

Memorial descritivo

[chronicrelief-frontend] (20/11)

Basicamente: Como fazer um cliente para uma aplicação mobile com o combo Angular/Ionic.

1. Instalando o necessário:

Supondo que você já possua o node.js instalado...

```
npm install -g @angular/cli  
npm install -g @capacitor/cli  
npm install -g @ionic/cli
```

Instala os clientes das tecnologias que vamos usar, o capacitor não é exatamente necessário, mas no meu computador deu erro por algum motivo e eu resolvi instalar ele dessa forma por precaução

2. Iniciando o projeto

Escolha um diretório apropriado e cole no terminal:

```
ionic start chronicrelief-frontend blank --type=angular
```

```
cd chronicrelief-frontend  
npm install npm install @ionic/angular @angular/router @angul
```

Isso cria um projeto vazio com o ionic e instala algumas dependências extras.

3. Gerando componentes

Basicamente: o ionic tem templates de quase qualquer coisa, então a gente consegue usar ele pra gerar componentes como serviços, paginas e etc

Serviços:

- `ionic generate service services/auth` : Gera o arquivo auth.service.ts dentro da pasta src/app/services/.

- `ionic generate service services/doencas` : Gera o arquivo `doencas.service.ts` dentro da pasta `src/app/services/`.
- `ionic generate service services/documentos` : Gera o arquivo `documentos.service.ts` dentro da pasta `src/app/services/`.
- `ionic generate service services/medicacoes` : Gera o arquivo `medicacoes.service.ts` dentro da pasta `src/app/services/`.
- `ionic generate service services/formulario-ms` : Gera o arquivo `formulario-ms.service.ts` dentro da pasta `src/app/services/`.
- `ionic generate service services/formulario-ps` : Gera o arquivo `formulario-ps.service.ts` dentro da pasta `src/app/services/`. (Você precisará criar esse serviço e o componente correspondente)
- `ionic generate service services/formulario-jp` : Gera o arquivo `formulario-jp.service.ts` dentro da pasta `src/app/services/`. (Você precisará criar esse serviço e o componente correspondente)
- `ionic generate service services/home` : Gera o arquivo `home.service.ts` dentro da pasta `src/app/services/`.
- `ionic generate guard guards/auth` : Gera o arquivo `auth.guard.ts` dentro da pasta `src/app/guards/`.

2. Páginas/Componentes:

- `ionic generate page pages/register` : Gera os arquivos para a página `register` na pasta `src/app/pages/register/`.
- `ionic generate page pages/doencas` : Gera os arquivos para a página `doencas` na pasta `src/app/pages/doencas/`.
- `ionic generate page pages/documentos` : Gera os arquivos para a página `documentos` na pasta `src/app/pages/documentos/`.
- `ionic generate page pages/medicacoes` : Gera os arquivos para a página `medicacoes` na pasta `src/app/pages/medicacoes/`.
- `ionic generate page pages/formulario-ms` : Gera os arquivos para a página `formulario-ms` na pasta `src/app/pages/formulario-ms/`.
- `ionic generate page pages/formulario-ps` : Gera os arquivos para a página `formulario-ps` na pasta `src/app/pages/formulario-ps/`. (Você precisará criar esse serviço e o componente correspondente)

- `ionic generate page pages/formulario-jp` : Gera os arquivos para a página formulario-jp na pasta src/app/pages/formulario-jp/. (Você precisará criar esse serviço e o componente correspondente)
- `ionic generate page pages/home` : Gera os arquivos para a página home na pasta src/app/pages/home/.
- `ionic generate page pages/login` : Gera os arquivos para a página login na pasta src/app/pages/login/.

Ctrl+c, Ctrl+v:

```
ionic generate service services/auth
ionic generate service services/doencas
ionic generate service services/doencas
ionic generate service services/documentos
ionic generate service services/medicacoes
ionic generate service services/formulario-ms
ionic generate service services/formulario-ps
ionic generate service services/formulario-jp
ionic generate service services/home
ionic generate page pages/register
ionic generate page pages/doencas
ionic generate page pages/documentos
ionic generate page pages/medicacoes
ionic generate page pages/formulario-ms
ionic generate page pages/formulario-ps
ionic generate page pages/formulario-jp
ionic generate page pages/home
ionic generate page pages/login
```

Pra criar o guard você precisa escolher "canActivate":

```
ionic generate guard guards/auth
```

4. Código do projeto em si:

É só copiar e colar.

