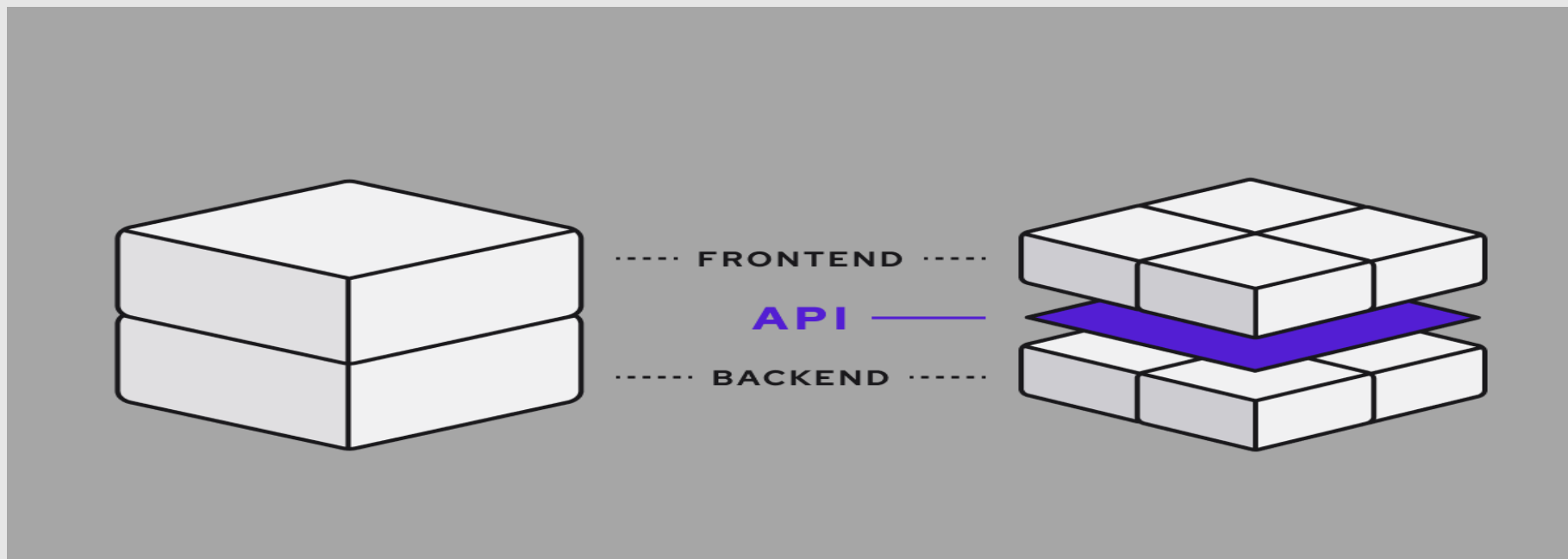


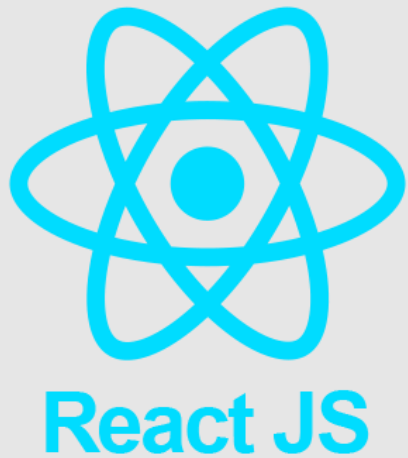


# FULL-STACK APLIKACIJA

# Full-Stack Aplikacija



# Tehnologije u Full-Stack Razvoju



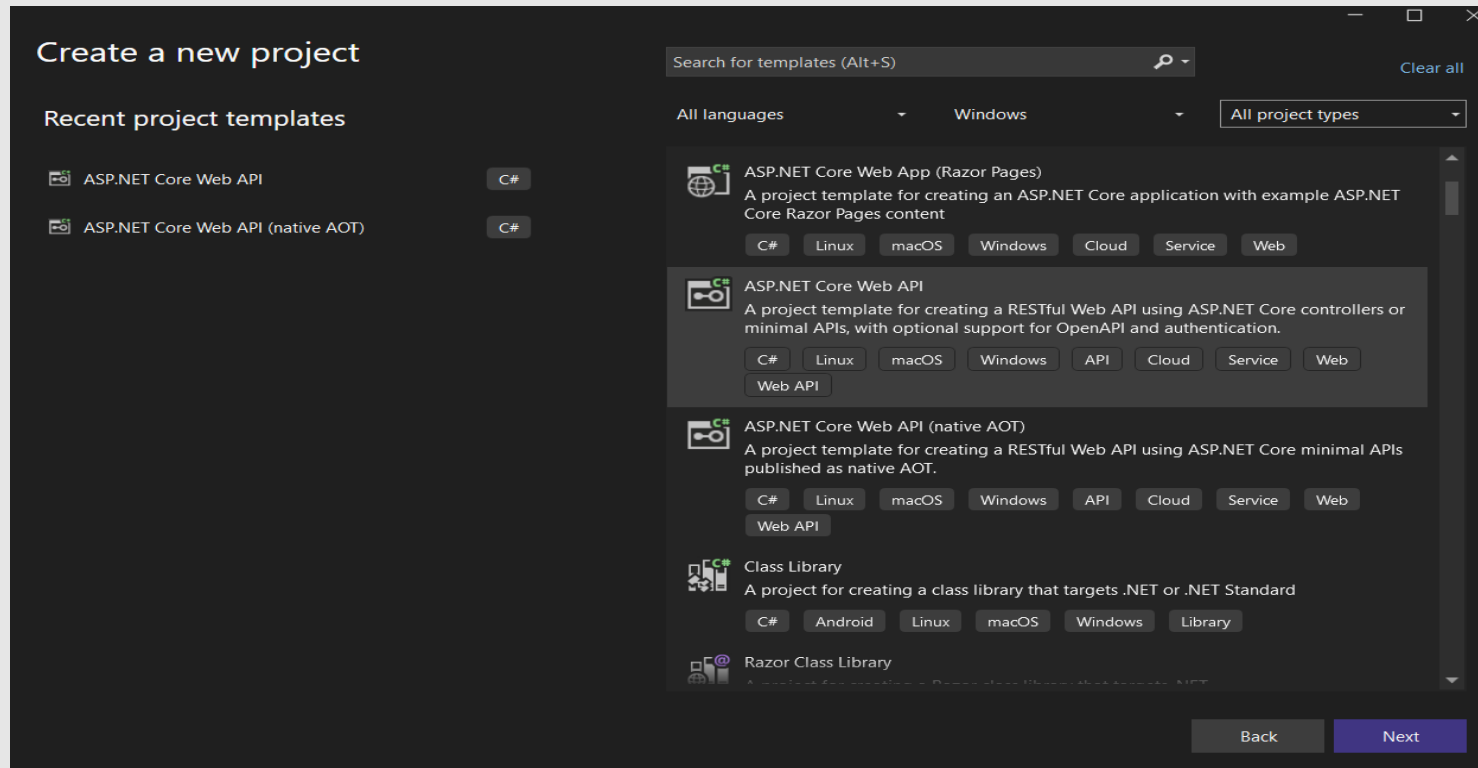
# Struktura kreirane React aplikacije

- ***npx create-react-app naziv aplikacije***-komanda za kreiranje React aplikacije

```
my-app/  
├─ node_modules/  
├─ public/  
│   ├─ index.html  
│   ├─ favicon.ico  
│   └─ manifest.json  
├─ src/  
│   ├─ App.css  
│   ├─ App.js  
│   ├─ App.test.js  
│   ├─ index.css  
│   ├─ index.js  
│   └─ reportWebVitals.js  
├─ .gitignore  
├─ package.json  
├─ README.md  
└─ yarn.lock / package-lock.json
```

