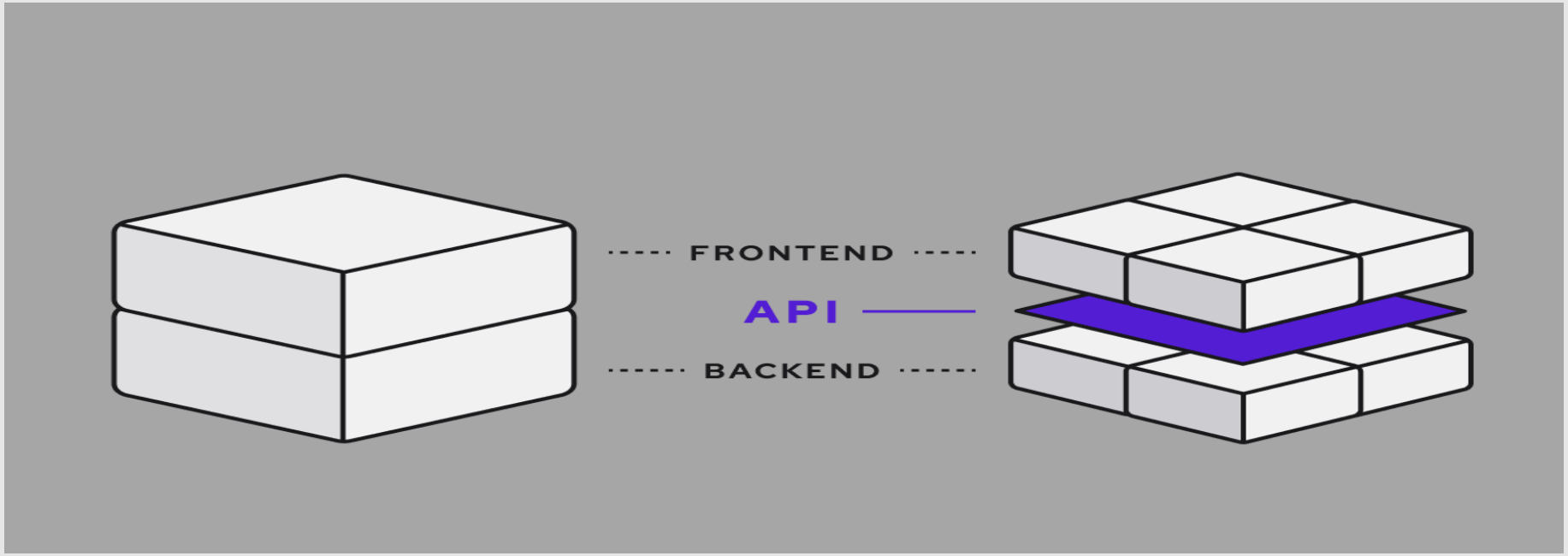
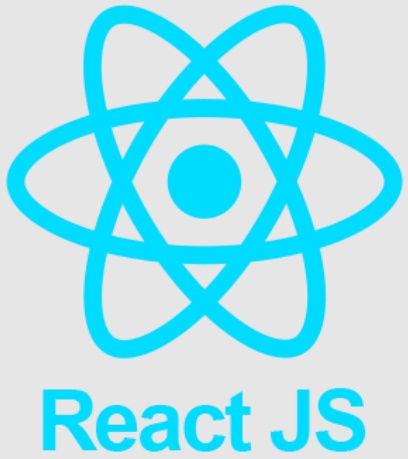
FULL-STACK APLIKACIJA

Full-Stack Aplikacija



Tehnologije u Full-Stack Razvoju



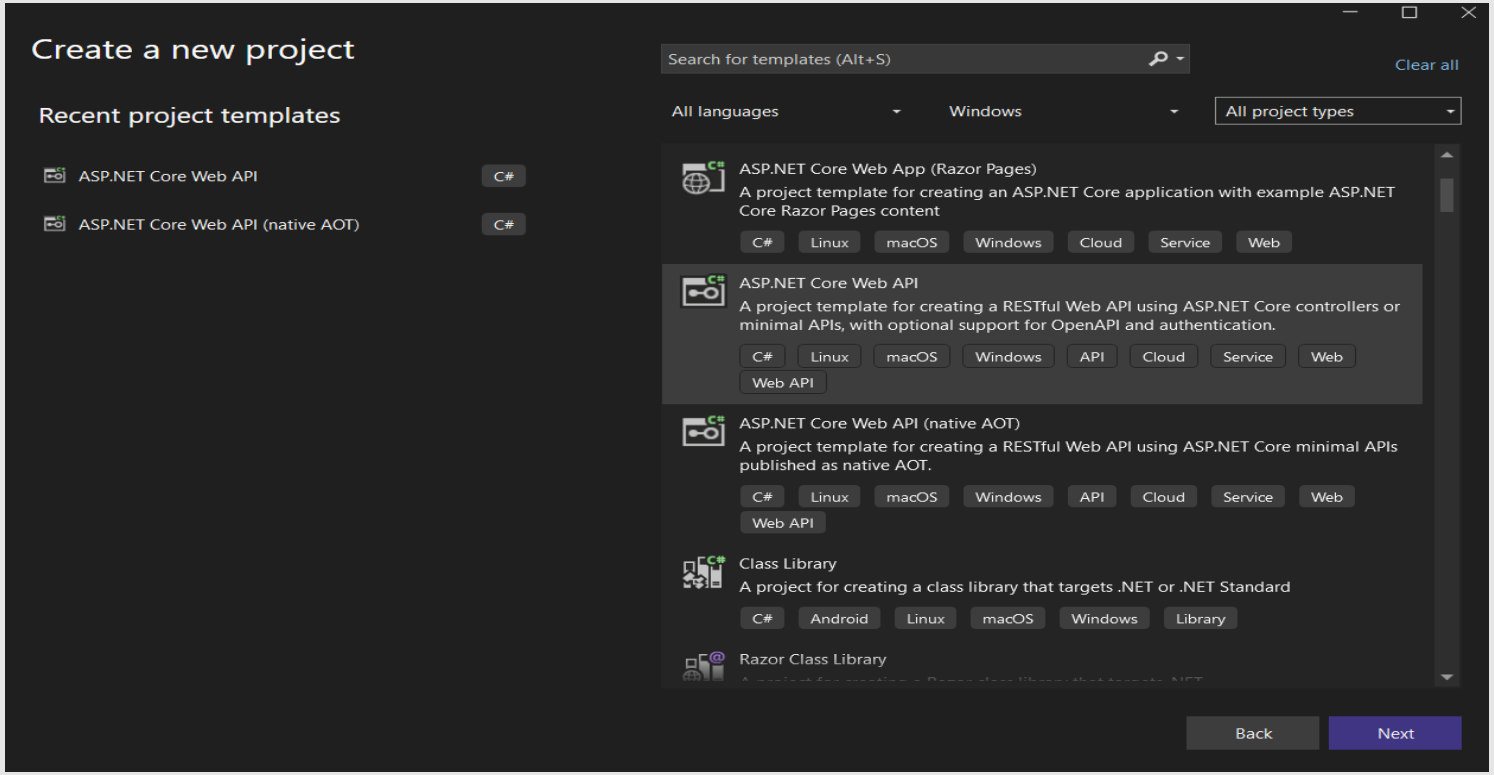
Struktura kreirane React aplikacije

