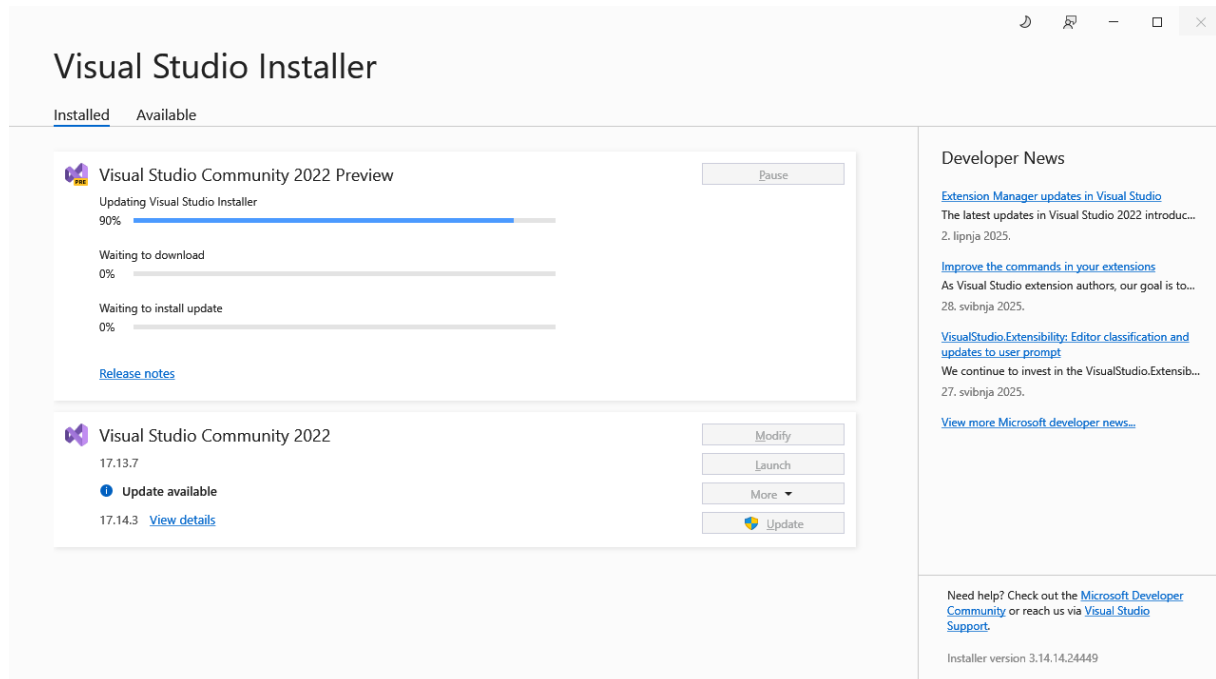


Po potrebi prvo napravite UPDATE MS Visual studio. Pokrenemo program Visual Studio Installer. Ja sam zadnji puta napravio 03. lipnja 2025. i napravio upgrade Visual Studio 2022 Preview na verziju 17.14.3 Preview 1.0 (.NET 9).

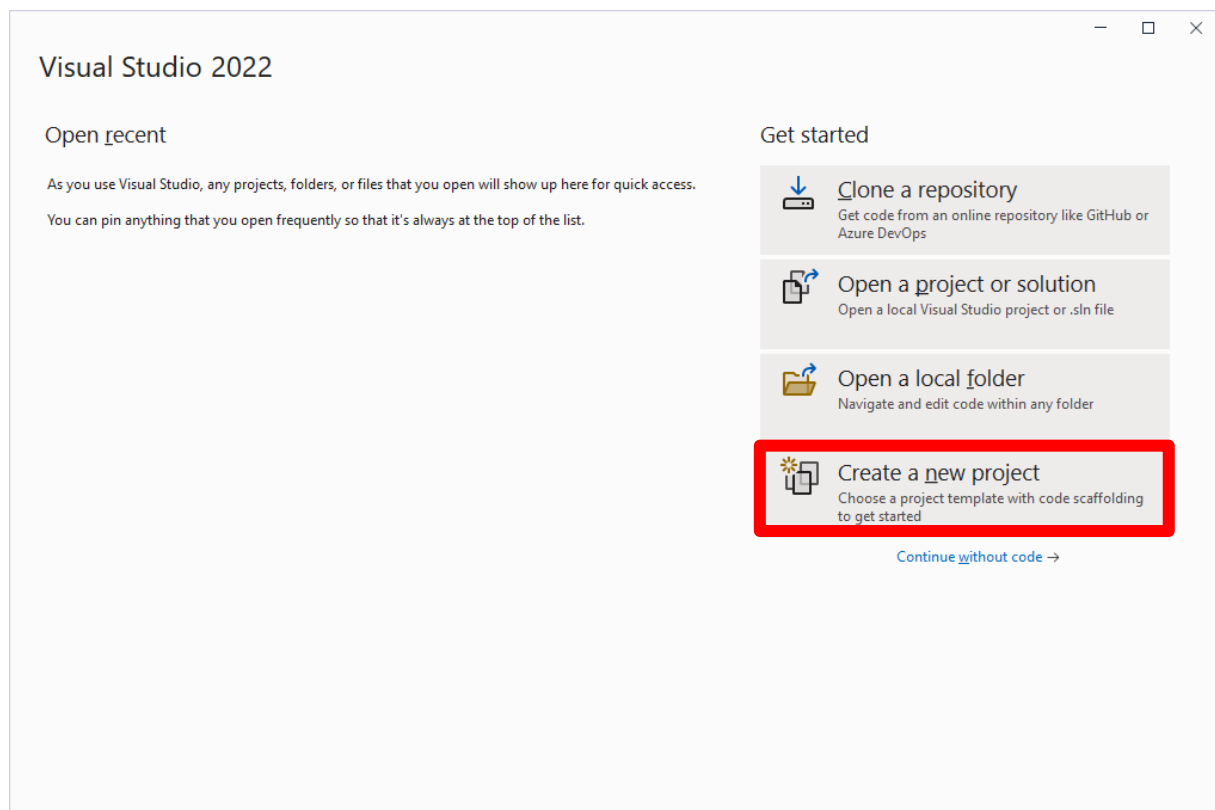


Možda će trebati restart, pa ako traži napravite restart računala!

