

# Alap lépések

## Új projekt létrehozása(CSS-sel és Routinggal):

`ng new iskola --style=css --routing=true`

SSR & SSG – **NO**

Zoneless application – **NO**

AI Tools - **NO**

---

## Belépés a projekt mappába

**(ADDIG NE ADJATOK KI MÁS PARANCST!)**

`cd iskola`

---

## Bootstrap beállítása

`npm install bootstrap`

`angular.json` módosítása

```
"styles": [  
  "node_modules/bootstrap/dist/css/bootstrap.min.css",  
  "src/styles.css"  
]
```

---

## Felesleges elemek eltávolítása

`app.html felesleges` elemeinek törlése, csak ez a sor maradjon benne!

```
<router-outlet />
```

---

## HTTP kérés kezelése

`src/app/app.config.ts`

hozzáadjuk a `provideHttpClient();` sort, a mintán látható módon, felül automatikusan jelenik meg egy új sor:

```
import { ApplicationConfig, provideBrowserGlobalErrorListeners,  
provideZoneChangeDetection } from '@angular/core';  
import { provideRouter } from '@angular/router';  
  
import { routes } from './app.routes';  
import { provideHttpClient } from '@angular/common/http';  
  
export const appConfig: ApplicationConfig = {  
  providers: [  

```

```
provideBrowserGlobalErrorListeners(),
provideZoneChangeDetection({ eventCoalescing: true }),
provideRouter(routes),
provideHttpClient() //Ezzel bővítjük, BE KELL GÉPELNI!
]
};
```

## Komponensek létrehozása

**Az alábbi összes parancsot, a terminálban kell kiadni!**

### Service-k létrehozása

(a generáláskor a \*.spec.ts állomány is létrejön, de ez NEM HIBA!):

**A service fájlok kezelik a REST API kéréseket, az útvonalak megadására ügyeljenek!**

*ng generate service services/api*

vagy rövidebben:

*ng g s services/api*

### Model fájlok létrehozása

- Létrehozzuk a modeleket, tartalmazó komponenseket a CRUD feladatokhoz:

*ng generate interface models/diak*

- majd a legördülő menüs elemhez

*ng generate interface models/osztaly*

### CRUD komponensek létrehozása

- Diák listázó komponens elkészítése:

*ng generate component components/diakok-lista.component*

- Új tanuló létrehozó felület elkészítése **(ITT LEHET HASZNÁLNI A RESPONSIVE ŰRLAPOS FELADAT KÓDJÁT MAJD FRONTENDEN)**

*ng generate component components/diak-letrehozas.components*

- Tanuló szerkesztése felület elkészítése **(ITT LEHET HASZNÁLNI A RESPONSIVE ŰRLAPOS FELADAT KÓDJÁT MAJD FRONTENDEN)**

*ng generate component components/diak-szerkesztes.component*

- HIBA komponens létrehozása

*ng generate component components/hiba*

# Komponensek tartalmának elkészítése

## Service komponensek elkészítése, útvonalak megadásával

a létrejött **src/app/services/api.ts** állomány tartalmát módosítjuk az alábbira:

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

@Injectable({
  providedIn: 'root'
})
export class ApiService {
  private readonly baseUrl = 'http://localhost/iskola';

  constructor(private http: HttpClient) {}

  // -----
  // OSZTÁLYOK (READ)
  // -----
  getOsztalyok(): Observable<any> {
    return this.http.get<any>(`${this.baseUrl}/osztalyok_read.php`);
  }

  // -----
  // DIÁKOK (CRUD)
  // -----
  getDiakok(): Observable<any> {
    return this.http.get<any>(`${this.baseUrl}/tanulok_read.php`);
  }

  createDiak(payload: any): Observable<any> {
    return this.http.post<any>(`${this.baseUrl}/tanulok_create.php`, payload);
  }

  updateDiak(payload: any): Observable<any> {
    return this.http.put<any>(`${this.baseUrl}/tanulok_update.php`, payload);
  }

  deleteDiak(id: number): Observable<any> {
    return
this.http.delete<any>(`${this.baseUrl}/tanulok_delete.php?id=${id}`);
  }
}
```

## Modellek elkészítése

módosítjuk a **src/app/models/diak.ts** állomány tartalmát a következőre!

