

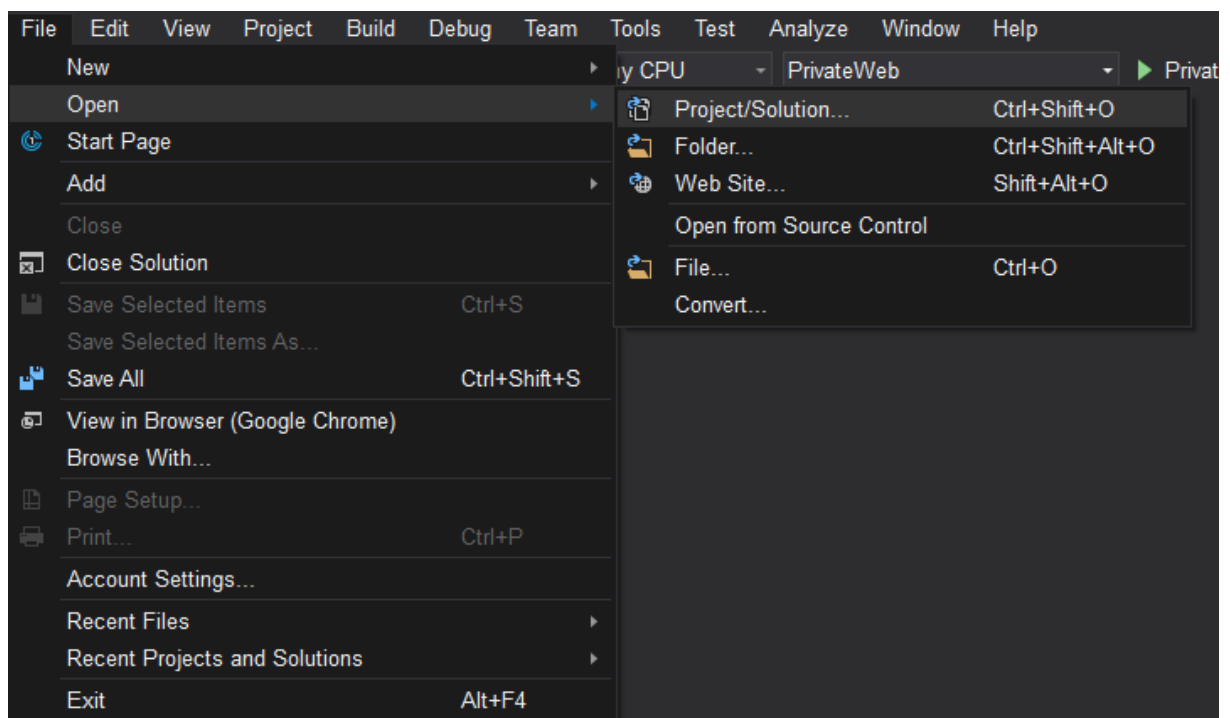
# 1. Fejlesztői környezet beállítása

## 1.1 Előkövetelmények

- .Net Core 2.1 SDK telepítése:  
<https://dotnet.microsoft.com/download/dotnet-core/2.1>
- Visual Studio 2017 / 2019 telepítése:  
<https://visualstudio.microsoft.com/downloads/>
- Microsoft Sql Management Studio telepítése:  
<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>

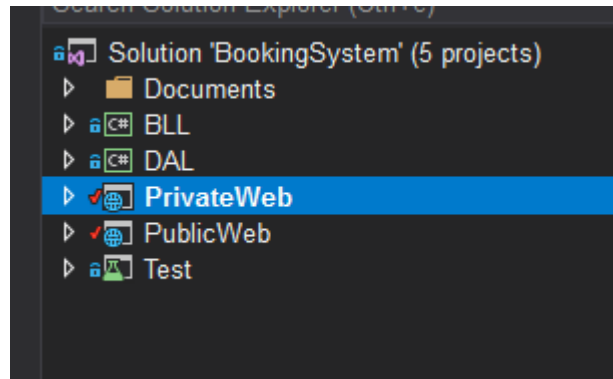
## 1.2 Fejlesztői környezet beállítása

A Visual Studio telepítése utána az alkalmazás Solutionje már megnyitható.