- services/auth.service.ts:

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
@Injectable({
  providedIn: 'root',
})
export class AuthService {
  private apiUrl = 'http://localhost:3000/api/users';

  constructor(private http: HttpClient) {}

  register(userData: { nome: string; email: string; senha: string }) {
    return this.http.post(`${this.apiUrl}/register`, userData)
  }

  login(credentials: { email: string; senha: string }) {
    return this.http.post(`${this.apiUrl}/login`, credentials)
  }
}

```

- pages/register.pages.ts:

```

import { Component, OnInit } from '@angular/core';
import { AuthService } from 'src/app/services/auth.service';

@Component({
  selector: 'app-register',
  template: `
    <ion-content>
      <ion-card>
        <ion-card-header>
          <ion-card-title>Registro</ion-card-title>
        </ion-card-header>
        <ion-card-content>
          <form (ngSubmit)="onSubmit()">
            <ion-input
              [(ngModel)]="nome"
              name="nome"
              label="Nome"
            >

```

```

        placeholder="Seu nome"
    ></ion-input>
    <ion-input
        [(ngModel)]="email"
        name="email"
        type="email"
        label="Email"
        placeholder="seu@email.com"
    ></ion-input>
    <ion-input
        [(ngModel)]="senha"
        name="senha"
        type="password"
        label="Senha"
    ></ion-input>
    <ion-button expand="block" type="submit">Registra
</form>
</ion-card-content>
</ion-card>
</ion-content>
,
}))
export class RegisterPage {
    nome: string = '';
    email: string = '';
    senha: string = '';

    constructor(private authService: AuthService) {}

    onSubmit() {
        this.authService.register({
            nome: this.nome,
            email: this.email,
            senha: this.senha
        }).subscribe({
            next: (response) => {
                // Tratar sucesso do registro
            },

```

```

        error: (error) => {
            // Tratar erro de registro
        }
    });
}
}

```

- services/doencas.service.ts:

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { AuthService } from '../auth.service';
@Injectable({
    providedIn: 'root'
})
export class DoencasService {
    private apiUrl = 'http://localhost:3100/api/doencas';

    constructor(private http: HttpClient, private authService: AuthService) {}

    criarDoenca(doencaData: any) {
        return this.http.post(this.apiUrl, doencaData);
    }

    listarDoencas() {
        return this.http.get(this.apiUrl);
    }

    obterDoenca(id: string) {
        return this.http.get(`${this.apiUrl}/${id}`);
    }

    atualizarDoenca(id: string, doencaData: any) {
        return this.http.put(`${this.apiUrl}/${id}`, doencaData);
    }

    deletarDoenca(id: string) {
        return this.http.delete(`${this.apiUrl}/${id}`);
    }
}

```

```
}  
}
```

- `doencas.page.ts`:

```
import { Component, OnInit } from '@angular/core';  
import { DoencasService } from 'src/app/services/doencas.serv  
  
@Component({  
  selector: 'app-doencas',  
  template: `  
    <ion-content>  
      <ion-list>  
        <ion-item *ngFor="let doenca of doencas">  
          {{ doenca.nome }}  
          <ion-button (click)="deletarDoenca(doenca._id)">Del  
        </ion-item>  
      </ion-list>  
  
      <ion-card>  
        <ion-card-header>  
          <ion-card-title>Nova Doença</ion-card-title>  
        </ion-card-header>  
        <ion-card-content>  
          <form (ngSubmit)="criarDoenca()">  
            <ion-input  
              [(ngModel)]="novaDoenca.nome"  
              name="nome"  
              label="Nome da Doença"  
            ></ion-input>  
            <ion-input  
              [(ngModel)]="novaDoenca.sintomas"  
              name="sintomas"  
              label="Sintomas"  
            ></ion-input>  
            <ion-datetime  
              [(ngModel)]="novaDoenca.dataDiagnostico"  
              name="dataDiagnostico"
```

```

        label="Data do Diagnóstico"
      ></ion-datetime>
      <ion-button type="submit">Adicionar Doença</ion-b
    </form>
  </ion-card-content>
</ion-card>
</ion-content>
,
})
export class DoencasPage implements OnInit {
  doencas: any[] = [];
  novaDoenca = {
    nome: '',
    sintomas: '',
    dataDiagnostico: new Date().toISOString()
  };

  constructor(private doencasService: DoencasService) {}

  ngOnInit() {
    this.listarDoencas();
  }

  listarDoencas() {
    this.doencasService.listarDoencas().subscribe({
      next: (doencas: any) => {
        this.doencas = doencas;
      }
    });
  }

  criarDoenca() {
    this.doencasService.criarDoenca(this.novaDoenca).subscribe(
      next: () => {
        this.listarDoencas();
        // Limpar formulário
        this.novaDoenca = {
          nome: '',

