

Exercise (Instructions): Angular Template-driven Forms Part 1

Objectives and Outcomes

In this exercise you will learn about using Angular template-driven forms in your Angular application. You will also learn to use the Angular material dialog component to show a form in a dialog view in the Angular application. At the end of this exercise you will be able to:

- Add a dialog view using the Angular Material dialog component to your application
- Design a template-driven form in your Angular application.

Triggering the Dialog View

- Add a link to the toolbar in the header component, which when clicked will open a Login form in a dialog view:

```
1 <span class="flex-spacer"></span>
2 <a md-button (click)="openLoginForm()"><span class="fa fa-sign-in fa-lg"
  ></span> Login</a>
```

Adding a Dialog Component

- Next create an Angular component named LoginComponent to your Angular application:

```
1 ng g component login
```

- To make this component into a Dialog component, open login.component.ts and import MdDialog and MdDialogRef as follows:

```
1 import {MdDialog, MdDialogRef} from '@angular/material';
2 |
```

- Then update the template file as follows:

```

1 <md-toolbar md-dialog-title color="primary">
2   Login
3   <span class="flex-spacer"></span>
4   <button md-button md-dialog-close>&times;</button>
5 </md-toolbar>

```

- To make this component be opened from another component, you need to declare this as an `EntryComponent` in the `AppModule`, by adding the following to the `NgModule` decorator:

```

1 @NgModule({
2   . . .
3   entryComponents: [
4     LoginComponent
5   ],
6   . . .
7 })

```

- Now, to trigger the Dialog view of the Login component, open `header.component.ts` and update it as follows:

```

1   . . .
2   import { MdDialog, MdDialogRef } from '@angular/material';
3   import { LoginComponent } from '../login/login.component';
4   . . .
5   export class HeaderComponent implements OnInit {
6     constructor(public dialog: MdDialog ) { }
7     ngOnInit() {
8       openLoginForm() {
9         this.dialog.open(LoginComponent, {width: '500px', height: '450px'});
10      }
11    }
12  }

```

Add in the Login Template-driven Form

- To use the Template-driven forms, import the Angular Forms module into `app.module.ts` by updating it as follows:

```

1 import:import { FormsModule } from '@angular/forms';
2   . . .
3   @NgModule({
4     . . .
5     imports: [
6       . . .
7       FormsModule
8     ],
9   })

```

- Open login.component.html file and add in the form as follows:

```

1  <p>{{ user | json }}</p>
2
3  <form novalidate (ngSubmit)="onSubmit()">
4
5      <md-dialog-content>
6          <p>
7              <md-input-container>
8                  <input mdInput placeholder="Username" type="text" [(ngModel)]="user
9                      .username" name="username" required>
10             </md-input-container>
11             <md-input-container>
12                 <input mdInput placeholder="Password" type="password" [(ngModel)]="user
13                     .password" name="password" required>
14             </md-input-container>
15             <md-checkbox [(ngModel)]="user.remember" name="remember">Remember Me</md
16                 -checkbox>
17         </p>
18     </md-dialog-content>
19     <md-dialog-actions>
20         <span class="flex-spacer"></span>
21         <button md-button md-dialog-close>Cancel</button>
22         <button type="submit" md-button class="background-primary text-floral-white"
23             >Login</button>
24     </md-dialog-actions>
25 </form>

```

- We use the Angular Material support for styling the form elements using the Form element components.
- Then, open login.component.ts file and update its contents as follows:

```

1  . . .
2
3  export class LoginComponent implements OnInit {
4
5      user = {remember: false};
6
7      constructor(public dialogRef: MdDialogRef<LoginComponent>) { }
8
9      ngOnInit() {
10     }
11
12     onSubmit() {
13         console.log("User: ", this.user);
14         this.dialogRef.close();
15     }
16
17 }

```

- Save all the changes and do a Git commit with the message "Template Forms Part 1".

Conclusions

In this exercise you learnt to design a template-driven form and also a Dialog view using the Angular material dialog component.