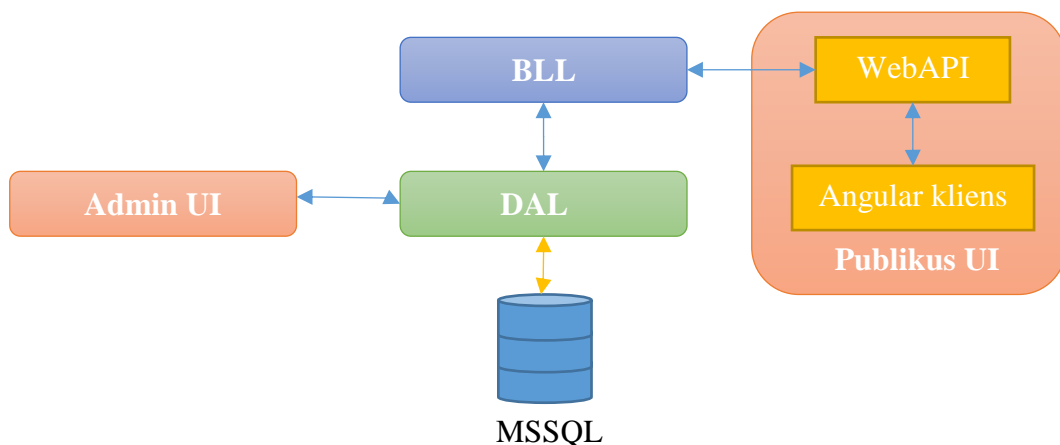
1. ábra: Solution megnyitása

A megnyitás után a solution projektjei a Solution Explorer fülön látszódnak

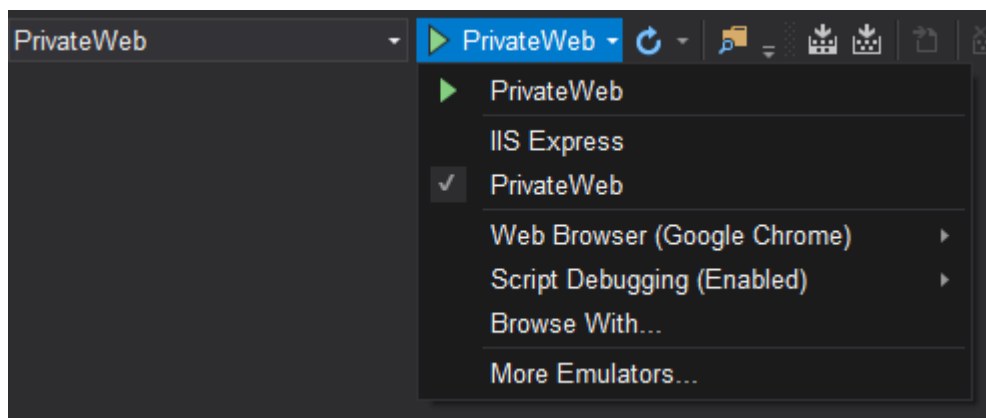


**2. ábra: Az alkalmazás projektjei**

A DAL projekt az alkalmazás adatbázis elérésért felelős. A séma itt van definiálva, ami Code Firstel készült. A BLL-ben valósul meg az üzleti logikai réteg. A PrivateWeb és a PublicWeb a két UI réteg.



Az ügyintéző portált a PrivateWeb projekt valósítja meg. Ez egy ASP.NET Core MVC webalkalmazás. Az alkalmazást a fenti menüsorból lehet elindítani, miután kiválasztottuk, hogy az alkalmazás miként fusson. A debugolás úgy működik, mint a többi .Net alkalmazás esetén. Miután elindult az alkalmazás, a futás során a debug pontok helyén megáll az alkalmazás (<https://docs.microsoft.com/en-us/visualstudio/debugger/how-to-enable-debugging-for-aspnet-applications?view=vs-2019>).



**3. ábra: PrivateWeb alkalmazás indítása**

A PublicWeb alkalmazás alapja szintén egy ASP.NET Core MVC projekt. Az eltérés annyi, hogy a kliens egy Angular alkalmazás. Indítás után az SPA middleware elindítja a kliens alkalmazás. A debugoláshoz fontos bekapcsolni a Script Debugging opciót, így a TypeScript állományokban is megtalálja a debug pontokat.