```



```

        sintomas: '',
        dataDiagnostico: new Date().toISOString()
    };
    }
    });
}

deletarDoenca(id: string) {
    this.doencasService.deletarDoenca(id).subscribe({
        next: () => {
            this.listarDoencas();
        }
    });
}
}

```

- app.module.ts:

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { AppComponent } from './app.component';
import { AppRoutingModuleModule } from './app-routing.module';
import { HttpClientModule } from '@angular/common/http';

// Pages (importações de todas as suas páginas)
import { HomePage } from './pages/home/home.page';
import { RegisterPage } from './pages/register/register.page';
import { DoencasPage } from './pages/doencas/doencas.page';
import { DocumentosPage } from './pages/documentos/documentos';
import { MedicacoesPage } from './pages/medicacoes/medicacoes';
import { FormularioMSPage } from './pages/formulario-ms/formu
// import { FormularioPSPage } from './pages/formulario-ps/fo
// import { FormularioJPPage } from './pages/formulario-jp/fo
import { LoginPage } from './pages/login/login.page';

```

```

// Services (importações de todos os seus serviços)
import { AuthService } from './services/auth.service';
import { DoencasService } from './services/doencas.service';
import { DocumentosService } from './services/documentos.serv
import { MedicacoesService } from './services/medicacoes.serv
import { FormularioMSService } from './services/formulario-ms
// import { FormularioPSService } from './services/formulario
// import { FormularioJPService } from './services/formulario
import { HomeService } from './services/home.service';

@NgModule({
  declarations: [
    AppComponent,
    HomePage,
    RegisterPage,
    DoencasPage,
    DocumentosPage,
    MedicacoesPage,
    FormularioMSPage,
    // FormularioPSPage,
    // FormularioJPPage,
    LoginPage,
  ],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModuleM
  providers: [
    { provide: RouteReuseStrategy, useClass: IonicRouteStrate
    AuthService,
    DoencasService,
    DocumentosService,
    MedicacoesService,
    FormularioMSService,
    // FormularioPSService,
    // FormularioJPService,
    HomeService,
    // Adicione outros providers se necessário

```

```

    ],
    bootstrap: [AppComponent],
  })
  export class AppModule {}

```

- services/documentos.service.ts:

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class DocumentosService {
  private apiUrl = 'http://localhost:3100/api/documentos';

  constructor(private http: HttpClient) {}

  uploadDocumento(documento: {
    filename: string,
    contentType: string,
    base64Data: string
  }) {
    return this.http.post(`${this.apiUrl}/upload`, documento)
  }

  listarDocumentos() {
    return this.http.get(this.apiUrl);
  }

  obterDocumento(id: string) {
    return this.http.get(`${this.apiUrl}/${id}`);
  }
}

```

- documentos.page.ts:

```

import { Component, ElementRef, OnInit, ViewChild } from '@angular/core';
import { AlertController } from '@ionic/angular';
import { DocumentosService } from 'src/app/services/documentos';

@Component({
  selector: 'app-documentos',
  template: `
    <ion-content>
      <ion-header>
        <ion-toolbar>
          <ion-title>Documentos</ion-title>
        </ion-toolbar>
      </ion-header>

      <ion-content>
        <ion-button expand="block" (click)="seleccionarArchivo">
          Seleccionar Documento
        </ion-button>

        <input
          type="file"
          #fileInput
          style="display: none"
          (change)="onFileSelected($event)"
        >

        <ion-list>
          <ion-item-sliding *ngFor="let doc of documentos">
            <ion-item>
              <ion-label>
                <h2>{{ doc.filename }}</h2>
                <p>{{ doc.contentType }}</p>
              </ion-label>
            </ion-item>
            <ion-item-options>
              <ion-item-option color="primary" (click)="visualizar">
                <ion-icon name="eye-outline"></ion-icon>
                Visualizar
              </ion-item-option>
            </ion-item-options>
          </ion-item-sliding>
        </ion-list>
      </ion-content>
    `
})
export class DocumentosPage implements OnInit {
  documentos: any[];

  constructor(
    private alertController: AlertController,
    private documentosService: DocumentosService
  ) {}

  ngOnInit() {
    this.documentosService.getDocumentos().subscribe(
      (documentos) => {
        this.documentos = documentos;
      }
    );
  }

  seleccionarArchivo() {
    const fileInput = document.getElementById('fileInput') as HTMLInputElement;
    fileInput.click();
  }

  onFileSelected(event: any) {
    const file = event.target.files[0];
    if (file) {
      this.documentosService.uploadFile(file).subscribe(
        (response) => {
          // Handle successful upload
        },
        (error) => {
          // Handle upload error
        }
      );
    }
  }

  visualizar(doc: any) {
    // Implement visualization logic
  }
}

