Harmony IT Solution

Angular

Tahaluf Training Center 2022

1. Form Group.
2. Spinner Library.
3. Shared Module.

Objective

The Objective of this lecture

- Get to know about the form group and how to use it to validate the group of the form control.
- Install and use the spinner library.
- Understand the concept of the shared module and which component and module will contain it.

Form Group

Overview of Form Group

FormGroups are collections of Form controls that track the values and validity of control instances in the group.

One of the building blocks of angular forms is the **FormGroup**.

Overview of Form Group

FormGroups encapsulate all information related to a collection of Form Controls that addresses this problem.

Each of these controls has a value and a validation status.

Overview of Form Group

Angular forms can be used by importing the FormsModule (for template-driven forms) and ReactiveFormsModule (for reactive forms) from the @angular/forms package.

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms’;

  imports: [
        FormsModule,
    ReactiveFormsModule
  ],
```

Overview of Form Group

Angular forms can be used by importing the FormsModule (for template-driven forms) and ReactiveFormsModule (for reactive forms) from the @angular/forms package.

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms’;

  imports: [
        FormsModule,
    ReactiveFormsModule
  ],
```

Overview of Form Group

FormGroup Demo will be in the register component.

Register form template link:
https://codepen.io/anandaprojapati/pen/GmrwYE

Angular material link:
https://material.angular.io/components/input/overview

Form Group Example

Create a new object from formGroup class in register.component.ts

```
registerform:FormGroup=new FormGroup({
email : new FormControl('', [Validators.required,
Validators.email]),
password: new FormControl('',[Validators.required,
Validators.minLength(8)]),
confirmpassword: new FormControl('',[Validators.required,
Validators.minLength(8)]),
firstname:new FormControl('',[Validators.required]),
lastname:new FormControl('',[Validators.required]),
gender:new FormControl(),
options:new FormControl() })
```

Form Group Example

Update the form in register.component.html :

1. Create a new instance from FormGroup.
2. Make the form tag read from registerform object.
3. Replace the name property for each input in the form
4. by **formControlName = 'theNameOfFormControl'**
5. Add the mat-error section for each input.

Harmony IT Solution

Form Group Example

```
<div class="form_wrapper">
  <div class="form_container">
    <div class="title_container">
    <h2>Responsive Registration Form</h2>
    </div>
    <div class="row clearfix">
      <div class="">
        <form [formGroup]="registerform">
          <div class="row clearfix">
            <div class="col_half">
              <mat-label>First Name</mat-label>
              <div class="input_field"> <span><i aria-hidden="true" class="fa fa-user"></i></span>
                <mat-form-field class="example-full-width">
                  <input type="text" matInput name="name" formControlName="firstname" />
                  <mat-error *ngIf="registerform.controls['firstname'].hasError('required')">
                    First Name is <strong>required</strong>
                  </mat-error>
                </mat-form-field>
              </div>
            </div>
```

Harmony IT Solution

Form Group Example

```
<div class="col_half">
  <mat-label>Last Name</mat-label>
  <div class="input_field"> <span><i aria-hidden="true" class="fa fa-user"></i></span>
    <mat-form-field>
      <input type="text" matInput name="name" formControlName="lastname" />
      <mat-error *ngIf="registerform.controls['lastname'].hasError('required')">
        Last Name is <strong>required</strong>
      </mat-error>
    </mat-form-field></div></div></div>
```

Form Group Example

```
<mat-label>Email</mat-label>
<div class="input_field"> <span aria-hidden="true" class="fa fa-envelope"></span>
<mat-form-field class="example-full-width">
<input type="email" matInput name="email" formControlName="email" />
<mat-error *ngIf="registerform.controls['email'].hasError('required') ">
  Email is <strong>required</strong>
</mat-error>
<mat-error *ngIf="registerform.controls['email'].hasError('email') ">
  Please enter a valid email address
</mat-error>
</mat-form-field>
</div>
```

Form Group Example

```
<mat-label>Password</mat-label>
<div class="input_field"><span><i aria-hidden="true" class="fa fa-lock"></i></span>
  <mat-form-field class="example-full-width">
    <input matInput type="password" name="password" formControlName="password" />
    <mat-error *ngIf="registerform.controls['password'].hasError('minlength')">
      Please enter a strong password
    </mat-error>
    <mat-error *ngIf="registerform.controls['password'].hasError('required')">
      password is <strong>required</strong>
    </mat-error>
  </mat-form-field>
</div>
```

Form Group Example

```
    <input type="radio" id="rd1" value="male" formControlName="gender">
    <label for="rd1">Male</label>
    <input type="radio" id="rd2" value="female" formControlName="gender">
    <label for="rd2">Female</label>
  </div>
<div class="input_field select_option">
    <label>Select a country</label>
    <select formControlName="options">
      <option>Jordan</option>
      <option>UAE</option>
      </select>
```

Form Group Example

```
<button class="btn btn-success example-full-width"
[disabled]="!registerform.valid" (click)="submit()"
type="submit" value="Register">Register
</button>
```

Form Group Example

In register.component.ts

```
submit(){
console.log(this.registerform.value);
}
```