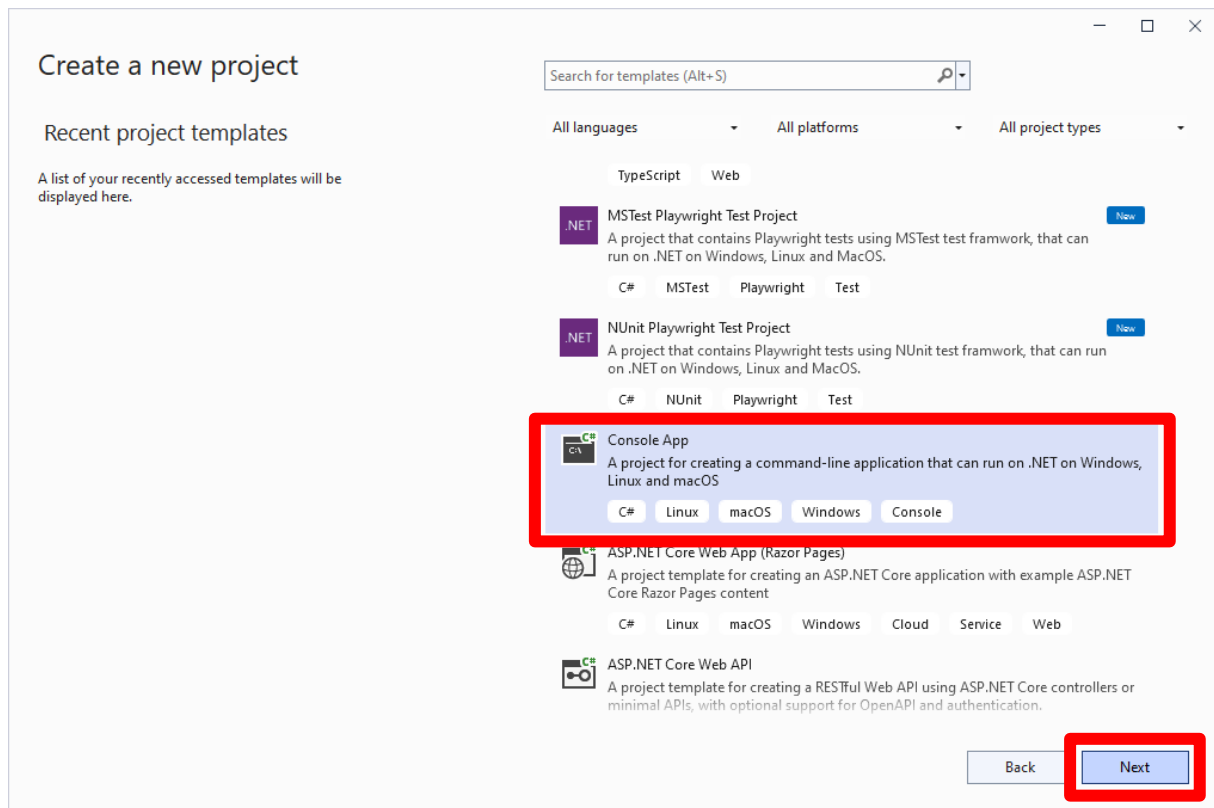
BACKEND

Raditi ćemo paralelno na dva projekta(Project): **Učenje** i **WebAPI** u sklopu rješenja (Solution) .

Prvo odaberemo Create a new project.



Zatim odaberemo Console App i odaberemo Next



Projekt nazovemo **Ucenje**, postavimo ga na lokaciju **BACKEND** (direktorij koji smo kreirali u našem git repozitoriju – pokraj direktorija SQL) te postavimo ime rješenja **BACKEND** (Checkbox Place solution and projekt into sam directory **JE OTKAČEN**) te odaberemo Next.

Configure your new project

Console App C# Linux macOS Windows Console

Project name
Ucenje

Location
C:\Users\Katedra\Documents\GitHub

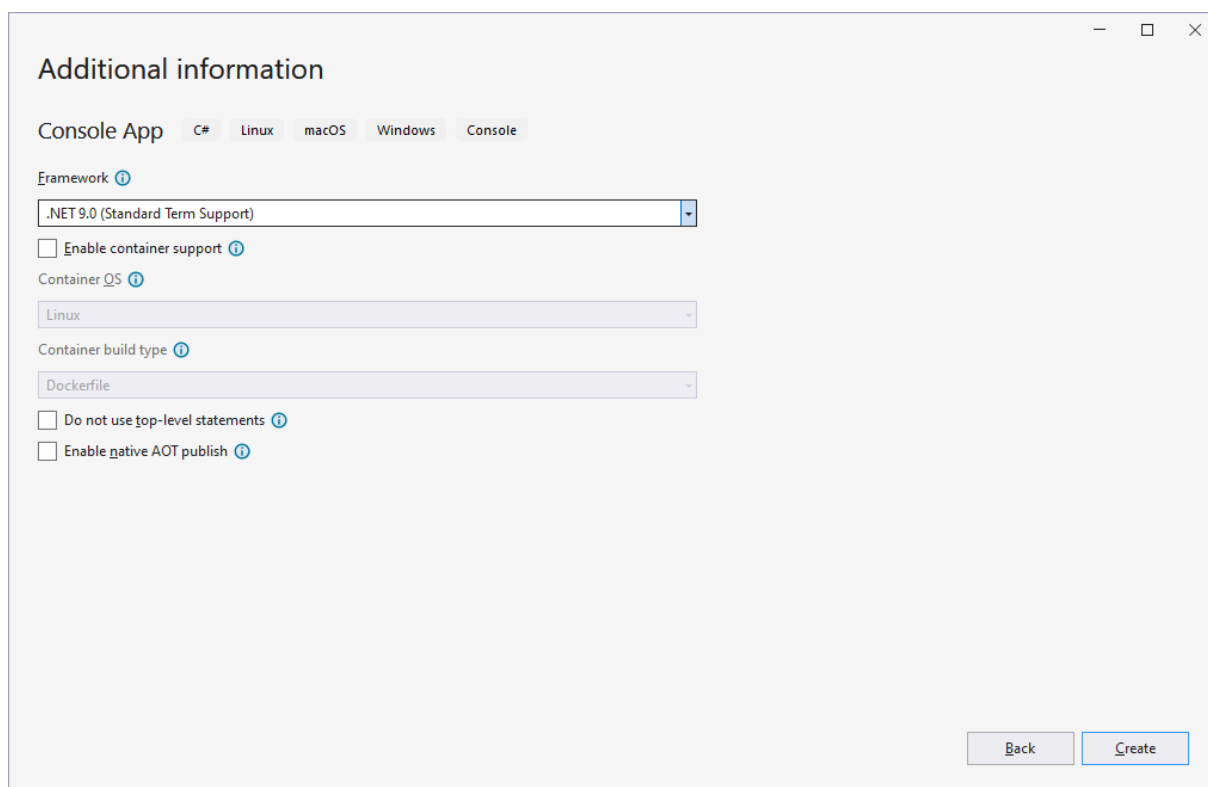
Solution name [?](#)
BACKEND

☐ Place solution and project in the same directory

Project will be created in "C:\Users\Katedra\Documents\GitHub\BACKEND\Ucenje\"

Back Next

Odaberemo radnu okolinu (Framework) 9.0



Additional information

Console App C# Linux macOS Windows Console

Framework ⓘ

.NET 9.0 (Standard Term Support)