```

```

        </ion-item-option>
      </ion-item-options>
    </ion-item-sliding>
  </ion-list>
</ion-content>
</ion-content>
,
}))
export class DocumentosPage implements OnInit {
  @ViewChild('fileInput') fileInput!: ElementRef;
  documentos: any[] = [];

  constructor(
    private documentosService: DocumentosService,
    private alertController: AlertController
  ) {}

  ngOnInit() {
    this.listarDocumentos();
  }

  seleccionarArchivo() {
    this.fileInput.nativeElement.click();
  }

  onFileSelected(event: any) {
    const file = event.target.files[0];
    if (file) {
      const reader = new FileReader();
      reader.onload = (e: any) => {
        const base64Data = e.target.result.split(',')[1];
        this.uploadDocumento({
          filename: file.name,
          contentType: file.type,
          base64Data: base64Data
        });
      };
      reader.readAsDataURL(file);
    }
  }
}

```

```

    }
  }

  uploadDocumento(documento: {
    filename: string,
    contentType: string,
    base64Data: string
  }) {
    this.documentosService.uploadDocumento(documento)
      .subscribe({
        next: () => {
          this.listarDocumentos();
        },
        error: (erro) => {
          this.mostrarAlerta('Erro ao fazer upload', erro.mes
        }
      });
  }

  listarDocumentos() {
    this.documentosService.listarDocumentos()
      .subscribe({
        next: (docs: any) => {
          this.documentos = docs;
        },
        error: (erro) => {
          this.mostrarAlerta('Erro', 'Não foi possível listar
        }
      });
  }

  visualizarDocumento(id: string) {
    this.documentosService.obterDocumento(id)
      .subscribe({
        next: (doc: any) => {
          // Implementar lógica de visualização
          // Pode ser um modal ou download
          this.mostrarAlerta('Documento', doc.filename);
        }
      });
  }
}

```

```

        }
    });
}

async mostrarAlerta(titulo: string, mensagem: string) {
    const alert = await this.alertController.create({
        header: titulo,
        message: mensagem,
        buttons: ['OK']
    });

    await alert.present();
}
}

```

- services/medicacoes.service.ts:

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
    providedIn: 'root'
})
export class MedicacoesService {
    private apiUrl = 'http://localhost:3100/api/medicacoes';

    constructor(private http: HttpClient) {}

    criarMedicacao(medicacao: any) {
        return this.http.post(this.apiUrl, medicacao);
    }

    listarMedicacoes() {
        return this.http.get(this.apiUrl);
    }
}

```

- pages/medicacoes.page.ts:

```

import { Component, OnInit } from '@angular/core';
import { AlertController } from '@ionic/angular';
import { MedicacoesService } from 'src/app/services/medicacoes';

@Component({
  selector: 'app-medicacoes',
  template: `
    <ion-content>
      <ion-header>
        <ion-toolbar>
          <ion-title>Medicações</ion-title>
        </ion-toolbar>
      </ion-header>

      <ion-content>
        <ion-card>
          <ion-card-header>
            <ion-card-title>Nova Medicação</ion-card-title>
          </ion-card-header>
          <ion-card-content>
            <form (ngSubmit)="criarMedicacao()">
              <ion-input
                [(ngModel)]="novaMedicacao.nome"
                name="nome"
                label="Nome da Medicação"
              ></ion-input>