```
export interface Diak {  
  id: number;  
  vezeteknev: string;  
  keresztnév: string;  
  email: string;  
  osztaly_id: number;  
  osztaly_nev: string;  
  szuletesi_datum: string | null;  
}
```

módosítjuk a **src/app/models/osztaly.ts** állomány tartalmát a következőre!

```
export interface Osztaly {  
  id: number;  
  nev: string;  
}
```

## Tanulók listázása & törlése funkciók elkészítése

Módosítjuk a **diakok-lista.component.ts** tartalmát az alábbira:

**Gyakori hiba: fontos a CommonModule importálása, alapból nem tartalmazza, nem fog működni az \*ngFor elem a html részben!**

```
import { Component, OnInit } from '@angular/core';  
import { ApiService } from '../services/api';  
import { CommonModule } from '@angular/common';  
import { RouterLink } from '@angular/router';  
  
@Component({  
  selector: 'app-diakok-lista',  
  standalone: true,  
  imports:[CommonModule,RouterLink],  
  templateUrl: './diakok-lista.component.html',  
  styleUrls: ['./diakok-lista.component.css']  
})  
export class DiakokListaComponent implements OnInit {  
  
  diakok: any[] = [];  
  loading = false;  
  error = '';  
  
  constructor(private api: ApiService) {}  
}
```

```

ngOnInit(): void {
  this.loadDiakok();
}

loadDiakok(): void {
  this.loading = true;
  this.error = '';

  this.api.getDiakok().subscribe({
    next: (res) => {
      this.diakok = res.data;
      this.loading = false;
    },
    error: () => {
      this.error = 'Hiba történt a diákok betöltésekor.';
      this.loading = false;
    }
  });
}

//Tanuló törlése, így ehhez a funkcióhoz nem kell külön komponens
deleteDiak(id: number): void {
  if (!confirm('Biztosan törlöd a kiválasztott diákot?')) return;

  this.api.deleteDiak(id).subscribe({
    next: () => {
      // újratöltjük a listát, hogy azonnal eltűnjön a sor
      this.loadDiakok();
    },
    error: () => {
      alert('Hiba történt a törlés során.');
```

A [src/app/components/diakok-lista.component/diakok-lista.component.html](#) fájlt módosítjuk, az alábbiak szerint

```

<div class="container mt-4">

  <h2 class="mb-3">Diákok listája</h2>

  <table class="table table-striped table-bordered table-hover">
    <thead class="table-dark">
      <tr>
        <th>#</th>
      </tr>
    </thead>
  </table>
</div>
```

```

        <th>Vezetéknév</th>
        <th>Keresztnév</th>
        <th>Email</th>
        <th>Osztály</th>
        <th>Születési dátum</th>
        <th style="width: 160px;">Műveletek</th>
    </tr>
</thead>

<tbody>
    <tr *ngFor="let d of diakok; let i = index">
        <td>{{ i + 1 }}</td>
        <td>{{ d.vezeteknev }}</td>
        <td>{{ d.keresztnev }}</td>
        <td>{{ d.email }}</td>
        <td>{{ d.osztaly_nev }}</td>
        <td>{{ d.szuletesi_datum }}</td>
        <td>
            <button class="btn btn-sm btn-warning me-2">
                Szerkesztés
            </button>

            <button class="btn btn-sm btn-danger" (click)="deleteDiak(d.id)">
                Törlés
            </button>
        </td>
    </tr>
</tbody>
</table>
</div>

```

## Tanulók listázása & törlése routing elkészítése

**src/app/app.routes.ts** fájlba felvesszük a listázó aloldalt kezdőoldalnak, routes tömb bővítésével:

**{ path: '', component: DiakokListaComponent }** – BE KELL GÉPELNI, különben nem jön létre az import rész!