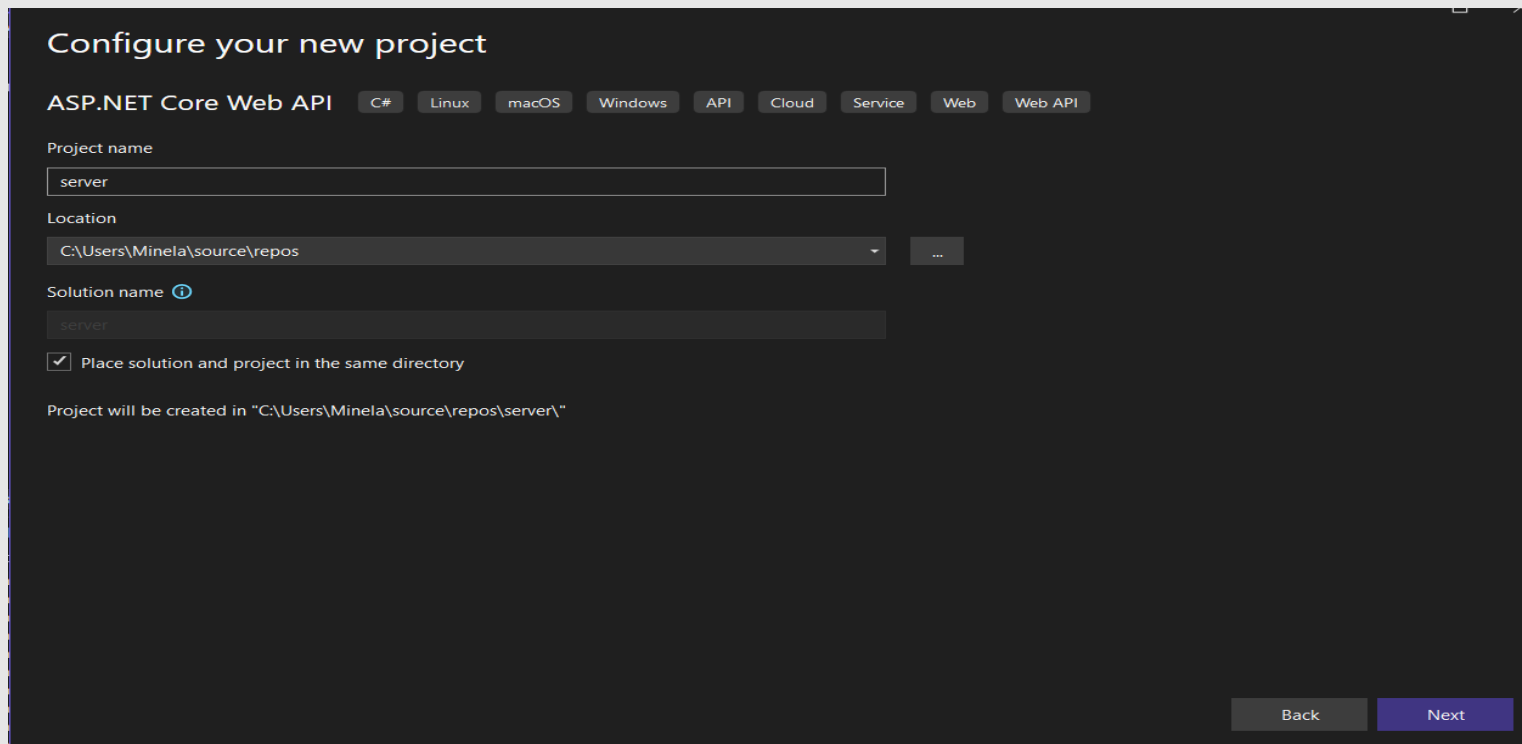
# Kreiranje .NET Framework aplikacije

- Kreiramo aplikaciju u Visual Studio editoru (Korak I)



# Kreiranje .NET Framework aplikacije

- Kreiramo aplikaciju u Visual Studio editoru (Korak II)



The screenshot shows the 'Configure your new project' dialog in Visual Studio. The title is 'Configure your new project'. Below the title, there are several tabs: 'ASP.NET Core Web API' (selected), 'C#', 'Linux', 'macOS', 'Windows', 'API', 'Cloud', 'Service', 'Web', and 'Web API'. The 'Project name' field contains 'server'. The 'Location' field shows the path 'C:\Users\Minela\source\repos' with a dropdown arrow and a '...' button. The 'Solution name' field also contains 'server'. There is a checkbox labeled 'Place solution and project in the same directory' which is checked. At the bottom, it says 'Project will be created in "C:\Users\Minela\source\repos\server\''. At the bottom right, there are 'Back' and 'Next' buttons.