              <ion-input
                [(ngModel)]="novaMedicacao.dosagem"
                name="dosagem"
                label="Dosagem"
              ></ion-input>

              <ion-input
                [(ngModel)]="novaMedicacao.frequencia"
                name="frequencia"
                label="Frequência"
              ></ion-input>
            </form>
          </ion-card-content>
        </ion-card>
      </ion-content>
    </ion-content>
  `
})
export class MedicacoesPage implements OnInit {
  constructor(
    private alertController: AlertController,
    private medicacoesService: MedicacoesService
  ) {}

  ngOnInit() {}

  criarMedicacao() {
    // Lógica para criar uma nova medicação
  }
}

```



```

        <ion-datetime
            [(ngModel)]="novaMedicacao.horarioAlarme"
            name="horarioAlarme"
            label="Horário do Alarme"
        ></ion-datetime>

        <ion-datetime
            [(ngModel)]="novaMedicacao.dataInicio"
            name="dataInicio"
            label="Data de Início"
        ></ion-datetime>

        <ion-datetime
            [(ngModel)]="novaMedicacao.dataFim"
            name="dataFim"
            label="Data de Término"
        ></ion-datetime>

        <ion-button type="submit">Adicionar Medicação</ion-button>
    </form>
</ion-card-content>
</ion-card>

<ion-list>
    <ion-item *ngFor="let medicacao of medicacoes">
        <ion-label>
            <h2>{{ medicacao.nome }}</h2>
            <p>Dosagem: {{ medicacao.dosagem }}</p>
            <p>Horário: {{ medicacao.horarioAlarme }}</p>
        </ion-label>
    </ion-item>
</ion-list>
</ion-content>
</ion-content>
    ,
})
export class MedicacoesPage implements OnInit {

```

```

novaMedicacao = {
  nome: '',
  dosagem: '',
  frequencia: '',
  horarioAlarme: new Date().toISOString(),
  dataInicio: new Date().toISOString(),
  dataFim: new Date().toISOString(),
  ocorrencia: {
    tipo: 'semanal',
    intervalos: 1,
    diasDaSemana: ['segunda', 'quarta', 'sexta']
  }
};

medicacoes: any[] = [];

constructor(
  private medicacoesService: MedicacoesService,
  private alertController: AlertController
) {}

ngOnInit() {
  this.listarMedicacoes();
}

criarMedicacao() {
  this.medicacoesService.criarMedicacao(this.novaMedicacao)
    .subscribe({
      next: () => {
        this.listarMedicacoes();
        this.limparFormulario();
      },
      error: (erro) => {
        this.mostrarAlerta('Erro', erro.message);
      }
    });
}

```

```

listarMedicacoes() {
  this.medicacoesService.listarMedicacoes()
    .subscribe({
      next: (medicacoes: any) => {
        this.medicacoes = medicacoes;
      },
      error: (erro) => {
        this.mostrarAlerta('Erro', 'Não foi possível listar
      }
    });
}

limparFormulario() {
  this.novaMedicacao = {
    nome: '',
    dosagem: '',
    frequencia: '',
    horarioAlarme: new Date().toISOString(),
    dataInicio: new Date().toISOString(),
    dataFim: new Date().toISOString(),
    recorrancia: {
      tipo: 'semanal',
      intervalos: 1,
      diasDaSemana: ['segunda', 'quarta', 'sexta']
    }
  };
}

async mostrarAlerta(titulo: string, mensagem: string) {
  const alert = await this.alertController.create({
    header: titulo,
    message: mensagem,
    buttons: ['OK']
  });

  await alert.present();
}
}

```

- services/formulatio-ms.service.ts:

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class FormularioMSService {
  private apiUrl = 'http://localhost:3100/api/diario';

  constructor(private http: HttpClient) {}

  criarRegistroDiario(registro: any) {
    return this.http.post(`${this.apiUrl}/create`, registro);
  }

  listarRegistrosDiarios() {
    return this.http.get(this.apiUrl);
  }
}
```