```

import { Routes } from '@angular/router';
import { DiakokListaComponent } from '../components/diakok-
lista.component/diakok-lista.component';

```

```
export const routes: Routes = [
  { path: '', component: DiakokListaComponent }
];
```

## Új tanuló létrehozó felület létrehozása

**fontos a CommonModule és FormsModule importálása, alpból nem tartalmazza**

**src/app/components/diak-letrehozas.component/diak-letrehozas.component.ts**

```
import { Component, OnInit } from '@angular/core';
import { CommonModule } from '@angular/common';

import { ApiService } from '../../services/api';
import { Osztaly } from '../../models/osztaly';
import { FormsModule } from '@angular/forms';

@Component({
  selector: 'app-diak-letrehozas',
  standalone: true,
  imports: [CommonModule, FormsModule],
  templateUrl: './diak-letrehozas.component.html',
  styleUrls: ['./diak-letrehozas.component.css']
})
export class DiakLetrehozasComponent implements OnInit {

  // Osztályok a legördülőhöz
  osztalyok: Osztaly[] = [];

  // Egyszerű "úrlapmodell" (template-driven irány, FormsModule majd később)
  form = {
    vezeteknev: '',
    keresztnév: '',
    email: '',
    osztaly_id: 0,
    szuletesi_datum: '' // üres string is ok (backend null-ként kezeli, ha
    üresen küldjük vagy kihagyjuk)
  };

  constructor(private api: ApiService) {}

  ngOnInit(): void {
    this.loadOsztalyok();
  }

  loadOsztalyok(): void {
```

```

this.api.getOsztalyok().subscribe({
  next: (res) => {
    this.osztalyok = res.data;
  },
  error: () => {
    alert('Nem sikerült betölteni az osztályokat.');
```

```

  }
});
}

createDiak(): void {
  // Minimál validáció (HTML-ben is lesz required)
  if (!this.form.vezeteknev || !this.form.keresztnev || !this.form.email ||
this.form.osztaly_id <= 0) {
    alert('Töltsd ki a kötelező mezőket!');
    return;
  }

  // Payload összeállítás: ha a dátum üres, küldjük null-ként
  const payload = {
    vezeteknev: this.form.vezeteknev.trim(),
    keresztnev: this.form.keresztnev.trim(),
    email: this.form.email.trim(),
    osztaly_id: Number(this.form.osztaly_id),
    szuletesi_datum: this.form.szuletesi_datum.trim() === '' ? null :
this.form.szuletesi_datum.trim()
  };

  this.api.createDiak(payload).subscribe({
    next: () => {
      alert('Sikerés mentés!');
```

```

      // Űrlap ürítése
      this.form = {
        vezeteknev: '',
        keresztnev: '',
        email: '',
        osztaly_id: 0,
        szuletesi_datum: ''
      };
    },
    error: () => {
      alert('Hiba történt a mentés során (email lehet, hogy már foglalt).');
```

```

    }
  });
}
}

```



<src/app/components/diak-letrehozas.component/diak-letrehozas.component.html>

```
<div class="container mt-4">

  <div class="card bg-dark text-light border-secondary">
    <div class="card-body">

      <h2 class="card-title mb-4">Új diák felvétele</h2>

      <form (ngSubmit)="createDiak()">

        <!-- Vezetéknév + Keresztnév -->
        <div class="row g-3 mb-3">
          <div class="col-md-6">
            <div class="form-floating">
              <input
                type="text"
                class="form-control"
                id="vezeteknev"
                placeholder="Vezetéknév"
                required
                name="vezeteknev"
                [(ngModel)]="form.vezeteknev"
              />
              <label for="vezeteknev">Vezetéknév</label>
            </div>
          </div>

          <div class="col-md-6">
            <div class="form-floating">
              <input
                type="text"
                class="form-control"
                id="keresztnev"
                placeholder="Keresztnév"
                required
                name="keresztnev"
                [(ngModel)]="form.keresztnev"
              />
              <label for="keresztnev">Keresztnév</label>
            </div>
          </div>
        </div>