☐ Enable container support ⓘ

Container OS ⓘ

Linux

Container build type ⓘ

Dockerfile

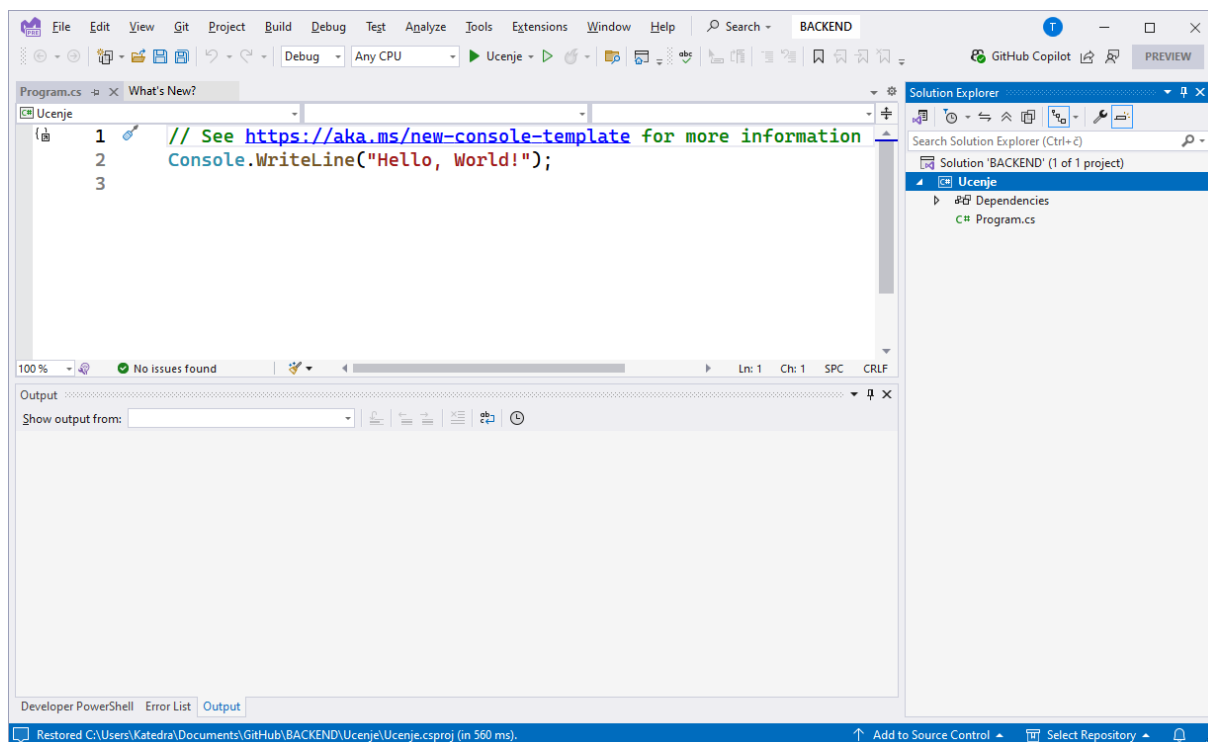
☐ Do not use top-level statements ⓘ

☐ Enable native AOT publish ⓘ

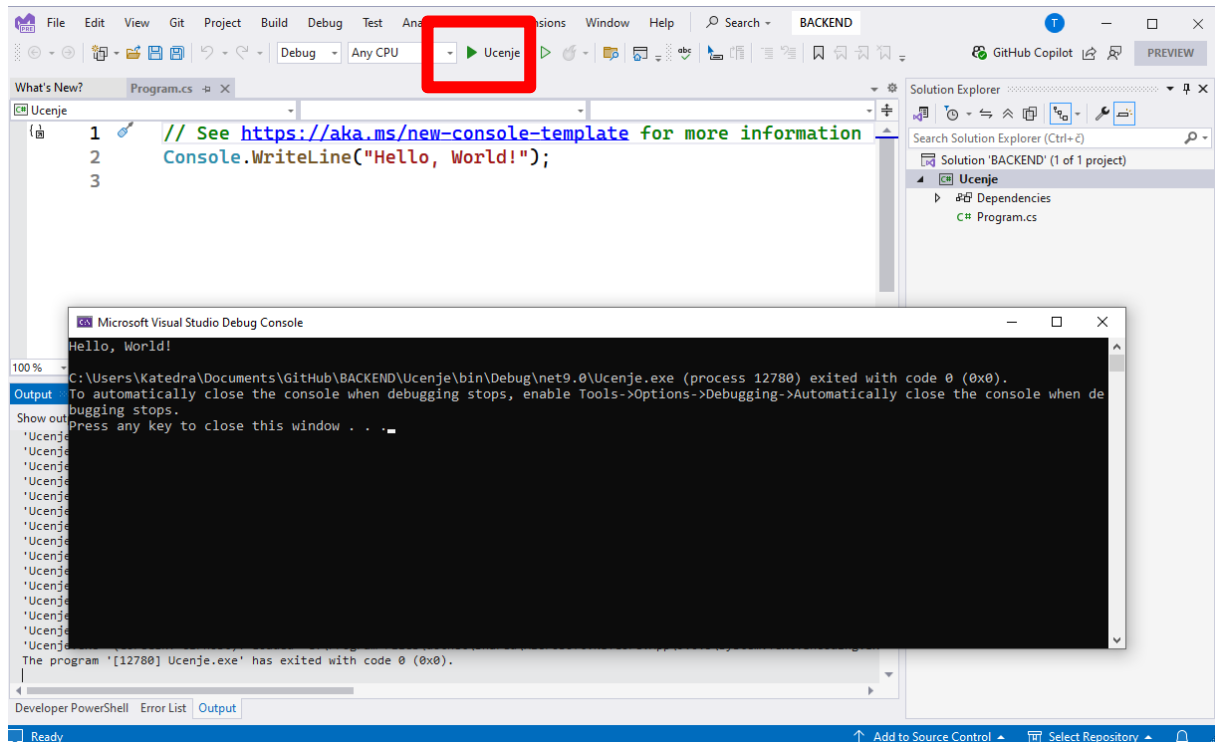
Back Create

Isto tako **NE OZNAČIMO** Checkbox Do not use top-level statements i Enable... te odaberemo Create.