- pages/fomulario-ms.page.ts:

```
import { Component, OnInit } from '@angular/core';
import { AlertController } from '@ionic/angular';
import { FormularioMSService } from 'src/app/services/formula

@Component({
  selector: 'app-formulario-ms',
  template: `
    <ion-content>
      <ion-header>
        <ion-toolbar>
          <ion-title>Registro Diário MS</ion-title>
        </ion-toolbar>
      </ion-header>
```

```

<ion-content>
  <ion-card>
    <ion-card-header>
      <ion-card-title>Novo Registro Diário</ion-card-ti
    </ion-card-header>
    <ion-card-content>
      <form (ngSubmit)="criarRegistroDiario()">
        <ion-datetime
          [(ngModel)]= "novoRegistro.data"
          name="data"
          label="Data"
        ></ion-datetime>

        <ion-textarea
          [(ngModel)]= "novoRegistro.sintomas"
          name="sintomas"
          label="Sintomas"
        ></ion-textarea>

        <ion-input
          [(ngModel)]= "medicacao"
          name="medicacao"
          label="Medicação"
        ></ion-input>
        <ion-button (click)="adicionarMedicacao()">
          Adicionar Medicação
        </ion-button>

        <ion-textarea
          [(ngModel)]= "novoRegistro.observacoes"
          name="observacoes"
          label="Observações"
        ></ion-textarea>

        <ion-button type="submit">Salvar Registro</ion-
      </form>
    </ion-card-content>
  </ion-card>

```

```

        <ion-list>
            <ion-item *ngFor="let registro of registrosDiarios">
                <ion-label>
                    <h2>{{ registro.data | date }}</h2>
                    <p>Sintomas: {{ registro.sintomas }}</p>
                </ion-label>
            </ion-item>
        </ion-list>
    </ion-content>
</ion-content>
,
}))
export class FormulárioMSPage implements OnInit {
    novoRegistro = {
        data: new Date().toISOString(),
        sintomas: '',
        medicacoes: [] as string[],
        observacoes: ''
    };

    medicacao = '';
    registrosDiarios: any[] = [];

    constructor(
        private formularioMSService: FormulárioMSService,
        private alertController: AlertController
    ) {}

    ngOnInit() {
        this.listarRegistrosDiarios();
    }

    adicionarMedicacao() {
        if (this.medicacao) {
            this.novoRegistro.medicacoes.push(this.medicacao);
            this.medicacao = '';
        }
    }
}

```

```

}

criarRegistroDiario() {
  this.formularioMSService.criarRegistroDiario(this.novoReg
    .subscribe({
      next: () => {
        this.listarRegistrosDiarios();
        this.limparFormulario();
      },
      error: (erro) => {
        this.mostrarAlerta('Erro', erro.message);
      }
    });
}

listarRegistrosDiarios() {
  this.formularioMSService.listarRegistrosDiarios()
    .subscribe({
      next: (registros: any) => {
        this.registrosDiarios = registros;
      }
    });
}

limparFormulario() {
  this.novoRegistro = {
    data: new Date().toISOString(),
    sintomas: '',
    medicacoes: [],
    observacoes: ''
  };
  this.medicacao = '';
}

async mostrarAlerta(titulo: string, mensagem: string) {
  const alert = await this.alertController.create({
    header: titulo,
    message: mensagem,
  });
}

```

```

        buttons: ['OK']
    });

    await alert.present();
}
}

```

- services/home.service.ts:

```

import { Injectable } from '@angular/core';
import { forkJoin, map } from 'rxjs';
import { DocumentosService } from '../documentos.service';
import { DoencasService } from '../doencas.service';
import { FormularioMSService } from '../formulario-ms.service';
import { MedicacoesService } from '../medicacoes.service';

@Injectable({
  providedIn: 'root'
})
export class HomeService {
  constructor(
    private doencasService: DoencasService,
    private medicacoesService: MedicacoesService,
    private documentosService: DocumentosService,
    private formularioMSService: FormularioMSService
    // private formularioPSService: FormularioPSService,
    // private formularioJPService: FormularioJPService
  ) {}

  async getHomeData(limit: number = 5) {
    const doencas$ = this.doencasService.listarDoencas();
    const medicacoes$ = this.medicacoesService.listarMedicaco
    const documentos$ = this.documentosService.listarDocument
    const registrosDiarios$ = this.formularioMSService.listar
    // const registrosSemanais$ = this.formularioPSService.li
    // const registrosMensais$ = this.formularioJPService.lis

    // Combina todas as requisições em uma única