Configure your new project

ASP.NET Core Web API C# Linux macOS Windows API Cloud Service Web Web API

Project name  
server

Location  
C:\Users\Minela\source\repos

Solution name ⓘ  
server

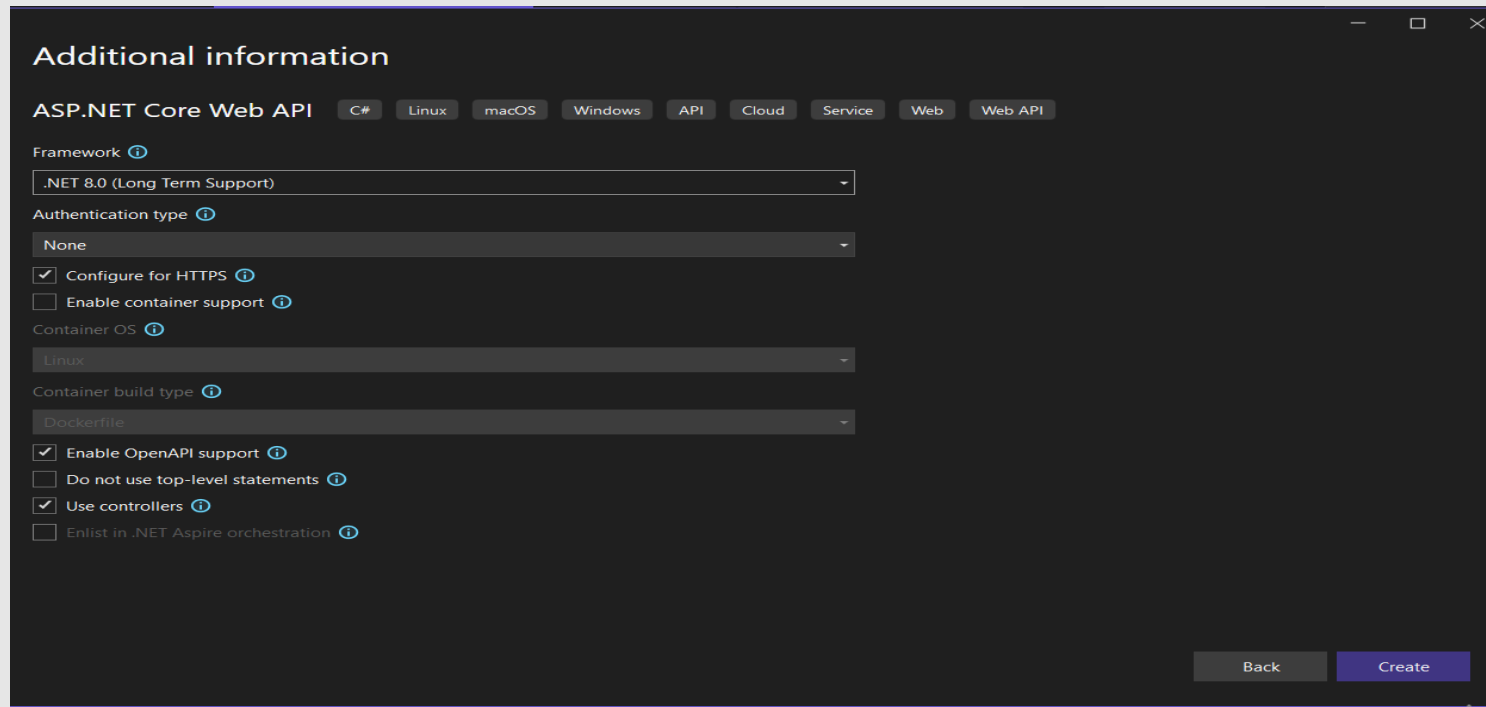
☒ Place solution and project in the same directory

Project will be created in "C:\Users\Minela\source\repos\server\'

Back Next

# Kreiranje .NET Framework aplikacije

- Kreiramo aplikaciju u Visual Studio editoru (Korak III)



The screenshot shows the 'Additional information' dialog box in Visual Studio, which is used to configure the details of a new .NET application. The dialog has a dark theme and a title bar with standard window controls. Below the title, there are tabs for different application types: 'ASP.NET Core Web API', 'C#', 'Linux', 'macOS', 'Windows', 'API', 'Cloud', 'Service', 'Web', and 'Web API'. The 'ASP.NET Core Web API' tab is currently selected. The configuration options include: 'Framework' set to '.NET 8.0 (Long Term Support)', 'Authentication type' set to 'None', 'Container OS' set to 'Linux', and 'Container build type' set to 'Dockerfile'. There are several checkboxes: 'Configure for HTTPS' is checked, 'Enable container support' is unchecked, 'Enable OpenAPI support' is checked, 'Do not use top-level statements' is unchecked, 'Use controllers' is checked, and 'Enlist in .NET Aspire orchestration' is unchecked. At the bottom right, there are 'Back' and 'Create' buttons.