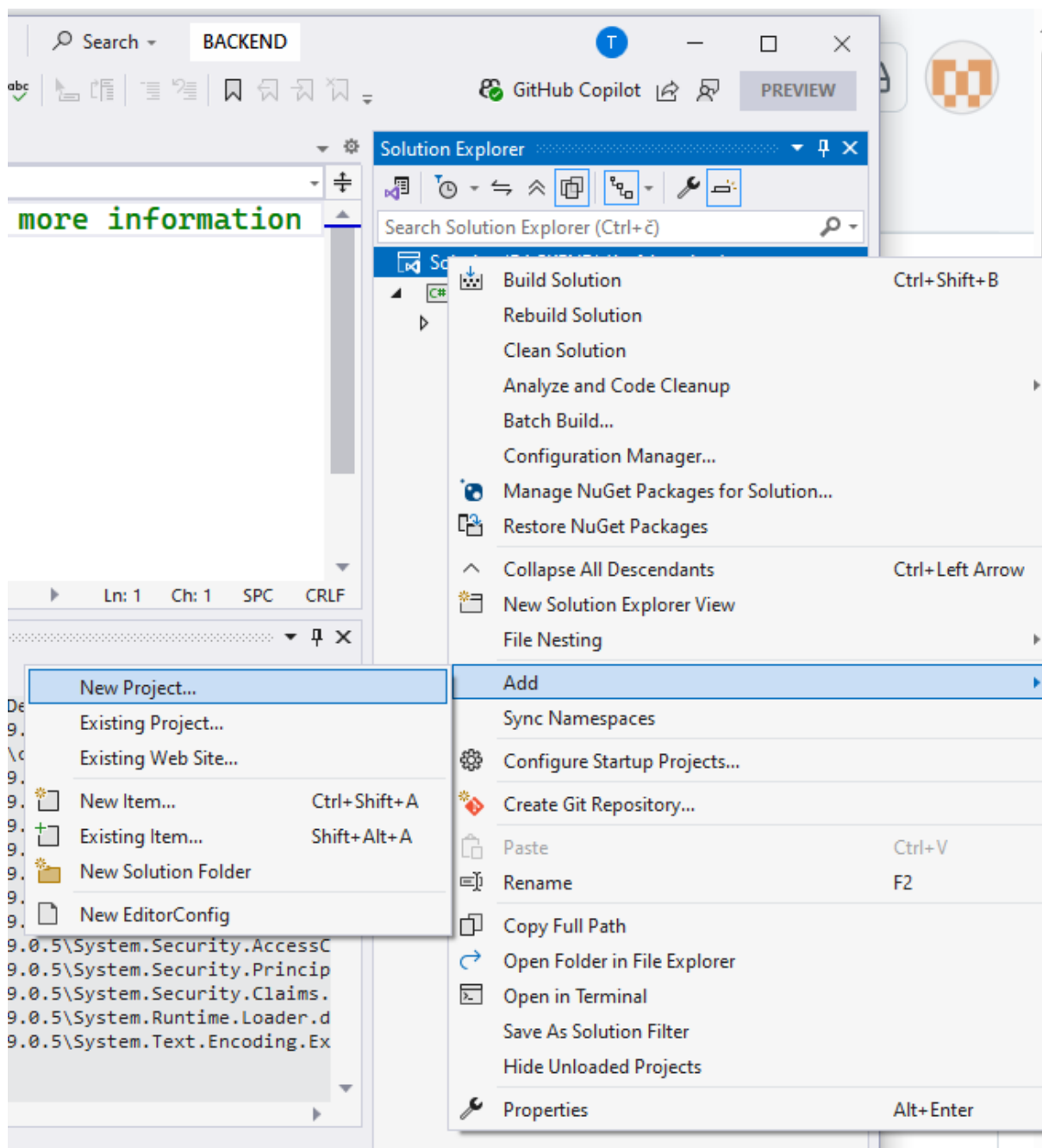
Ovo je naš Konzolni (za učenje) Hello, world program.



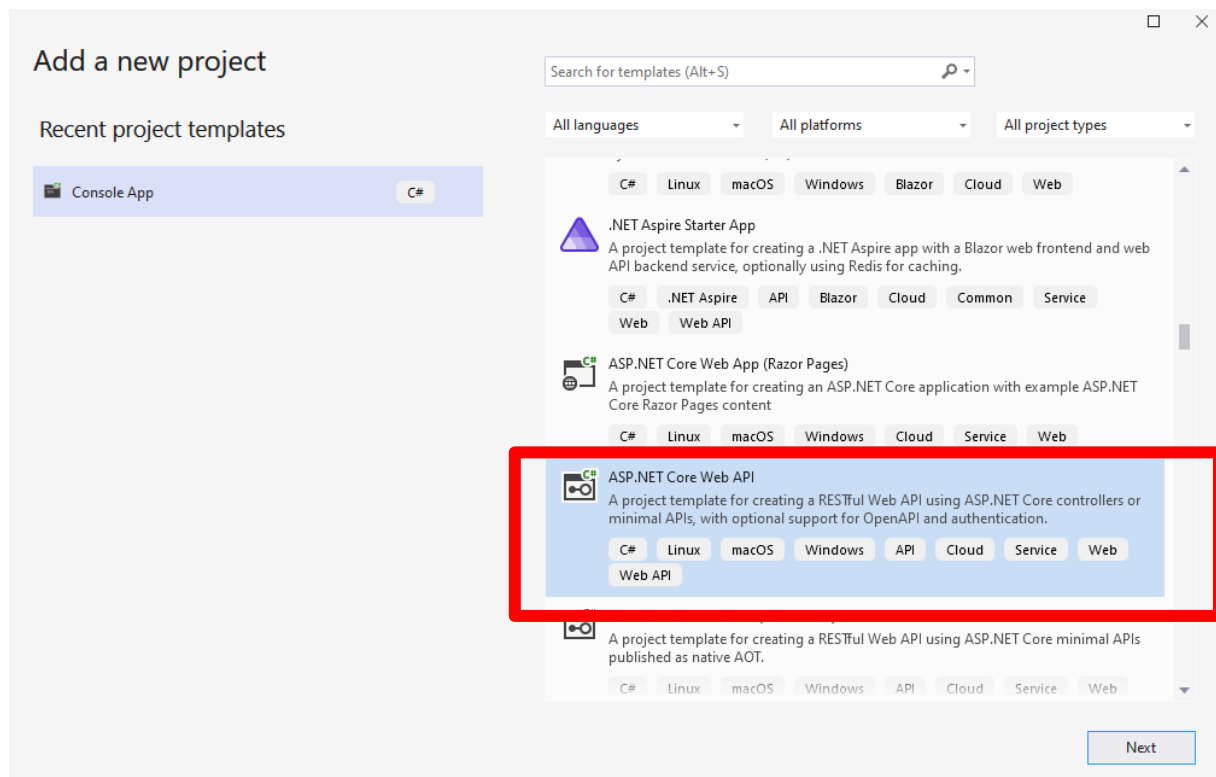
Program pokrećemo s klikom na zelenu PLAY ikonu te rezultat izvođenja programa bude vidljiv u CMD prozoru koji se otvori po uspješnom kreiranju (Build) izvršne datoteke. Pritiskom na bilo koju tipku u tom CMD prozoru, prozor se zatvara.