```



```

    return forkJoin({
      doencas: doencas$,
      medicacoes: medicacoes$,
      documentos: documentos$,
      registrosDiarios: registrosDiarios$,
      // registrosSemanais: registrosSemanais$,
      // registrosMensais: registrosMensais$
    }).pipe(
      map(data => {
        // Processa e ordena os dados
        return {
          doencas: this.processData(data.doencas, limit),
          medicacoes: this.processData(data.medicacoes, limit),
          documentos: this.processData(data.documentos, limit),
          registrosDiarios: this.processData(data.registrosDiarios, limit),
          // registrosSemanais: this.processData(data.registrosSemanais, limit),
          // registrosMensais: this.processData(data.registrosMensais, limit),
        };
      })
    );
  }

  private processData(data: any[] | undefined, limit: number) {
    if (!data || !Array.isArray(data)) { // Verifica se a data é um array
      return []; // Retorna um array vazio se não for
    }
    return data
      .sort((a, b) => new Date(b.createdAt).getTime() - new Date(a.createdAt).getTime())
      .slice(0, limit);
  }
}

```

- pages/home.page.ts:

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AuthService } from 'src/app/services/auth.service';
import { HomeService } from 'src/app/services/home.service';

```

```

@Component({
  selector: 'app-home',
  template: `
    <ion-header>
      <ion-toolbar>
        <ion-title>
          Chronic Relief
        </ion-title>
      </ion-toolbar>
    </ion-header>

    <ion-content>
      <!-- Resumo Rápido -->
      <ion-card>
        <ion-card-header>
          <ion-card-title>Resumo</ion-card-title>
        </ion-card-header>
        <ion-card-content>
          <ion-grid>
            <ion-row>
              <ion-col size="4">
                <div class="stat-box">
                  <h3>{{ totalMedicacoes }}</h3>
                  <p>Medicações</p>
                </div>
              </ion-col>
              <ion-col size="4">
                <div class="stat-box">
                  <h3>{{ totalDoencas }}</h3>
                  <p>Doenças</p>
                </div>
              </ion-col>
              <ion-col size="4">
                <div class="stat-box">
                  <h3>{{ totalDocumentos }}</h3>
                  <p>Documentos</p>
                </div>
              </ion-col>
            </ion-row>
          </ion-grid>
        </ion-card-content>
      </ion-card>
    </ion-content>
  `
})

```

```

        </ion-col>
      </ion-row>
    </ion-grid>
  </ion-card-content>
</ion-card>

<!-- Medicações -->
<ion-card>
  <ion-card-header>
    <ion-card-title>
      Medicações Recentes
      <ion-button fill="clear" size="small" routerLink=
        Ver Todas
        <ion-icon name="arrow-forward"></ion-icon>
      </ion-button>
    </ion-card-title>
  </ion-card-header>
  <ion-card-content>
    <ion-list>
      <ion-item *ngFor="let medicacao of medicacoes">
        <ion-label>
          <h2>{{ medicacao.nome }}</h2>
          <p>{{ medicacao.dosagem }} - {{ medicacao.hor
        </ion-label>
        <ion-note slot="end" color="primary">
          {{ medicacao.createdAt | date }}
        </ion-note>
      </ion-item>
    </ion-list>
  </ion-card-content>
</ion-card>

<!-- Doenças -->
<ion-card>
  <ion-card-header>
    <ion-card-title>
      Doenças
      <ion-button fill="clear" size="small" routerLink=

```

```

        Ver Todas
        <ion-icon name="arrow-forward"></ion-icon>
    </ion-button>
</ion-card-title>
</ion-card-header>
<ion-card-content>
    <ion-list>
        <ion-item *ngFor="let doenca of doencas">
            <ion-label>
                <h2>{{ doenca.nome }}</h2>
                <p>Diagnosticado em: {{ doenca.dataDiagnosticado }}</p>
            </ion-label>
        </ion-item>
    </ion-list>
</ion-card-content>
</ion-card>

<!-- Registros Diários -->
<ion-card>
    <ion-card-header>
        <ion-card-title>
            Registros Diários Recentes
            <ion-button fill="clear" size="small" routerLink="/registros-diarios">
                Ver Todos
                <ion-icon name="arrow-forward"></ion-icon>
            </ion-button>
        </ion-card-title>
    </ion-card-header>
    <ion-card-content>
        <ion-list>
            <ion-item *ngFor="let registro of registrosDiarios">
                <ion-label>
                    <h2>{{ registro.data | date }}</h2>
                    <p>{{ registro.sintomas }}</p>
                </ion-label>
            </ion-item>
        </ion-list>
    </ion-card-content>
</ion-card>

```

