email') &&
!registerForm.controls.email.hasError('required')">
Please enter a valid email address
</mat-error>
<mat-error
*ngIf="registerForm.controls.email.hasError('required')
">
Email is <strong>required</strong>
</mat-error>
</mat - form - field >
```

Create and validate form using form group


```
<mat-form-field class="example-full-width">
    <mat-label>Address</mat-label>
    <input matInput formControlName="address"
style="font-size: x-large;" placeholder="Ex. WI
Street">
 <mat-error
*ngIf="registerForm.controls.address.hasError('requ
ired')">
                  address is
<strong>required</strong>
                </mat-error>
</mat-form-field>
```

Create and validate form using form group

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

```
<mat-form-field class="example-full-width">
    <mat-label>Phone Number</mat-label>
    <input matInput formControlName="phoneNumber"
style="font-size: x-large;" placeholder="Ex. 077
988 9870">
    </mat - form - field >
```

Create and validate form using form group

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

```
<mat-form-field class="example-full-width">
<mat-label>Password</mat-label>
<input type="password" matInput
formControlName="password" style="font-size: x-
large;">
        <mat-error
*ngIf="registerForm.controls.password.hasError('minl
ength') &&
!registerForm.controls.password.hasError('required')
">
Please enter a valid password
</mat-error>
 <mat-error
*ngIf="registerForm.controls.password.hasError('requ
ired')">
password is <strong>required</strong>
</mat-error>
</mat - form - field >
```

Exercise

Add new input called Confirm Password and check if the confirm password is equal to password .

Solution

```
<mat - form - field class="example-full-
width" >
<mat-label>Confirm Password</mat-label>
<input type="password" matInput
formControlName="passwordConfirm"
style="font-size: x-large;">
```

Solution

```
 <mat-error
*ngIf="registerForm.controls.passwordConfirm.hasErro
r('minlength') &&
!registerForm.controls.passwordConfirm.hasError('req
uired') &&
registerForm.controls.passwordConfirm.value !=
registerForm.controls.password.value">
password and confirm password does not
<strong>match</strong>

</mat-error>
```

Solution

```
 <mat-error *
ngIf="registerForm.controls.passwo
rdConfirm.hasError('required') " >
    Confirm password is required
</mat - error >
</mat - form - field >
```

Create and validate form using form group


```
<button mat-raised-button
color="primary“
type=“submit”>Register</button>
    </form>
```

Get value from form group :

```
In Register.component.ts
submit(){
    const formValue =
this.registerForm.value;
    console.log(formValue)

}
```

Create and validate form using form group

Make the button disabled if any validation is invalid :

```
<button mat-raised-button color="primary"
[disabled] = "!registerForm.valid" >
Register</button >
```

Exercise

Add in register.component.html paragraph :”
Already have an account ? Login
navigate it to login page

Solution

- To go from Register page to login page you must use routing

- Add in Register.component .html

```
 </div >
 <p (click)="goTologinPage()" style="cursor:
pointer; text-align: center;font-size:
1.7em;margin-top: 40px;
">Already have an account ? Login here </p>
    </div >
```

Solution

Add in register.component .ts

```
constructor(private route: Router)

goToLoginPage(){
    this.route.navigate(['']);

}
```