- ***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
```

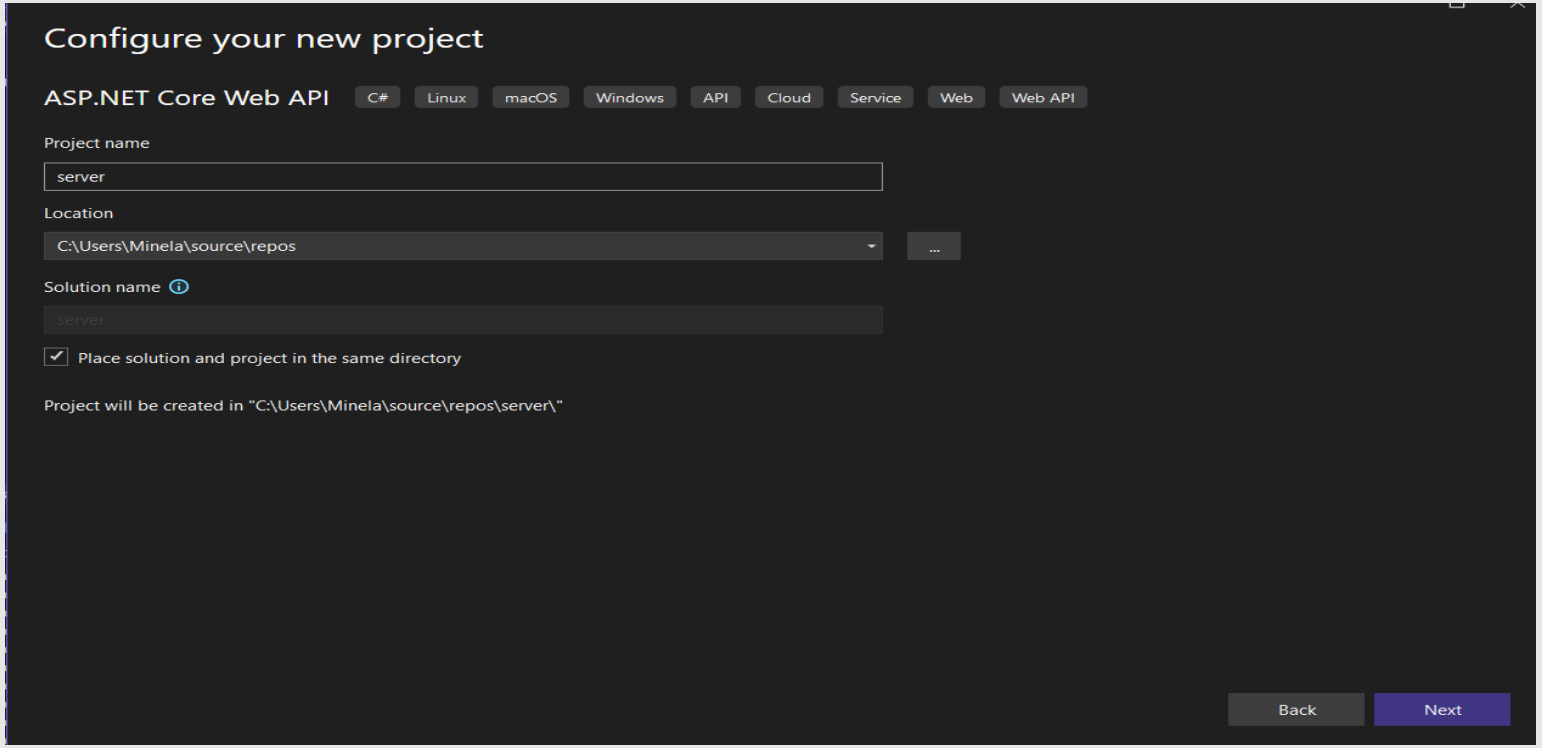
Kreiranje .NET Framework aplikacije

- Kreiramo aplikaciju u Visual Studio editoru (Korak I)



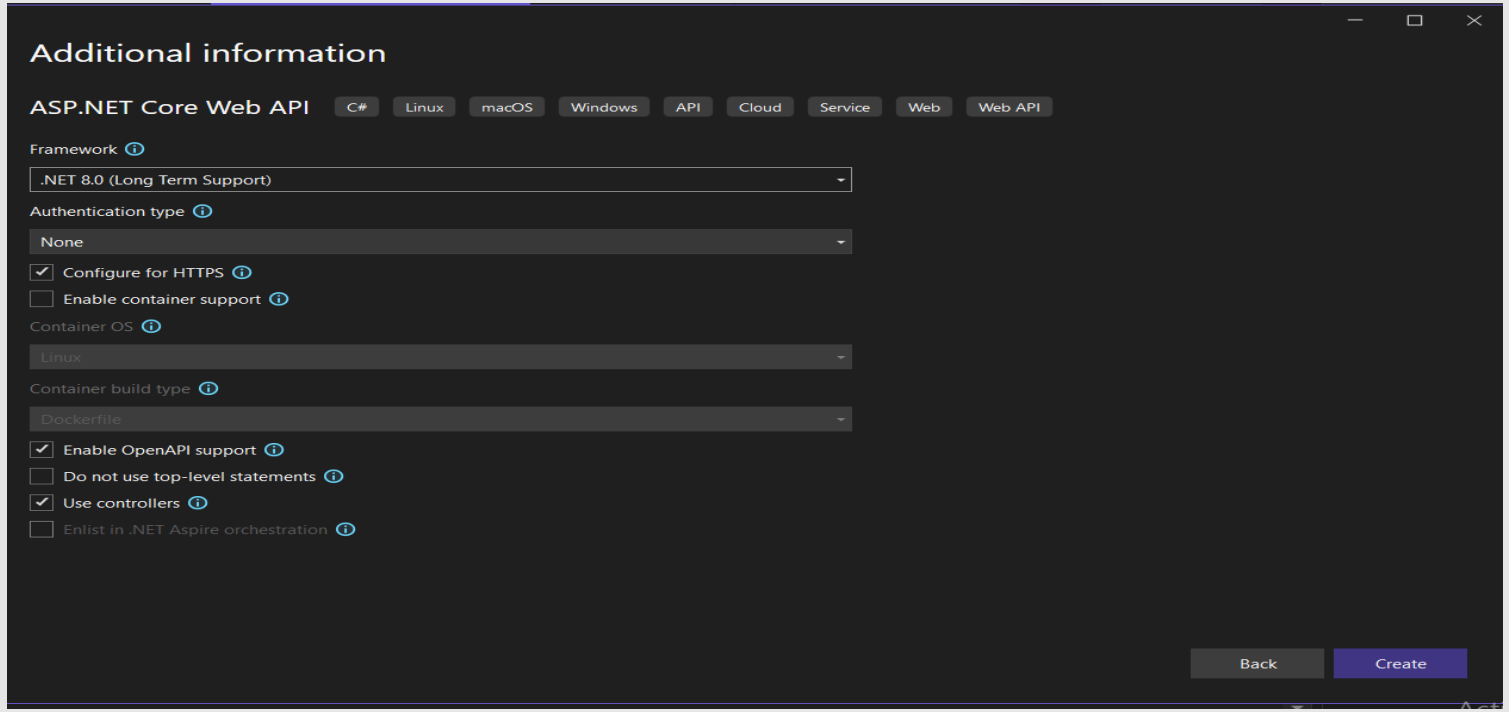