```

</ion-card>

<!-- Documentos -->
<ion-card>
  <ion-card-header>
    <ion-card-title>
      Documentos Recentes
      <ion-button fill="clear" size="small" routerLink=
        Ver Todos
        <ion-icon name="arrow-forward"></ion-icon>
      </ion-button>
    </ion-card-title>
  </ion-card-header>
  <ion-card-content>
    <ion-list>
      <ion-item *ngFor="let documento of documentos">
        <ion-label>
          <h2>{{ documento.filename }}</h2>
          <p>{{ documento.contentType }}</p>
        </ion-label>
        <ion-note slot="end" color="primary">
          {{ documento.createdAt | date }}
        </ion-note>
      </ion-item>
    </ion-list>
  </ion-card-content>
</ion-card>

<!-- FAB para Adicionar -->
<ion-fab vertical="bottom" horizontal="end" slot="fixed"
  <ion-fab-button>
    <ion-icon name="add"></ion-icon>
  </ion-fab-button>
  <ion-fab-list side="top">
    <ion-fab-button (click)="navegarPara('/medicacoes/nova')">
      <ion-icon name="medkit"></ion-icon>
    </ion-fab-button>
    <ion-fab-button (click)="navegarPara('/doencas/nova')">

```

```

        <ion-icon name="fitness"></ion-icon>
      </ion-fab-button>
      <ion-fab-button (click)="navegarPara('/documentos/u
        <ion-icon name="document"></ion-icon>
      </ion-fab-button>
      <ion-fab-button (click)="navegarPara('/diario/novo'
        <ion-icon name="calendar"></ion-icon>
      </ion-fab-button>
    </ion-fab-list>
  </ion-fab>
</ion-content>
`,
styles: [`
  .stat-box {
    text-align: center;
    padding: 10px;
    border-radius: 8px;
    background: var(--ion-color-light);
  }

  .stat-box h3 {
    margin: 0;
    font-size: 24px;
    font-weight: bold;
    color: var(--ion-color-primary);
  }

  .stat-box p {
    margin: 5px 0 0;
    font-size: 14px;
    color: var(--ion-color-medium);
  }
`]
}))
export class HomePage implements OnInit {
  medicacoes: any[] = [];
  doencas: any[] = [];
  documentos: any[] = [];

```

```

registrosDiarios: any[] = [];
registrosSemanais: any[] = [];
registrosMensais: any[] = [];

totalMedicacoes = 0;
totalDoencas = 0;
totalDocumentos = 0;

itemLimit = 5; // Limite de itens por seção

constructor(
  private homeService: HomeService,
  private router: Router,
  private authService: AuthService
) {}

ngOnInit() {
  this.carregarDados();
}

async carregarDados() {
  (await this.homeService.getHomeData(this.itemLimit))
    .subscribe({
      next: (data: any) => {
        this.medicacoes = data.medicacoes;
        this.doencas = data.doencas;
        this.documentos = data.documentos;
        this.registrosDiarios = data.registrosDiarios;
        this.registrosSemanais = data.registrosSemanais;
        this.registrosMensais = data.registrosMensais;

        // Atualiza totais
        this.totalMedicacoes = this.medicacoes.length;
        this.totalDoencas = this.doencas.length;
        this.totalDocumentos = this.documentos.length;
      },
      error: (error) => {
        console.error('Erro ao carregar dados:', error);
      }
    });
}

```

```

        }
    });
}

navegarPara(rota: string) {
    this.router.navigate([rota]);
}

doRefresh(event: any) {
    this.carregarDados();
    setTimeout(() => {
        event.target.complete();
    }, 1000);
}
}

```

- guards/auth.guard.ts:

```

import { Injectable } from '@angular/core';
import { CanActivate, CanActivateFn, Router } from '@angular/router';
import { AuthService } from '../services/auth.service';

@Injectable({
    providedIn: 'root'
})
export class AuthGuard implements CanActivate {
    constructor(
        private authService: AuthService,
        private router: Router
    ) {}

    canActivate(): boolean {
        if (this.authService.isAuthenticated()) {
            return true;
        }

        this.router.navigate(['/login']);
    }
}

```



```
        return false;  
    }  
}
```