

API Web Application-MS Visual Studio Code-.NET Entity Framework

Ver primero todos los videos y luego ver de nuevo replicando los pasos:

https://drive.google.com/drive/folders/1KYm0ozDHziZoGDk6nw1ecHj84_YW7Mfy?usp=sharing

- ✓ Paquetes a instalar en las distintas capas de la solución
- SwashBuckle.AspNetCore 6.5.0 - En el proyecto inicial, en el ejemplo "WebApplication1". Puede ser que venga ya instalado por defecto.
- Microsoft.AspNetCore.WebApi.Cors 5.2.9 – Proyecto "WebApplication1"
- Microsoft.EntityFrameworkCore.Design 7.0.9 – Proyecto "WebApplication1"
- Microsoft.EntityFrameworkCore Core 7.0.0 – Proyecto Entities
- Microsoft.EntityFrameworkCore.SqlServer 7.0.0 – Proyectos Entities y Data
- Microsoft.EntityFrameworkCore.Tools 7.0.0 – Proyecto Data
- Microsoft.Extensions.Configuration 7.0.0.0 – Proyecto Data
- Microsoft.Extensions.Configuration.Json 7.0.0.0 – Proyecto Data
- ✓ Eliminar

`<PrivateAssets>all</PrivateAssets>`

- ✓ Añadir

`<GenerateRuntimeConfigurationFiles>True</GenerateRuntimeConfigurationFiles>`

En el paso 5 a vosotros no os dará error porque ya tendréis instalado el paquete que a mí me faltaba

- ✓ Para conectar al servidor, en appsettings.json

```
"AllowedHosts": "*",
"ConnectionStrings": {
  "ServiceContext": "(Pega aquí tu conexión string);Initial
Catalog=APIWEB2;Integrated
Security=True;MultipleActiveResultSets=True;App=EntityFramework;TrustServerCertificate=True"
}
```

- ✓ Añadir constructor en Program

```
builder.Services.AddDbContext<ServiceContext>(
    options =>
options.UseSqlServer("name=ConnectionStrings:ServiceContext"));
```

- ✓ Para establecer las relaciones entre tablas, mirad la sección "Mapping relationships in EF Core" en [este](#) enlace.