Desni klik miša na Solution Učenje u dijelu Solution explorer (desni gornji dio zaslona) idemo na Add pa na New Project... kako bi kreirali i ASP .NET Core Web API projekt.



U predloščima pronađemo ASP .NET Core Web API te odaberemo Next



Projekt nazovemo WebAPI te postavimo lokaciju kako stoji jer je to drugi projekt u istom rješenju i odaberemo Next.

□

×

Configure your new project

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

Project name

Location

▼

⌵

Project will be created in "C:\Users\Katedra\Documents\GitHub\BACKEND\WebAPI\."

Back

Next

Opcije ostavio kako je prikazano te odaberemo Create

□

×

Additional information

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

Framework ⓘ

.NET 9.0 (Standard 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 ⓘ

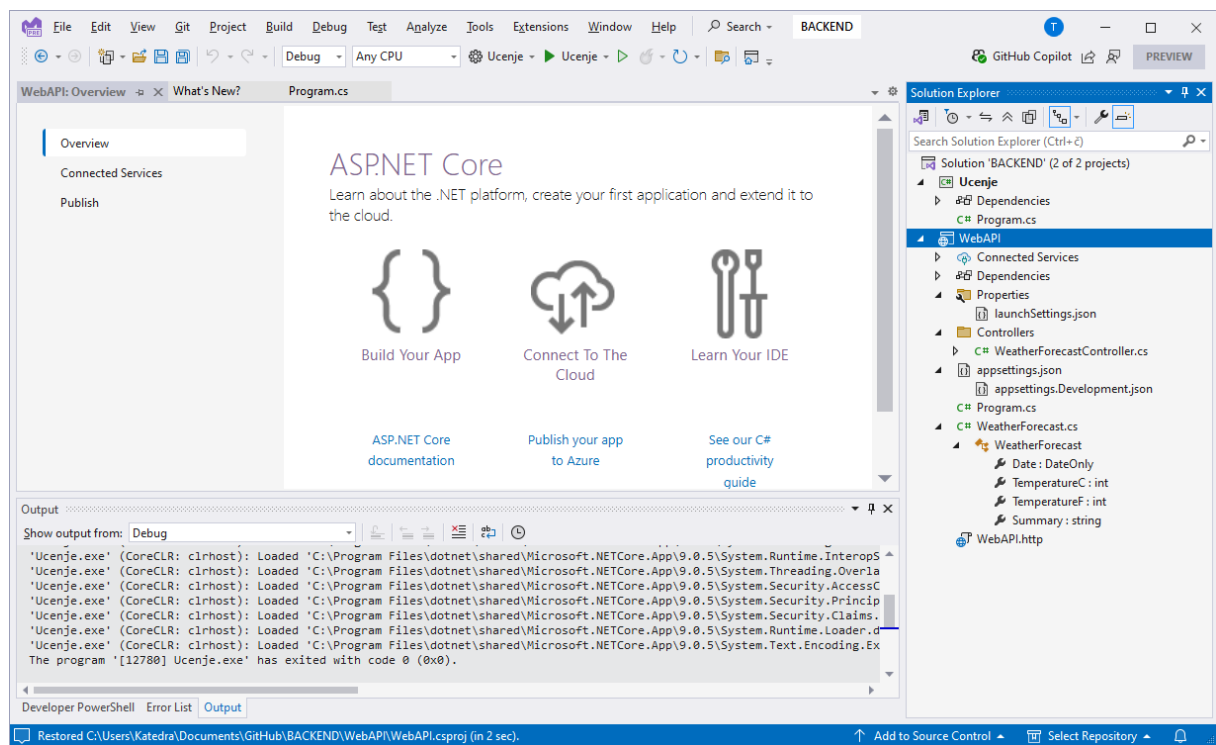
☒ Use controllers ⓘ

☐ Enlist in .NET Aspire orchestration ⓘ

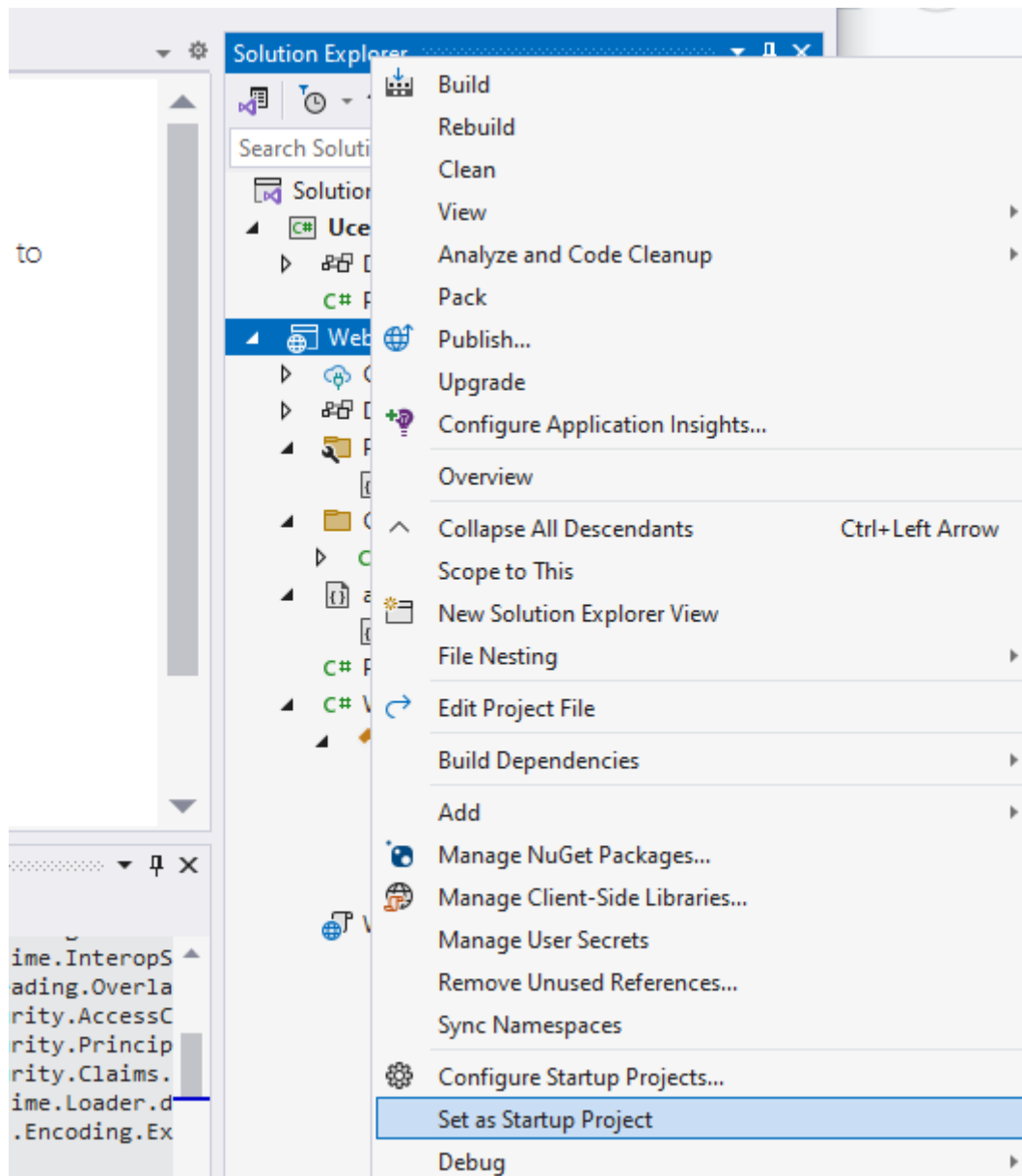
Back

Create

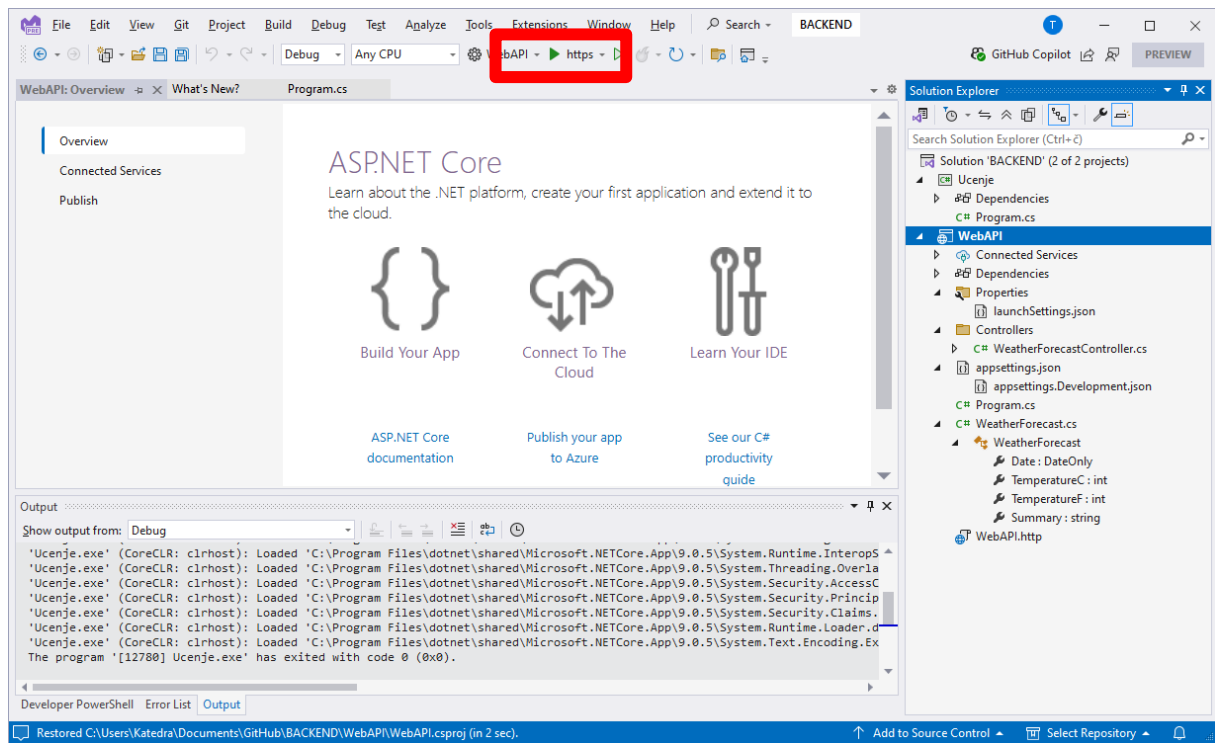
Ovaj drugi projekt je po strukturi puno kompleksniji ali to nas za sada ne zanima – ta kompleksnost je predmet učenja.



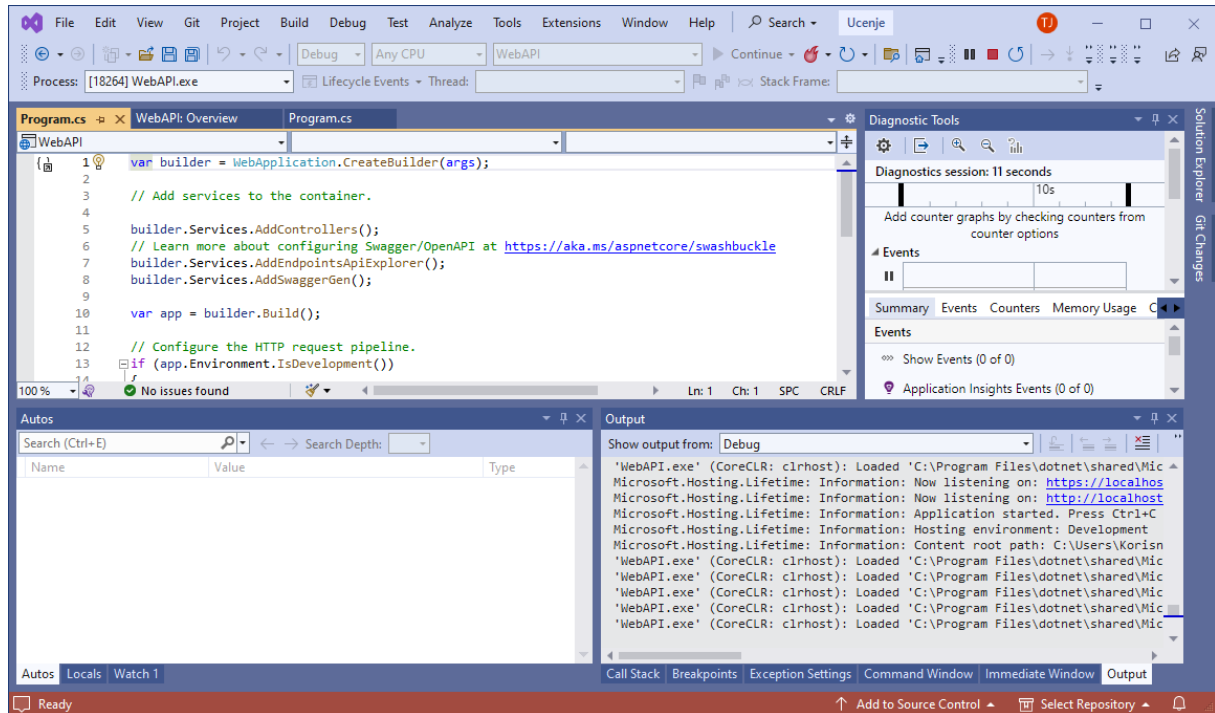
Kako sada u rješenju imamo dva projekta pokretanje je ostalo na projektu UcenjeCS. Da bi omogućili pokretanje WebAPI projekta moramo ga postaviti kao Start Up projet. Stoga radimo desni klik na naziv projekta i odaberemo Set as Startup Project