**Additional information**

ASP.NET Core Web API C# Linux macOS Windows API Cloud Service Web Web API

Framework ⓘ  
.NET 8.0 (Long Term Support)

Authentication type ⓘ  
None

☒ Configure for HTTPS ⓘ  
☐ Enable container support ⓘ

Container OS ⓘ  
Linux

Container build type ⓘ  
Dockerfile

☒ Enable OpenAPI support ⓘ  
☐ Do not use top-level statements ⓘ  
☒ Use controllers ⓘ  
☐ Enlist in .NET Aspire orchestration ⓘ

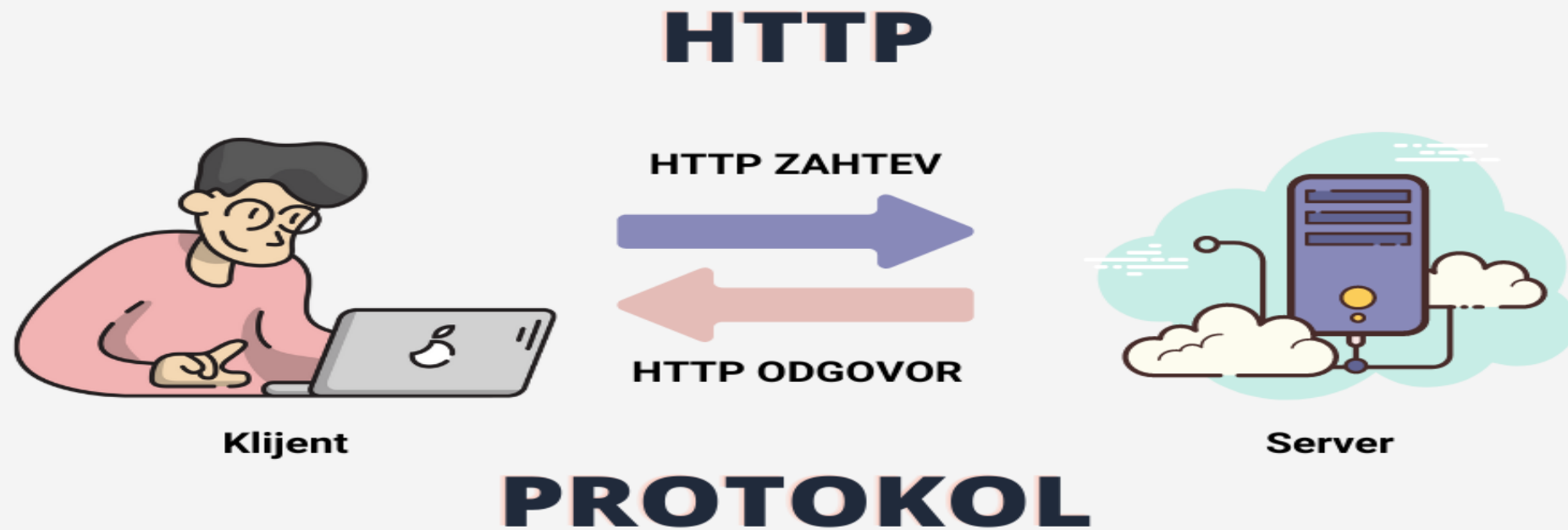
Back Create

# Struktura kreirane Web API aplikacije

```
my-api/  
├─ Controllers/  
|   └─ WeatherForecastController.cs  
├─ Properties/  
|   └─ launchSettings.json  
├─ appsettings.json  
└─ Program.cs
```



# HTTP ZAHTEVI



# HTTP ZAHTEVI

HTTP zahtev sadrži nekoliko ključnih komponenti:

