شركـــة تحالــف الإمـــارات للحــلـــول التـقـنـيـــة ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Angular

Tahaluf Training Center 2021

شركـــة تحالــف الإمـــارات للحــلـــول التـقـنـيـــة ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Chapter 4

Spinner

Using this library: ng-x spinner to add the spinner in your project

first you must install this package:

```
npm i ng-x spinner
```

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

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

Spinner

And in import section add NgxSpinnerModule.

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

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

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 >
```

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

Chapter 4

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'])
}
```

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

Chapter 4

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

In register.component.ts:

```
public myError = (controlName: string, errorName:
string) =>{
  return
this.registerForm.controls[controlName].hasError(errorNa
me);
  }
```

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="myError('fullName',
'required') ">
          Full Name 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>Email</mat-label>
        <input matInput formControlName="email" style="font
-size: x-large;" placeholder="Ex. pat@example.com">
         <mat-error *ngIf="myError('email', 'email') ">
        Please enter a valid email address
      </mat-error>
      <mat-error  *ngIf="myError('email', 'required') ">
        Email 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>Address</mat-label>
        <input matInput formControlName="address" s
tyle="font-size: x-
large;" placeholder="Ex. WI Street">
      <mat-error *ngIf="myError('address',
'required') ">
          address 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>Phone Number</mat-label>
        <input matInput formControlName="phoneNumber" st
yle="font-size: x-
large;" placeholder="Ex. 077 988 9870">
    </mat-form-field>
```

TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Create and validate form using form group

```
    <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="myError('password', 'minlength') ">
          Please enter a valid password
        </mat-error>
        <mat-error *ngIf="myError('password',
'required')">
          password  is <strong>required</strong>
        </mat-error>
    </mat-form-field>
</form>
```

Create and validate form using form group

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

Exercise:

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

Create and validate form using form group

Exercise Solution:

```
<mat-form-field class="example-full-width">
 <mat-label>Confirm Password</mat-label>

        <input (change)="onchang()" matInput formControl
Name="confirmPassword" style="font-size: x-
large;" placeholder="123456">

        <mat-
error *ngIf="formRegister.controls.confirmPassword.hasEr
ror('mismatch')"> Password is not match </mat-error>

        </mat-form-field>
```

Create and validate form using form group

Exercise Solution:

```
 onchang(){

    if (this.formRegister.controls.password.value== this.formRegister.controls.confirmPassword.value) {

    this.formRegister.controls.confirmPassword.setErrors(null);

    }

    else{

    this.formRegister.controls.confirmPassword.setErrors({ mismatch:true});

    }

    }
```

Create and validate form using form group

TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Exercise Solution:

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

</mat-error>
```

Create and validate form using form group

Exercise Solution:

```
 <mat-error *
ngIf="registerForm.controls.passwordConfirm.hasError('re
quired') " >
    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>
```

Create and validate form using form group

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 >
```

Create and validate form using form group

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

Exercise:

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

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

Exercise 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 >
```

Exercise Solution:

Add in register.component.ts

```
constructor(private route: Router)

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

    }
```