        <!-- Email -->
        <div class="form-floating mb-3">
          <input
```

```

        type="email"
        class="form-control"
        id="email"
        placeholder="Email"
        required
        name="email"
        [(ngModel)]="form.email"
    />
    <label for="email">Email</label>
</div>

<!-- Osztály -->
<div class="form-floating mb-3">
    <select
        class="form-select"
        id="osztaly"
        required
        name="osztaly_id"
        [(ngModel)]="form.osztaly_id"
    >
        <option value="" disabled>Válassz osztályt</option>
        <option *ngFor="let o of osztalyok" [value]="o.id">
            {{ o.nev }}
        </option>
    </select>
    <label for="osztaly">Osztály</label>
</div>

<!-- Születési dátum -->
<div class="form-floating mb-4">
    <input
        type="date"
        class="form-control"
        id="szuletesiDatum"
        placeholder="Születési dátum"
        name="szuletesi_datum"
        [(ngModel)]="form.szuletesi_datum"
    />
    <label for="szuletesiDatum">Születési dátum</label>
</div>

<!-- Mentés gomb -->
<button type="submit" class="btn btn-primary w-100">
    Mentés
</button>
</form>
</div>
</div>
</div>

```

## Navigációs sáv létrehozása

src/app/app.html fájlban

```
<!-- Felső navigáció -->
<nav class="navbar navbar-expand navbar-dark bg-dark">
  <div class="container-fluid">

    <span class="navbar-brand">
      Iskola - Diákok
    </span>

    <div class="ms-auto">
      <a
        class="btn btn-primary"
        routerLink="/diak-letrehozas"
      >
        + Új diák
      </a>
    </div>

  </div>
</nav>

<!-- Oldaltartalom -->
<div class="container-fluid mt-3">
  <router-outlet></router-outlet>
</div>
```

## RouterLink importja a projektbe, navigációs sáv működtetéshez

és az **src/app/app.ts** állományba a RouterLink-et

```
imports: [RouterOutlet, RouterLink],
```

## Meglévő tanuló szerkesztése funkció létrehozása

*src/app/components/diak-szerkesztes.component/diak-szerkesztes.component.ts*

```
import { Component, OnInit } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';

import { ApiService } from '../../services/api';
import { Osztaly } from '../../models/osztaly';

@Component({
  selector: 'app-diak-szerkesztes',
  standalone: true,
  imports: [CommonModule, FormsModule],
  templateUrl: './diak-szerkesztes.component.html',
  styleUrls: ['./diak-szerkesztes.component.css']
})
export class DiakSzerkesztesComponent implements OnInit {

  diakId = 0;

  osztalyok: Osztaly[] = [];

  // Ugyanaz az űrlapmodell, mint létrehozásnál
  form = {
    vezeteknev: '',
    keresztnév: '',
    email: '',
    osztaly_id: 0,
    szuletesi_datum: ''
  };

  constructor(
    private api: ApiService,
    private route: ActivatedRoute,
    private router: Router
  ) {}

  ngOnInit(): void {
    // ID kiolvasása az URL-ből
    this.diakId = Number(this.route.snapshot.paramMap.get('id'));

    if (!this.diakId) {
      alert('Hiányzó diák azonosító.');
```

```

    this.loadOsztalyok();
    this.loadDiak();
  }

loadOsztalyok(): void {
  this.api.getOsztalyok().subscribe({
    next: (res) => {
      this.osztalyok = res.data;
    },
    error: () => {
      alert('Nem sikerült betölteni az osztályokat.');
```

```

        osztaly_id: Number(this.form.osztaly_id),
        szuletesi_datum: this.form.szuletesi_datum === '' ? null :
this.form.szuletesi_datum
    };

    this.api.updateDiak(payload).subscribe({
      next: () => {
        alert('Sikeres módosítás.');
```

```

        this.router.navigate(['/']);
      },
      error: () => {
        alert('Hiba történt a módosítás során.');
```

