

## Aula 34 - Storage, Forms e Design Materials

### Docupedia Export

Author: Sílio Leonardo (CtP/ETS)

Date: 07-Jun-2023 13:46

## **Table of Contents**

<b>1 Session e Local Storage</b>	<b>4</b>
<b>2 Forms</b>	<b>7</b>
<b>3 Design Materials</b>	<b>11</b>
<b>4 Exercícios</b>	<b>15</b>

- [Session e Local Storage](#)
- [Forms](#)
- [Design Materials](#)
- [Exercícios](#)

# 1 Session e Local Storage

Todo navegador trás a possibilidade de se armazenar dados localmente. Isto é, no cliente. Isso significa que podemos guardar dados no computador do usuário para uso futuro e que seja disponível de forma global na aplicação. O Local Storage é mantido mesmo que o navegador é fechado. Sistemas como Termos, por exemplo, guardam resultados do usuário no local storage. O Session Storage é limpo ao encerrarmos a sessão, isto é, está associado a uma aba do navegador. Para acessar essa storage no angular é relativamente simples:

## login-page.component.html

```
1  <h1>
2    Login
3  </h1>
4
5  <p>
6    <label>Email/Username</label>
7    <br>
8    <input [(ngModel)]="email">
9  </p>
10
11  <app-password [seePassword]="false" [breakLineOnInput]="true" (valueChanged)="passwordChanged($event)" />
12
13  <p>
14    <button (click)="login()">Logar</button>
15  </p>
16
17  <p>
18    Não possui conta? <a routerLink="newaccount">Crie uma agora mesmo!</a>
19  </p>
20
21  <p>
22    Esqueceu sua senha? <a href="{{'/recover/' + email}}">Recupere agora!</a>
23  </p>
24
25  <router-outlet></router-outlet>
```

**login-page.component.ts**

```
1  import { Component } from '@angular/core';
2  import { Router } from '@angular/router';
3
4  @Component({
5    selector: 'app-login-page',
6    templateUrl: './login-page.component.html',
7    styleUrls: ['./login-page.component.css']
8  })
9  export class LoginPageComponent {
10    email = ""
11    link = ""
12    password = ""
13
14    constructor(private router: Router) { }
15
16    passwordChanged(event : any)
17    {
18      this.password = event
19    }
20
21    login()
22    {
23      // Aqui precisaríamos fazer essa verificação no banco de dados
24      if (this.email == "email@email.com" && this.password == "123")
25      {
26        // Isso evidentemente não é seguro, mas a ideia é bom e será melhorada no futuro
27        sessionStorage.setItem('user', 'pamella');
28        this.router.navigate(["/feed"])
29      }
30    }
31  }
```

**feed-page.component.ts**

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-feed-page',
5    templateUrl: './feed-page.component.html',
6    styleUrls: ['./feed-page.component.css']
7  })
8  export class FeedPageComponent implements OnInit {
9    user: string | null = null
10
11    ngOnInit(): void
12    {
13      this.user = sessionStorage.getItem("user")
14    }
15  }
```

**feed-page.component.html**

```
1  {{user != null ? "Bem-vind@, " + user : "Logue para ver ser feed"}}
```

Todas as outras interações de dados são feitas em servidores feitos futuramente por nós.

## 2 Forms

Você pode usar formulários reativos para criar cadastros mais simples e poderosos. O assunto é longo mas o básico é um sistema de validação usando o FormControl:

### app.module.ts