Kreiranje .NET Framework aplikacije

- Kreiramo aplikaciju u Visual Studio editoru (Korak II)



Kreiranje .NET Framework aplikacije

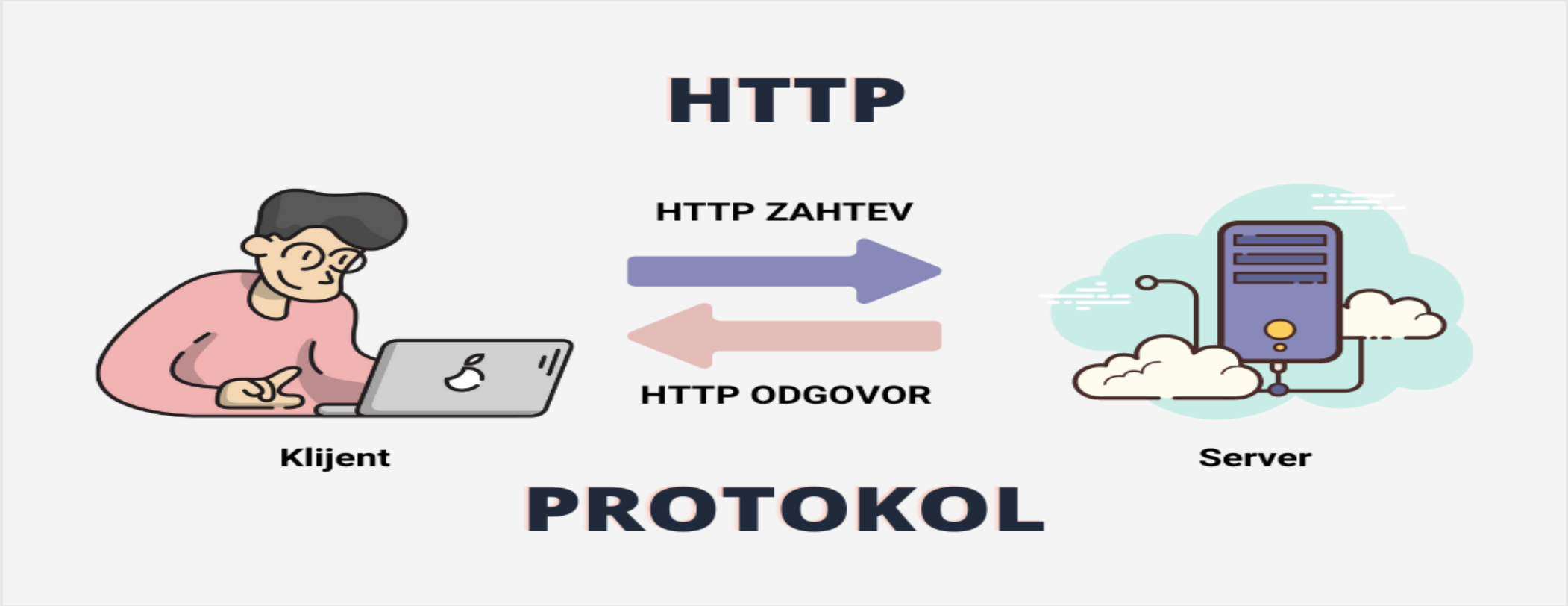
- Kreiramo aplikaciju u Visual Studio editoru (Korak III)



