

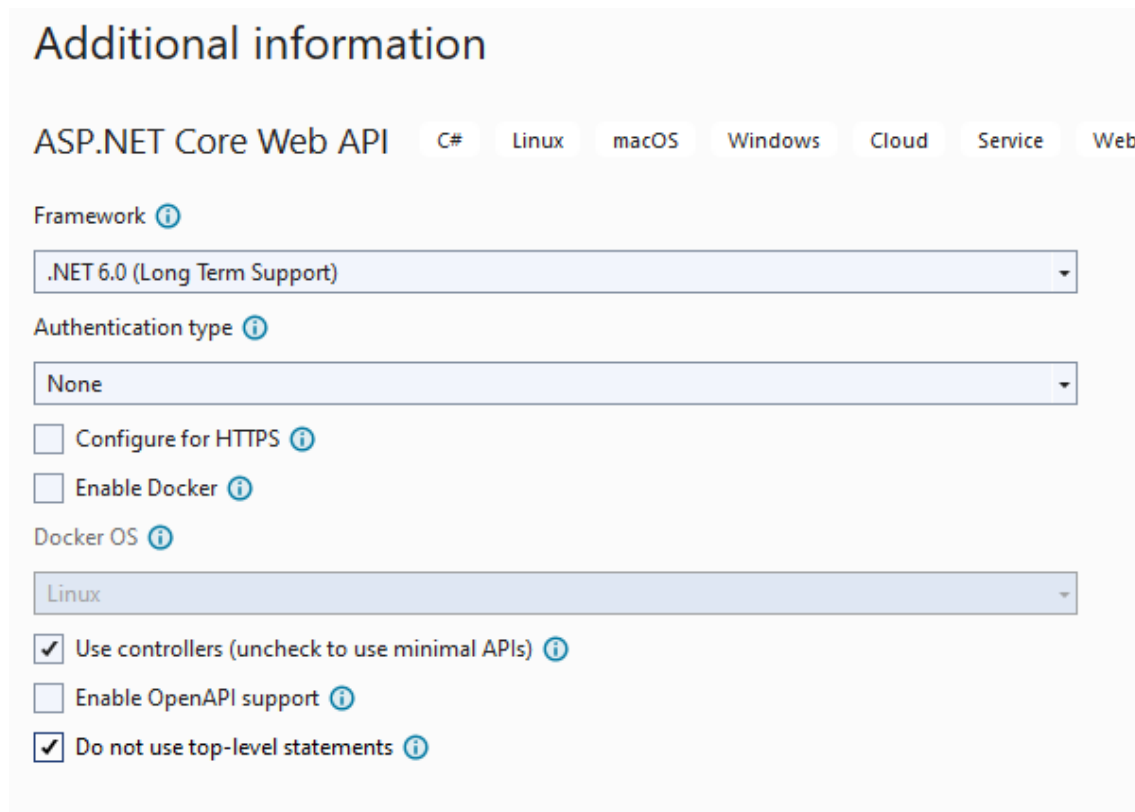
Web API készítése Entity Framework segítségével

Adatbázis feltételek

MySQL adatbázisban léteznie kell egy táblának. A MySQL szervernek futnia kell.

Web API projekt elkészítése

A projektben az **Additional information** oldalon az alábbi pipáknak kell szerepelniük.



Additional information

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

Framework ⓘ
[.NET 6.0 (Long Term Support)]

Authentication type ⓘ
[None]

☐ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

Docker OS ⓘ
[Linux]

☒ Use controllers (uncheck to use minimal APIs) ⓘ

☐ Enable OpenAPI support ⓘ

☒ Do not use top-level statements ⓘ

NuGet csomagok telepítése

A projekthez hozzá kell adni az alábbi **NuGet** csomagokat. **Ajánlott verziók zárójelben**

- Microsoft.EntityFrameworkCore (verzió: 6.0.23)
- Microsoft.EntityFrameworkCore.SqlServer (6.0.23, Controller varázslóhoz szükséges)
- Microsoft.EntityFrameworkCore.Tools (6.0.23)
- Microsoft.VisualStudio.Web.CodeGeneration.Design (6.0.16, Controller varázslóhoz)
- Pomelo.EntityFrameworkCore.MySql (6.0.2)