```
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { NavComponent } from './nav/nav.component';
7  import { LoginComponent } from './login-page/login-page.component';
8  import { HomeComponent } from './home-page/home-page.component';
9  import { NotFoundPageComponent } from './not-found-page/not-found-page.component';
10 import { FeedPageComponent } from './feed-page/feed-page.component';
11 import { CommunityPageComponent } from './community-page/community-page.component';
12 import { NewAccountPageComponent } from './new-account-page/new-account-page.component';
13 import { RecoverPageComponent } from './recover-page/recover-page.component';
14 import { UserPageComponent } from './user-page/user-page.component';
15 import { PasswordComponent } from './password/password.component';
16 import { FormsModule } from '@angular/forms';
17 import { CreatePasswordComponent } from './create-password/create-password.component';
18 import { HttpClientModule } from '@angular/common/http';
19 import { ReactiveFormsModule } from '@angular/forms'; // Added for use ReactiveForms
20
21 @NgModule({
22   declarations: [
23     AppComponent,
24     NavComponent,
25     LoginComponent,
26     HomeComponent,
27     NotFoundPageComponent,
28     FeedPageComponent,
29     CommunityPageComponent,
30     NewAccountPageComponent,
31     RecoverPageComponent,
```

```
32     UserPageComponent,  
33     PasswordComponent,  
34     CreatePasswordComponent  
35   ],  
36   imports: [  
37     BrowserModule,  
38     AppRoutingModule,  
39     FormsModule,  
40     HttpClientModule,  
41     ReactiveFormsModule // Added for use ReactiveForms  
42   ],  
43   providers: [],  
44   bootstrap: [AppComponent]  
45 })  
46 export class AppModule { }
```

#### new-account-page.component.ts

```
1  import { Component } from '@angular/core';  
2  import { FormControl, Validators } from '@angular/forms';  
3  import { CpfService } from '../services/cpf.service';  
4  
5  @Component({  
6    selector: 'app-new-account-page',  
7    templateUrl: './new-account-page.component.html',  
8    styleUrls: ['./new-account-page.component.css']  
9  })  
10 export class NewAccountPageComponent {  
11   cepvalue = ""  
12   ruavalue = ""  
13  
14   email = new FormControl('', [  
15     Validators.required,  
16     Validators.email,  
17     Validators.minLength(4)  
18   ]);  
19  
20   constructor(private cep: CpfService) { }
```



```
21  
22     cepAdded()  
23     {  
24         this.cep.getStreet(this.cepvalue)  
25             .subscribe(x =>  
26                 {  
27                     this.ruavalue = x.logradouro  
28                 })  
29     }  
30 }
```

#### new-account-page.component.html

```
1     <h1>  
2         Nova Conta  
3     </h1>  
4  
5     <label>Email</label>  
6     <br>  
7     <input type="text" [formControl]="email">  
8  
9     <div *ngIf="email.invalid" style="color: red;">  
10         <div *ngIf="email.errors?.['required']">  
11             Name is required.  
12         </div>  
13         <div *ngIf="email.errors?.['minlength']">  
14             Name must be at least 4 characters long.  
15         </div>  
16     </div>  
17  
18     <p>  
19         <label>CEP</label>  
20         <br>  
21         <input [(ngModel)]="cepvalue" (change)="cepAdded()">  
22     </p>  
23  
24     <p>  
25         <label>Rua</label>
```

```
26     <br>
27     <input [(ngModel)]="ruavalue">
28 </p>
29
30 <p>
31     <label>Username</label>
32     <br>
33     <input>
34 </p>
35
36 <app-create-password/>
37
38 <p>
39     <button>Criar Conta</button>
40 </p>
```

### 3 Design Materials

Para começar a deixar a aplicação mais bonita podemos, ao invés de investir pesadamente em html e css, podemos usar componentes feitos por outras pessoas com css feito por outras pessoas, como se fosse um bootstrap construído sobre o Angular. Um das bibliotecas de componentes que existem é o Angular Material. A instalação pode ser feita da seguinte maneira:

```
npm run ng add @angular/material
```

Algumas perguntas são feitas no terminal como tipografia e paleta de cores. Após isso você pode usar todos os componentes do Angular Material, basta adicionar no app module, como cita a documentação do Angular Material:

#### app.module.ts