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:

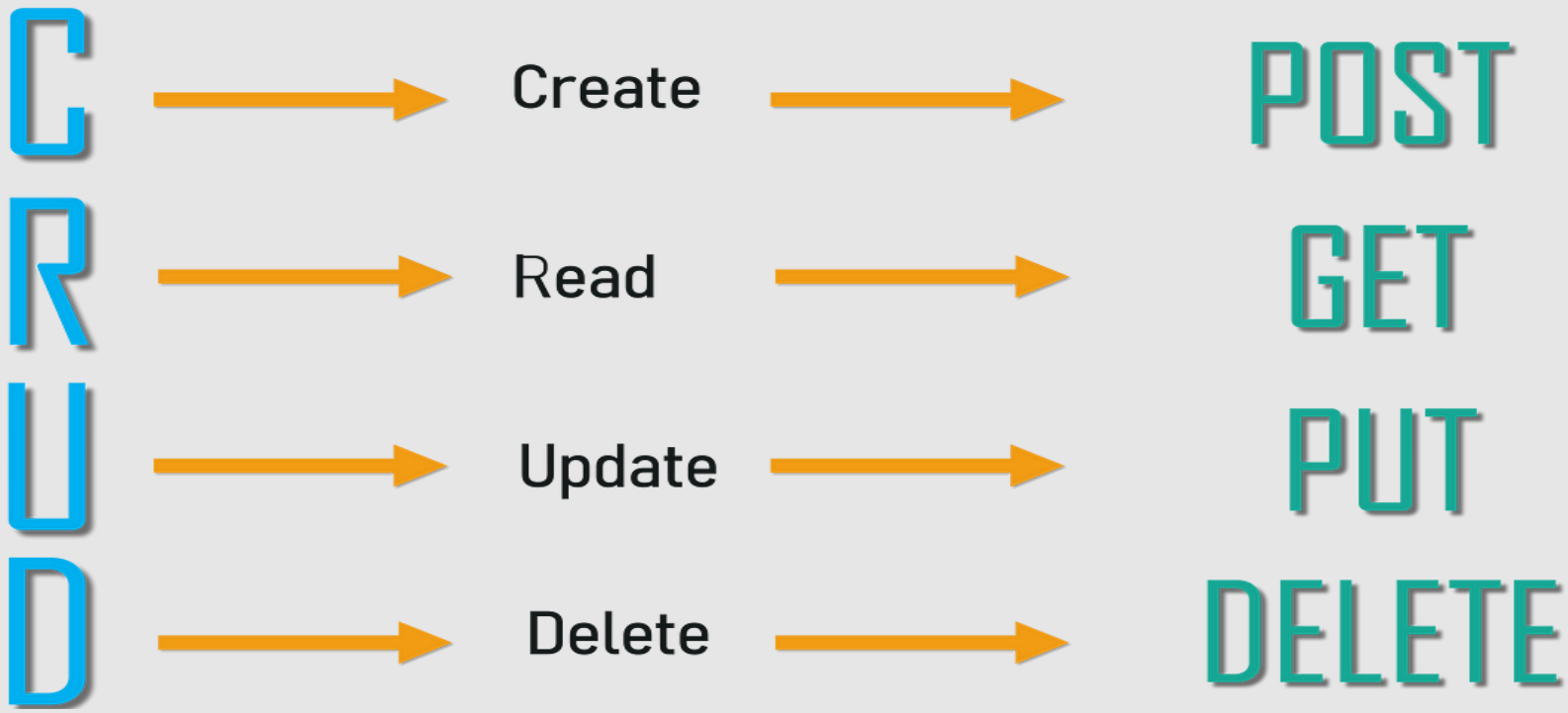
- **HTTP Metoda**: Definiše radnju koju klijent želi da izvrši na serveru.
- **URL (Uniform Resource Locator)**: Definiše resurs (npr. fajl, podatak) na serveru koji klijent traži.
