

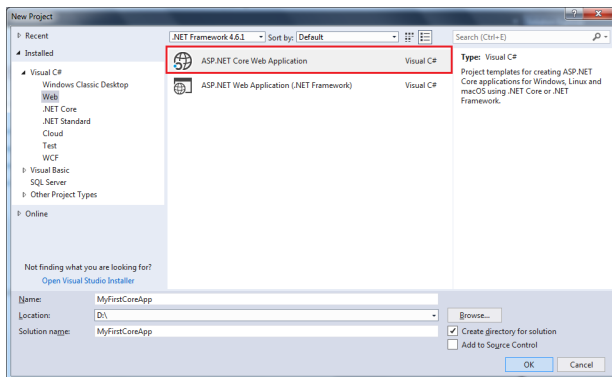
# A .NET Core keretrendszer

Jánosi-Rancz Katalin Tünde

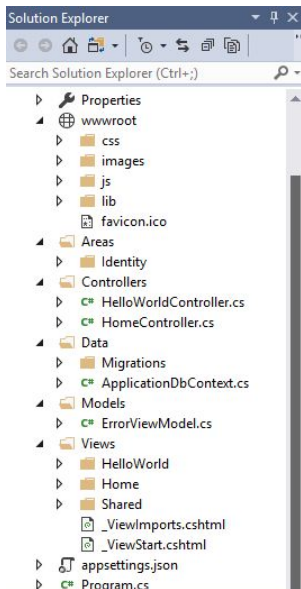
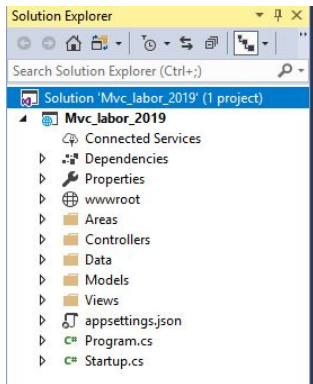
Sapientia EMTE  
tsuto@ms.sapientia.ro

# A .NET Core keretrendszer

- ▶ .NET Core 3.0.0-preview3 Verzió
- ▶ Cross-platform (Windows, Linux, macOS)
- ▶ Modularizált felépítés, csak az alkalmazáshoz szükséges komponenseknek kell jelen lennie.
- ▶ Nyílt forráskód (<https://github.com/dotnet/core>)



# Tegyünk mindent a helyére



- ▶ belépési pont

```
public class Program
{
    0 references | 0 exceptions
    public static void Main(string[] args)
    {
        BuildWebHost(args).Run();
    }

    1 reference | 0 exceptions
    public static IWebHost BuildWebHost(string[] args) :
        WebHost.CreateDefaultBuilder(args)
            .UseStartup<Startup>()
            .Build();
}
```

- ▶ ASP.NET Core applikációk beállítanak és elindítanak egy hostot. Ez a host lesz a felelős az applikáció elindításáért és élettartamának a vezérléséért (app startup and lifetime management)

- ▶ alkalmazásszintű vezérlés
- ▶ az alkalmazás elindítása után először hatjodik végre, lásd Program.cs

```
public class Startup
{
    // This method gets called by the runtime. Use this method to add services to the container.
    // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?linkid=213263
    // 0 references | 0 exceptions
    public void ConfigureServices(IServiceCollection services) { ... }

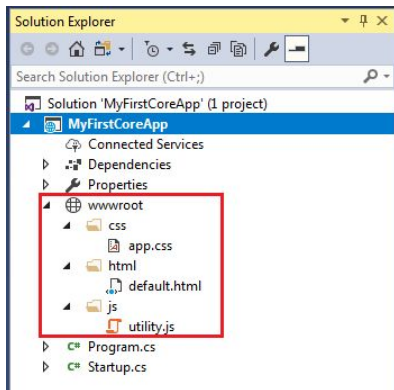
    // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
    // 0 references | 0 exceptions
    public void Configure(IApplicationBuilder app, IHostingEnvironment env) { ... }
}
```

- ▶ Minimálisan: a host beállítja szolgáltatásokat és egy kérés feldolgozó pipeline
- ▶ a **ConfigureServices** az a hely, ahol regisztrálni kell az osztály függőségeket (DI- Dependency Injection), ASP.NET Core a függő osztályokra Service-ként hivatkozik. Vagyis, amikor olvassuk, hogy "Service", akkor értsük mint osztály, amelyik fel lesz használva egy másik osztályban

```
public void ConfigureServices(IServiceCollection services)
{
    //register framework services
    services.AddMvc();
}

// This method gets called by the runtime. Use this method to
// register the middleware. Order is important for the middleware pipeline.
// Use the Order property to rearrange the order of the middleware.
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    app.UseDeveloperExceptionPage();
    app.UseStatusCodePages();
    app.UseStaticFiles();
    app.UseMvcWithDefaultRoute();
}
```

- ▶ meghatározzuk a keretrendszert, ami a későbbiekben majd megengedi a Razor használatát a View-kban.
- ▶ UseDeveloperExceptionPage: Szinkron és Aszinkron hibákat, úgynevezett Exception példányokat rögzít a pipeline-ból és HTML hibát hoz létre válaszként
- ▶ UseStatusCodePages: egy státuskód köztes réteget hoz létre egy alapértelmezett válaszkezelővel, amely ellenőrzi a 400 és 599 közötti állapotkódokat, amelyeknek nincs testük.
- ▶ UseStaticFiles: engedélyezi a statikus fájlkiszolgálást az adott opciókkal
- ▶ default Routing



- ▶ A wwwroot-ban resource-okat, publikus statikus állományokat
- ▶ csak azok a fileok, amelyek benne vannak a wwwroot folderben, csak azok kézbesíthetők http kérésen keresztül
- ▶ minden más file blokkolva van és nem használható alapértelmezetten
- ▶ ezek a statikus fileok elérhetőek a base URL és filenév segítségével:  
**http://localhost:<port>/css/app.css**





- ▶ **Area** lehetővé teszi, hogy nagy alkalmazásokat felosszunk kisebb unitokra, ahol minden unitnak lehet MVC struktúrája