Scaffold-DbContext utasítás

A **Package Manager Console**-on ki kell adni a következő utasítást, de még előtte:

- a helyes adatbázis nevébe kell illeszteni a connectionString-be
- **Fontos!** az alábbi utasítás csak 1 sorból álljon, a sortörést ki kell szedni

```
Scaffold-DbContext "server=localhost;user id=root;database= "
Pomelo.EntityFrameworkCore.MySql -OutputDir Models -DataAnnotations
-UseDatabaseNames -Force -NoPluralize
```

Projekt felépítése

A leképezett osztályok után a **Build** (Ctrl+Shift+B) parancsot kell kiadni a projekten.

Startup service hozzáadása

A Program.cs fájlban hozzá kell adni egy DbContext szolgáltatást az elkészült context osztályunk nevével.

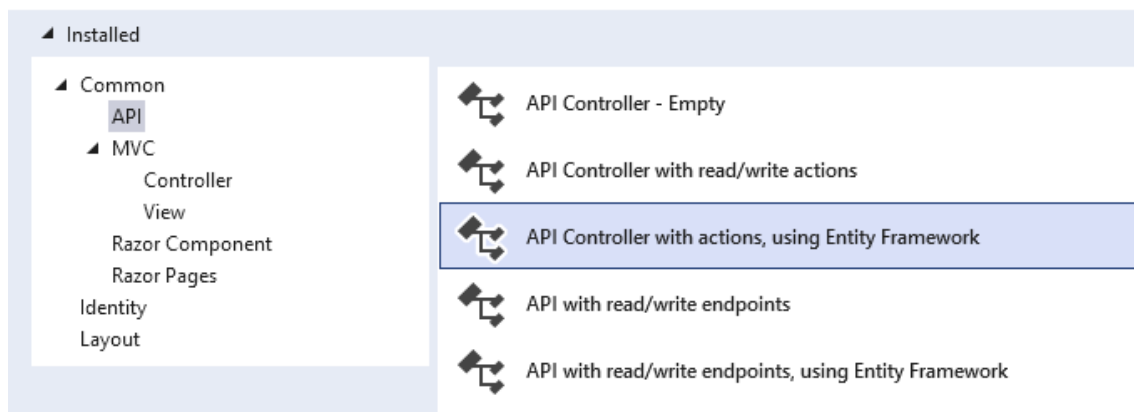
```
builder.Services.AddDbContext<ContextOsztály>();
```

Fontos, hogy a builder.Build() sor előtt legyen.

Controller készítése

Jobb klikk a Controllers mappán, majd Add ➤ Controller. Ki kell választani az API Controllert, Entity Framework műveletekkel.

Add New Scaffolded Item



A következő ablakban meg kell adni az MVC kapcsolatot:

- **Model class:** adatbázis tábla model osztálya
- **Data context class:** adatbázis kapcsolat osztálya
- **Controller name:** Ez a cím fog szerepelni az URL-ben. Fontos, hogy „Controller”-re végződjön az osztály neve.

launchsettings.json átírása

A Properties mappában lévő *launchsettings.json* fájlban át kell írni a `launchUrl` változót a saját controllerünk útvonalára (*api/ControllerNév*).

Tesztelése böngészőben

Futtatásnál válasszuk az **IIS Express helyett a Projektünk nevével megegyező profilt**.

Indítás után megjelenik a böngészőablak az adatbázisból lekérdezett JSON formátumú adatokkal.

Ki lehet kapcsolni az automatikus böngészőablak indítást: a projekt nevéen jobb klikk -> Properties ->

Debug -> General / Open debug launch profiles -> Launch browser: (pipa kivétele)

- Application
- Global Usings
- Build
- Package
- Code Analysis
- Linting
- ▾ **Debug**
 - General**
 - Resources
 - TypeScript

Debug

General

The management of launch profiles has moved to a dedicated dialog. It may be accessed via the link below, via the Debug menu in the menu bar, or via the Debug Target command on the Standard tool bar.

[Open debug launch profiles UI](#)