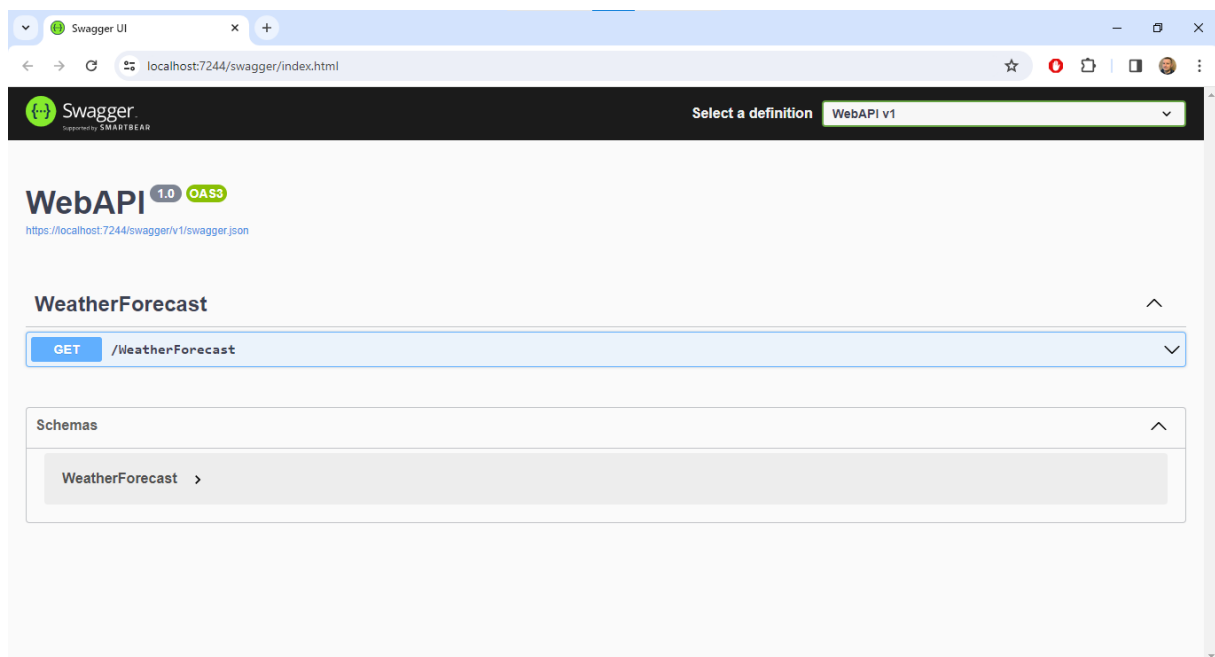
Sada možemo ponovo na zelenu PLAY ikonu pokrenuti projekt WebAPI.



Sada dobivamo drugačije pokretanje u odnosu Console aplikaciju (UcenjeCS) jer će se sada prilikom pokretanja podići i web server.

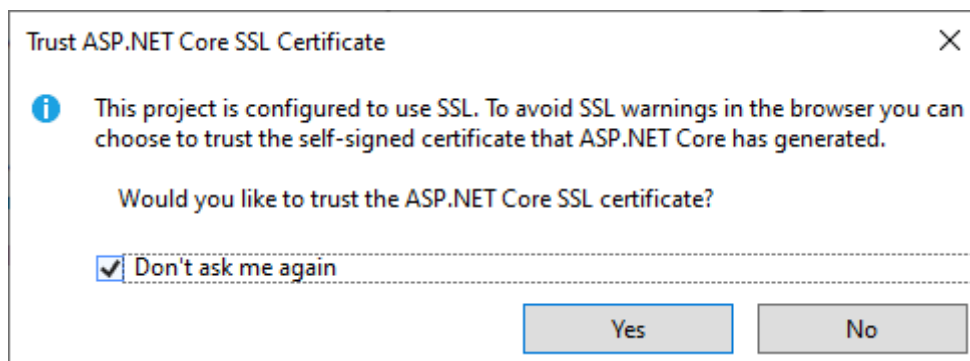


Te će se pokrenuti preglednik (Chrome ili Edge) s prikazom Swagger sučelja.