Exercises

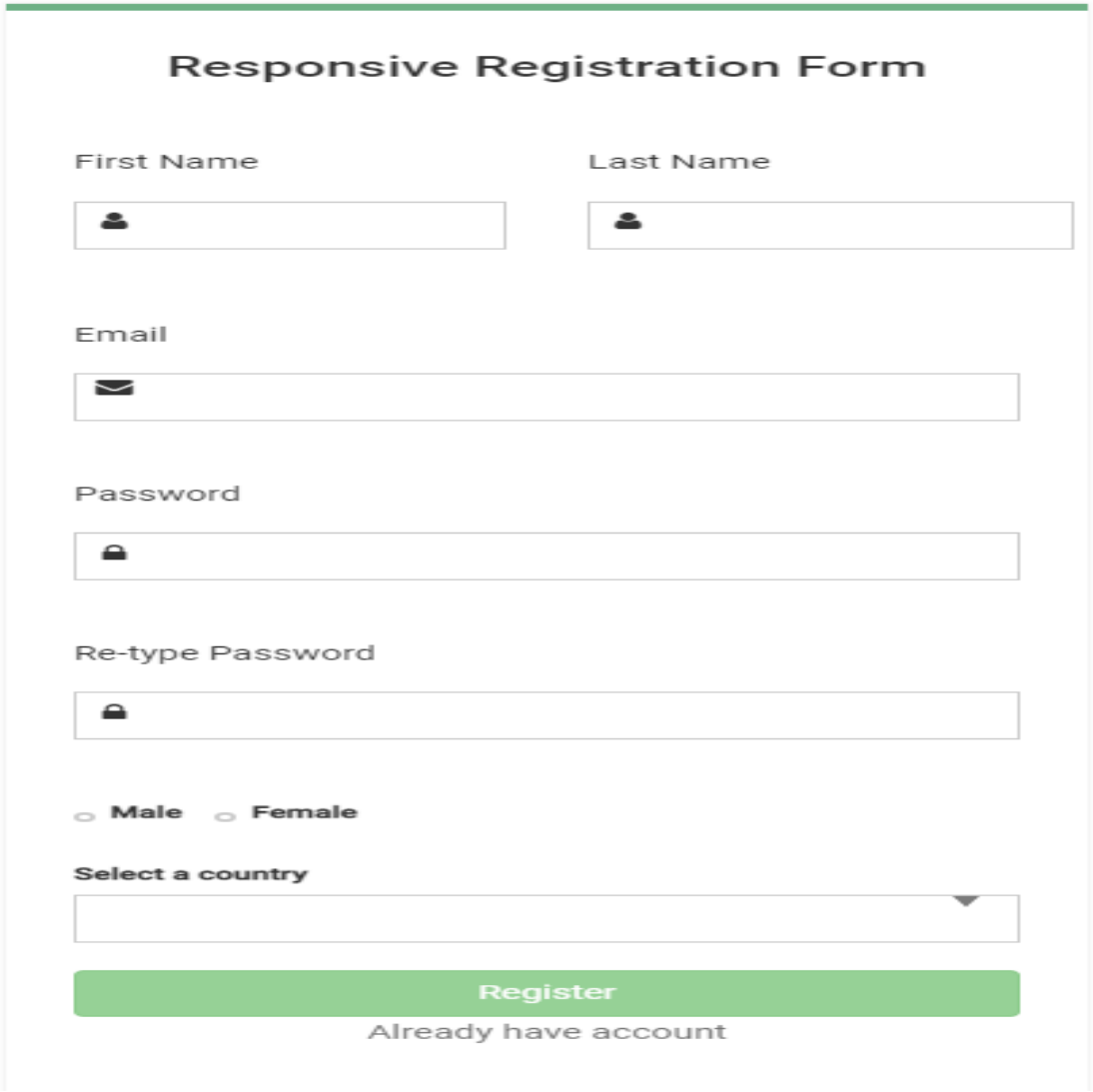
Add a new input called to confirm the password and check whether it's the same as the password.

Exercises

Add the following paragraph "Already have an account" to route it to the login component.

Harmony IT Solution

Form Group Example Result



Spinner Library

Overview of Spinner

An Angular library for loading spinners, ngx spinners shows that data is loading in real-time to the user, and can be used to give the user a notification.

A loader is used to display the loading state of data in an application.

How to add the Spinner to the project

Step one: from npm website (node package manager) searches to ngx-spinner library.

Step two: Install the library using the terminal.
This is the installation command.

```
npm i ngx-spinner
```

How to add the Spinner to the project

Step three: import the NgxSpinnerModule in the root module (App Module).

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

Step four: Adding the template you want in the App Component.

Step five: Add CSS animation files to angular.json config.

How to add the Spinner to the project

Step six: Add NgxSpinnerService service wherever you want to use the ngx-spinner and create an object of this service in the constructor.

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

private spinner: NgxSpinnerService
```

By using this object you can show and hide the spinner using the show and hide.

Spinner Example

Add the spinner in the register component when the user clicks on submit button.

1. In the app.module.ts:

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

And add the `NgxSpinnerModule` in the import array.

Spinner Example

2. Add the template in the app component:

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

Spinner Example

3. Add CSS animation files to angular.json config.

```
"node_modules/ngx-spinner/animations/ball-scale-multiple.css"
```

```
"styles": [
./node_modules/@angular/material/prebuilt-themes/
indigo-pink.css,
"src/styles.css",
"node_modules/ngx-spinner/animations/
ball-scale-multiple.css"
 ],
```

Spinner Example

4. Add NgxSpinnerService service in the register component

In register.component.ts

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

In the constructor :

```
constructor(private router:Router,
private spinner: NgxSpinnerService)
```

Spinner Example

5. Use the show and hide functions in the submit method.

```
submit(){
      console.log(this.registerform.value);
      this.spinner.show();
      setTimeout(() => {
        /** spinner ends after 5 seconds */
        this.spinner.hide();
      }, 5000);
    }
```

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