

Desenvolvimento Web Front-End

Fabrício Tonetto Londero

Aula 08 – Consumir API



Na API node

- npm i cors

```
1 import 'dotenv/config';
2 import express from 'express';
3 import mongoose from 'mongoose';
4 import cors from 'cors';
5
6 const app = express();
7 app.use(express.json()); // API aceita JSON
8
9 app.use(cors({
10   origin: 'http://localhost:4200', // URL do Angular
11   methods: ['GET', 'POST', 'PUT', 'PATCH', 'DELETE', 'OPTIONS'],
12   allowedHeaders: ['Content-Type', 'Authorization'],
13   credentials: false
14 }));
15
```

app.config.ts

```
You, 9 minutes ago | 1 author (You)
1 import { ApplicationConfig, provideBrowserGlobalErrorListeners, provideZoneChangeDetection } from '@angular/core';
2 import { provideRouter } from '@angular/router';
3 import { provideHttpClient, withFetch } from '@angular/common/http';
4
5
6 import { routes } from './app.routes';
7
8 export const appConfig: ApplicationConfig = {
9   providers: [
10     provideBrowserGlobalErrorListeners(),
11     provideZoneChangeDetection({ eventCoalescing: true }),
12     provideHttpClient(withFetch()),
13     provideRouter(routes)
14   ]
15 };
You, 2 weeks ago • initial commit
```

AlunosService – parte 1

```
1 import { Injectable, inject } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4
5 export interface Aluno {
6   _id?: string;
7   nome: string;
8   idade: number;
9   curso: string;
10  notas: number[];
11 }
12
13 @Injectable({
14   providedIn: 'root'
15 })
```

AlunosService – parte 2

```
16 export class AlunosService {
17     private http = inject(HttpClient);
18     private base = `http://localhost:3000/alunos`;
19
20     listar(): Observable<Aluno[]> {
21         return this.http.get<Aluno[]>(this.base);
22     }
23     buscarPorId(id: string): Observable<Aluno> {
24         return this.http.get<Aluno>(`${this.base}/${id}`);
25     }
26     criar(aluno: Aluno): Observable<Aluno> {
27         console.log(aluno);
28         return this.http.post<Aluno>(this.base, aluno);
29     }
}
```

AlunosService – parte 3

```
30  atualizar(id: string, aluno: Partial<Aluno>): Observable<Aluno> {
31      return this.http.patch<Aluno>(`${this.base}/${id}`, aluno);
32  }
33  excluir(id: string) {
34      return this.http.delete(`${this.base}/${id}`);
35  }
36  }
37
```

Componente Alunos – parte 1

```
1 import { Component, OnInit, inject } from '@angular/core';
2 import { AlunosService, Aluno } from '../Services/alunos-service';
3 import { FormsModule } from '@angular/forms';
4
5 @Component({
6   selector: 'app-alunos',
7   imports: [FormsModule],
8   templateUrl: './alunos.html',
9   standalone: true,
10  styleUrls: ['./alunos.css']
11 })
12 export class Alunos {
13   private api = inject(AlunosService);
14 }
```

Componente Alunos – parte 2

```
15  alunos: Aluno[] = [];  
16  carregando = false;  
17  salvando = false;  
18  erro = '';  
19  
20  // bindings do form  
21  nome = '';  
22  idade: number | null = null;  
23  curso = '';  
24  notasCsv = ''; // ex.: "7,8,9"  
25  
26  ngOnInit() { this.carregar(); }  
27
```

Componente Alunos – parte 3

```
28  carregar() {  
29    this.carregando = true;  
30    this.api.listar().subscribe({  
31      next: xs => { this.alunos = xs; this.carregando = false; },  
32      error: e => { this.erro = e.message ?? 'Falha ao carregar';  
33        this.carregando = false; }  
34    });  
35  }  
36
```

Componente Alunos – parte 4

```
37  criar() {  
38    if (!this.nome || this.idade == null || !this.curso) return;  
39  
40    const notas = this.notasCsv  
41      .split(',') .map(s => Number(s.trim()))  
42      .filter(n => !Number.isNaN(n));  
43  
44    const aluno: Aluno = {  
45      nome: this.nome,  
46      idade: Number(this.idade),  
47      curso: this.curso, notas  
48    };  
49  
50    this.salvando = true;  
51    this.api.criar(aluno).subscribe({  
52      next: _ => {  
53        this.nome = ''; this.idade = null; this.curso = '';  
54        this.notasCsv = '';  
55        this.salvando = false; this.carregar();  
56      },  
57      error: e => { this.erro = e.message ?? 'Falha ao criar';  
58        this.salvando = false; }  
59    });  
60  }
```

You, 4 minutes ago • consumindo API ...

Componente Alunos – parte 5

```
excluir(id?: string) {  
  if (!id) return;  
  this.api.excluir(id).subscribe({  
    next: _ => this.carregar(),  
    error: e => this.erro = e.message ?? 'Falha ao excluir'  
  });  
}  
  
}
```

Componente Alunos – HTML – Parte 1

```
1 <h1>Alunos</h1>
2 <section> You, 6 minutes ago • consumindo API ...
3   <h2>Novo aluno</h2>
4   <form (ngSubmit)="criar()">
5     <div>
6       <label>Nome</label>
7       <input [(ngModel)]="nome" name="nome" required />
8     </div>
9
10    <div>
11      <label>Idade</label>
12      <input type="number" [(ngModel)]="idade" name="idade" required />
13    </div>
14
15    <div>
16      <label>Curso</label>
17      <input [(ngModel)]="curso" name="curso" required />
18    </div>
19
20    <div>
21      <label>Notas (ex.: 7,8,9)</label>
22      <input [(ngModel)]="notasCsv" name="notasCsv" />
23    </div>
```

Componente Alunos – HTML – Parte 2

```
25     <button type="submit" [disabled]="salvando">Salvar</button>
26 </form>
27
28     @if (erro) { <p style="color: red">{{ erro }}</p> }
29 </section>
30
31 <hr>
32
33 <section>
34     <h2>Lista</h2>
35     <button (click)="carregar()" [disabled]="carregando">
36         Recarregar</button>
37     @if (carregando) {
38         <p>Carregando...</p>
39     } @else {
40         @if (alunos.length === 0) {
41             <p>Nenhum aluno.</p>
42         } @else {
43             <ul>
```

Componente Alunos – HTML – Parte 3

```
44     @for (a of alunos; track a._id) {  
45         <li>  
46             <strong>{{ a.nome }}</strong> – {{ a.idade }}  
47             – {{ a.curso }}  
48             <button (click)="excluir(a._id!)">Excluir</button>  
49         </li>  
50     }  
51 </ul>  
52 }  
53 }  
54 </section>  
55
```

Resultado

Alunos

Novo aluno

Nome

Idade

Curso

Notas (ex.: 7,8,9)

Lista

- **Maria Silva** — 20 — Engenharia
- **João Souza** — 22 — Medicina
- **Ana Lima** — 19 — Direito
- **Fabício** — 35 — ADS
- **Felipe** — 35 — ADS
- **Pedro** — 22 — ADS

Exercício

- Implemente o Editar aluno