```

      }
    });
  }
}
```

<src/app/components/diak-szerkesztes.component/diak-szerkesztes.component.html>

```

<div class="container mt-4">

  <div class="card bg-dark text-light border-secondary">
    <div class="card-body">

      <h2 class="card-title mb-4">Diák szerkesztése</h2>

      <form (ngSubmit)="updateDiak()">

        <!-- Vezetéknév + Keresztnév -->
        <div class="row g-3 mb-3">
          <div class="col-md-6">
            <div class="form-floating">
              <input
                type="text"
                class="form-control"
                id="vezeteknev"
                placeholder="Vezetéknév"
                required
                name="vezeteknev"
                [(ngModel)]="form.vezeteknev"
              />
              <label for="vezeteknev">Vezetéknév</label>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>
```

```

<div class="col-md-6">
  <div class="form-floating">
    <input
      type="text"
      class="form-control"
      id="keresztnev"
      placeholder="Keresztnév"
      required
      name="keresztnev"
      [(ngModel)]="form.keresztnev"
    />
    <label for="keresztnev">Keresztnév</label>
  </div>
</div>

<!-- Email -->
<div class="form-floating mb-3">
  <input
    type="email"
    class="form-control"
    id="email"
    placeholder="Email"
    required
    name="email"
    [(ngModel)]="form.email"
  />
  <label for="email">Email</label>
</div>

<!-- Osztály -->
<div class="form-floating mb-3">
  <select
    class="form-select"
    id="osztaly"
    required
    name="osztaly_id"
    [(ngModel)]="form.osztaly_id"
  >
    <option value="" disabled>Válassz osztályt</option>
    <option *ngFor="let o of osztalyok" [value]="o.id">
      {{ o.nev }}
    </option>
  </select>
  <label for="osztaly">Osztály</label>
</div>

<!-- Születési dátum -->
<div class="form-floating mb-4">

```

```

        <input
            type="date"
            class="form-control"
            id="szuletesiDatum"
            placeholder="Születési dátum"
            name="szuletesi_datum"
            [(ngModel)]="form.szuletesi_datum"
        />
        <label for="szuletesiDatum">Születési dátum</label>
    </div>

    <!-- Mentés gomb -->
    <button type="submit" class="btn btn-primary w-100">
        Mentés
    </button>

</form>

</div>
</div>
</div>

```

## Tanuló lista szerkesztés funkcióval való bővítése

Elhelyezzük az „új” szerkesztés gomb kódot, ezúttal már hivatkozunk a szerkesztés fájlra

```

<button class="btn btn-sm btn-warning me-2" [routerLink]="['/diak-
szerkesztes', d.id]">
    Szerkesztés
</button>

```

## Importáljuk a routingba a diák szerkesztését

```
{ path: 'diak-szerkesztes/:id', component: DiakSzerkesztesComponent }
```

**Gyakori hiba: importáljuk a RouterLink elemet is a lista komponensnél**

**hogyha még nem tettük meg, különben nem működik a Szerkesztés gomb kattintása!**

```
imports: [CommonModule, RouterLink],
```

megjelenik a

```
import { RouterLink } from '@angular/router';
```

sor...



## HIBA komponens elkészítése

(amennyiben rossz elérési útvonal van megadva, a hiba oldalra ugorjon a projekt!)

*src/app/components/hiba/hiba.html*

Elhelyezem az **alert üzenetet**, a **hiba komponens**-en belül, *hiba.html* állományba a megadott üzenettel.

```
<div class="alert alert-danger" role="alert">
  Hiba! A keresett oldal nem található!
</div>
```

Majd végül hivatkozunk rá a routes részen belül, az adott komponensre:

```
// Hiba komponens meghívása, amennyiben rossz útvonalat adnánk meg!
{ path: '**', component: Hiba }
```

## Tesztetem az elkészült projektet!

*ng serve* paranccsal elindítom a szerveret, ha még nem tettem volna meg, és megnézem minden funkció üzemképes-e!