```
// ...
import { MatSlideToggleModule } from '@angular/material/slide-toggle';

@NgModule ({
  // ...
  imports: [
    // ...
    MatSlideToggleModule
  ]
})
class AppModule {}
```

Vamos fazer um exemplo básico adicionando um botão mais bonito, mas você pode olhar mais componentes em: <https://material.angular.io/components>. Em cada componente, na aba API podemos ver como importar e usar o componente:

Autocomplete  
Badge  
Bottom Sheet  
**Button**  
Button toggle  
Card  
Checkbox  
Chips  
Core  
Datepicker  
Dialog

OVERVIEW   **API**   EXAMPLES

### API reference for Angular Material button

```
import {MatButtonModule} from '@angular/material/button';
```

### Directives

#### MatButton

Material Design button component. Users interact with a button to perform an action. See <https://material.io/components/buttons>

The MatButton class applies to native button elements and captures the appearances for "text button", "outlined button", and "contained button" per the Material Design specification. MatButton additionally captures an additional "flat" appearance, which matches "contained" but without elevation.

Selector: `button[mat-button]` `button[mat-raised-button]` `button[mat-flat-button]` `button[mat-stroked-button]`

Exported as: `matButton`

**app.module.ts**

```

1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { NavComponent } from './nav/nav.component';
7  import { LoginPageComponent } from './login-page/login-page.component';
8  import { HomePageComponent } from './home-page/home-page.component';
9  import { NotFoundPageComponent } from './not-found-page/not-found-page.component';
10 import { FeedPageComponent } from './feed-page/feed-page.component';
11 import { ComunityPageComponent } from './comunity-page/comunity-page.component';
12 import { NewAccountPageComponent } from './new-account-page/new-account-page.component';
13 import { RecoverPageComponent } from './recover-page/recover-page.component';
14 import { UserPageComponent } from './user-page/user-page.component';
15 import { PasswordComponent } from './password/password.component';
16 import { FormsModule } from '@angular/forms';
17 import { CreatePasswordComponent } from './create-password/create-password.component';
18 import { HttpClientModule } from '@angular/common/http';
19 import { ReactiveFormsModule } from '@angular/forms';
20 import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

```

```
21 import { MatButtonModule } from '@angular/material/button'; // Added for use Angular Material Button
22
23
24 @NgModule({
25   declarations: [
26     AppComponent,
27     NavComponent,
28     LoginComponent,
29     HomeComponent,
30     NotFoundPageComponent,
31     FeedPageComponent,
32     ComunityPageComponent,
33     NewAccountPageComponent,
34     RecoverPageComponent,
35     UserPageComponent,
36     PasswordComponent,
37     CreatePasswordComponent
38   ],
39   imports: [
40     BrowserModule,
41     AppRoutingModule,
42     FormsModule,
43     HttpClientModule,
44     ReactiveFormsModule, BrowserAnimationsModule,
45     MatButtonModule // Added for use Angular Material Button
46   ],
47   providers: [],
48   bootstrap: [AppComponent]
49 })
50 export class AppModule { }
```

#### login-page.component.html

```
1 <h1>
2   Login
3 </h1>
4
5 <p>
```

```
6      <label>Email/Username</label>
7      <br>
8      <input [(ngModel)]="email">
9    </p>
10
11    <app-password [seePassword]="false" [breakLineOnInput]="true" (valueChanged)="passwordChanged($event)" />
12
13    <p>
14      <button mat-button color="primary" (click)="login()">Logar</button>
15    </p>
16
17    <p>
18      Não possui conta? <a routerLink="newaccount">Crie uma agora mesmo!</a>
19    </p>
20
21    <p>
22      Esqueceu sua senha? <a href="{{'/recover/' + email}}">Recupere agora!</a>
23    </p>
24
25    <router-outlet></router-outlet>
```

## 4 Exercícios

Explore o Angular Material e os Forms e aplique-os sobre o nosso sistema.