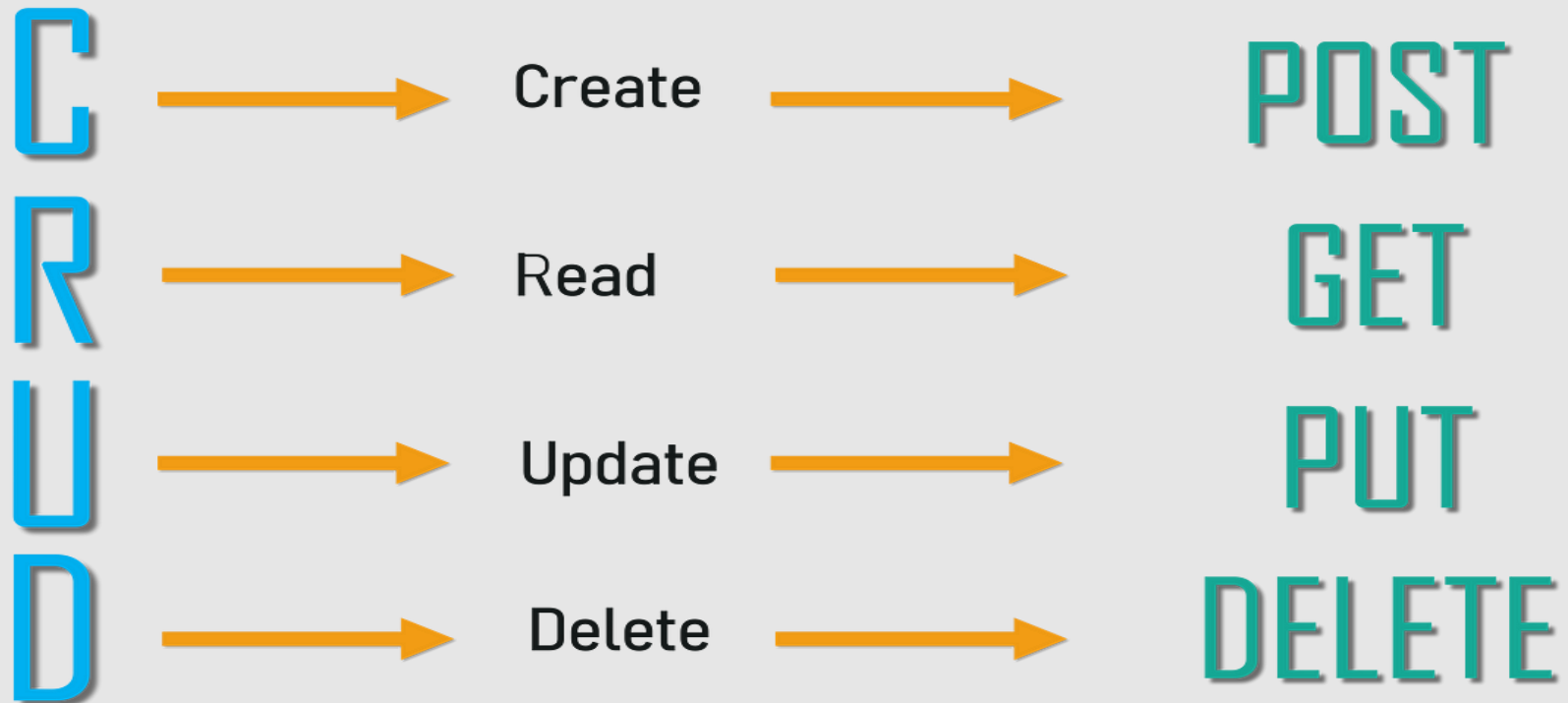
- **HTTP Metoda:** Definiše radnju koju klijent želi da izvrši na serveru.
- **URL (Uniform Resource Locator):** Definiše resurs (npr. fajl, podatak) na serveru koji klijent traži.
- **HTTP Verzija:** Verzija HTTP protokola koja se koristi (npr. HTTP/1.1, HTTP/2).
- **Zaglavlja (Headers):** Sadrže dodatne informacije o zahtevu, kao što su tip sadržaja, autentifikacija, dozvole, i sl.
- **Telo (Body):** Koristi se u nekim HTTP metodama (npr. POST, PUT) za slanje podataka na server.

GET /api/weatherforecast HTTP/1.1

# Objašnjenje toka aplikacije: od frontenda ka backendu i nazad

- **Korisnikova akcija:** Korisnik interaguje sa frontendom (npr. klikne na dugme za prikaz proizvoda).
- **Zahtev sa frontenda ka API-ju:** React aplikacija koristi fetch (ili axios) da bi poslala HTTP zahtev API-ju na backendu.
- **Obrada zahteva na backendu:** .NET API kontroler prima zahtev, obrađuje ga (npr. dobija podatke iz baze podataka), i vraća odgovor (obično u JSON formatu).
- **Odgovor API-ja ka frontendu:** React dobija odgovor (npr. listu proizvoda) i prikazuje te podatke korisniku.

# CRUD STRUKTURA



# Primer CRUD-a

## Weather App

**Weather Forecasts**

Datum prognoze	Temperatura (°C)	Opis	Akcije
2024-10-24	30°C	suncano	<button>Edit</button> <button>Delete</button>
2023-10-24	90°C	suncano	<button>Edit</button> <button>Delete</button>
2023-10-24	20°C	suncano	<button>Edit</button> <button>Delete</button>
2024-10-08	2°C	kisovito	<button>Edit</button> <button>Delete</button>

**Add New Forecast**

Add Forecast

<https://github.com/minelaganovic/WeatherForecastClient>

<https://github.com/minelaganovic/WeatherForecastServer>