- **HTTP Verzija**: Verzija HTTP protokola koja se koristi (npr. HTTP/1.1, HTTP/2).
- **Zaglavlja (Headers)**: Sadrže dodatne 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

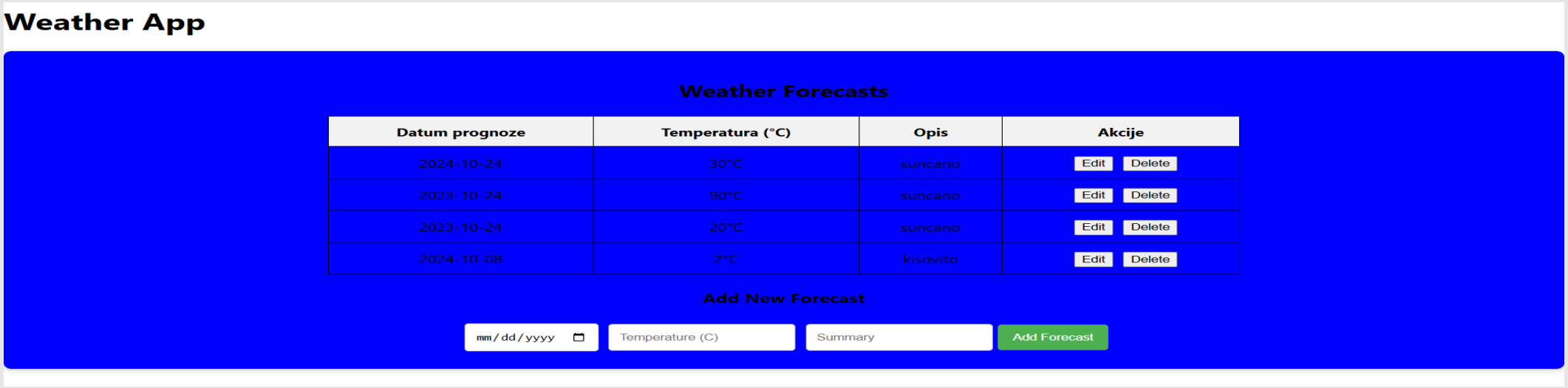
Objašnjenje toka aplikacije: od frontenda ka backendu i nazad

- **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



https://github.com/minelaganovic/WeatherForecastClient

https://github.com/minelaganovic/WeatherForecastServer