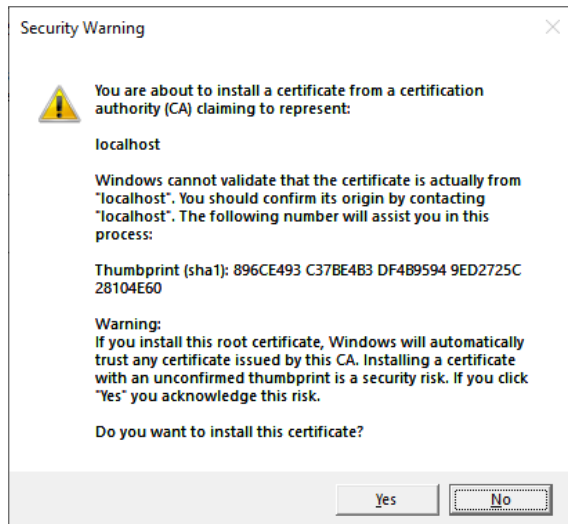


Na git-u će biti dodaci koje je potrebno odraditi da bi se javio swagger sučelje

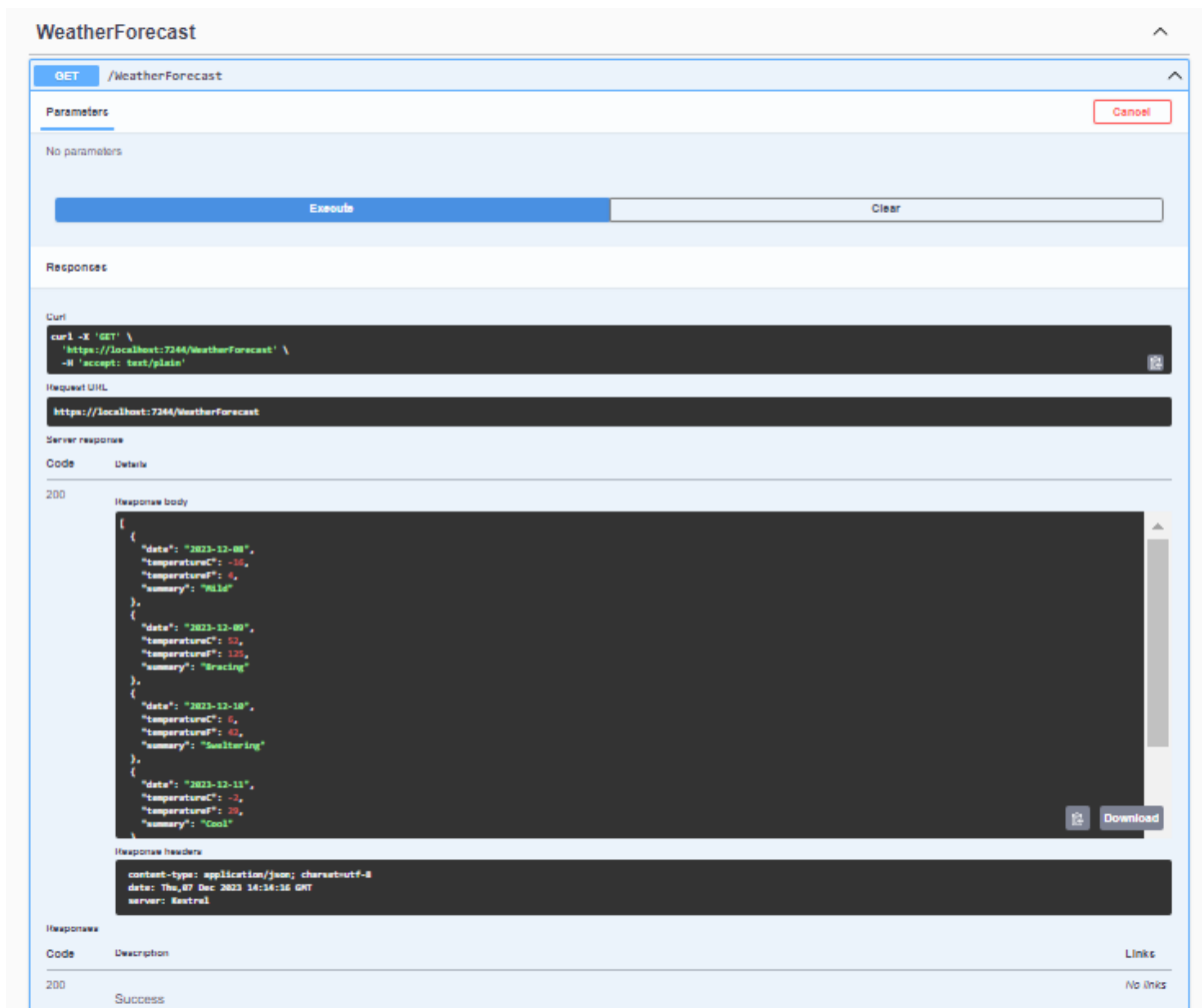
Ukoliko nas pita želimo li vjerovati SSL certifikatu, pritisnemo Yes



I Yes na instalaciji certifikata



Swagger je alat koji nam omogućuje vizualni pregled i korištenje dostupnih ruta. Kliknemo na GET /EatherForecast te klik na opciju Try it out (gore desno) pa Execute i dobijemo rezultat izvođenja te rute.



The screenshot displays the Swagger UI for the WeatherForecast API. The top bar shows the endpoint GET /WeatherForecast. The Parameters section is empty, indicating no query or body parameters. The Execute button is prominent. The Responses section shows a 200 status code with a JSON response body and headers.

Parameters

No parameters

Execute **Clear**

Responses

Code **Details**

200

Response body

```
{
  "data": "2023-12-08",
  "temperature": -10,
  "temperaturef": 0,
  "summary": "Wild"
},
{
  "data": "2023-12-09",
  "temperature": 10,
  "temperaturef": 32,
  "summary": "Bracing"
},
{
  "data": "2023-12-10",
  "temperature": 0,
  "temperaturef": 32,
  "summary": "Smoldering"
},
{
  "data": "2023-12-11",
  "temperature": -2,
  "temperaturef": 28,
  "summary": "Cool"
}
```

Response headers

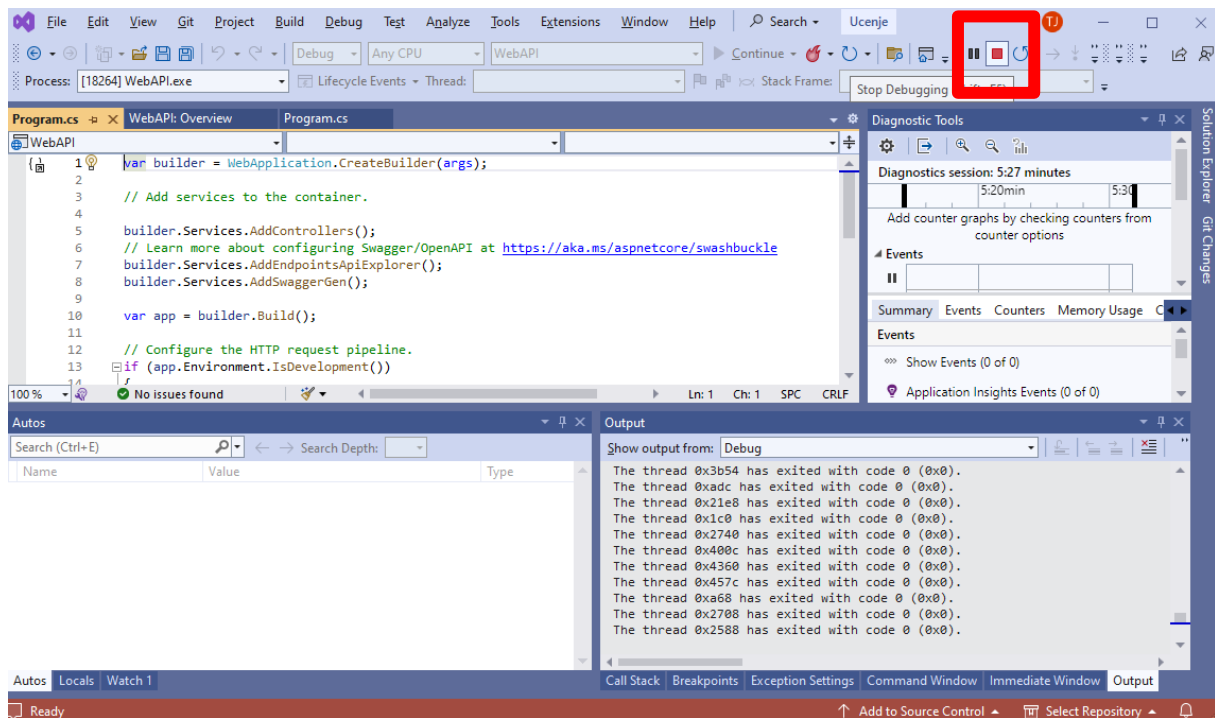
```
content-type: application/json; charset=utf-8
date: Thu, 07 Dec 2023 14:14:16 GMT
server: Kestrel
```

Response

Code	Description	Links
200	Success	No links

Ovo su nam polazišne točke.

WebAPI projekt se zaustavlja pomoću crvene stop ikone.



Happy coding!