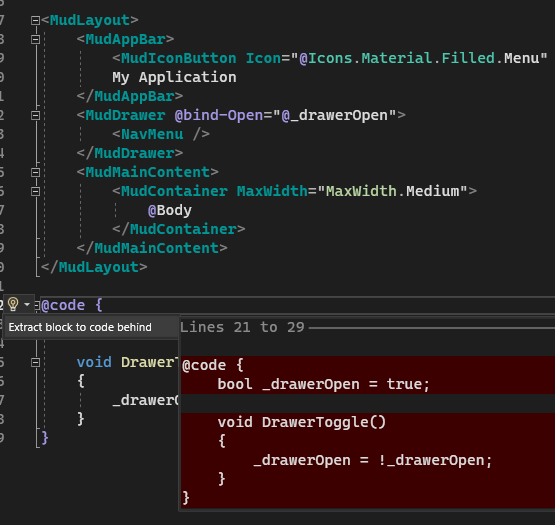
# Mudblazor téma és téma választó

Az előző részben elkészült egy mudblazor app annak felülete. A mostani tananyagban MudBlazor alapokon témákat állítunk be, megvalósítjuk a sötét és világos téma közötti váltást és a választott témát elmentjük a böngészőbe tárolva, ha újra letöltjük a weblapot, a legutolsó téma legyen az aktuális.

## Előkészületek

* [C# Partial Classes and Methods (tutorialsteacher.com)](https://www.tutorialsteacher.com/csharp/csharp-partial-class)
* [Tip: Quickly create the code behind of a Razor page - Blog of Christophe Peugnet](https://www.peug.net/en/tip-quickly-create-the-code-behind-of-a-razor-page/)



A jelenlegi applikációnkban a MainLayout.razor állomány az amelyik a weblap felépítését írja le. A razor állományok mellé parciális osztályként az oldalon futó kódokat lehet létrehozni. Készítsünk a fentiek alapján az oldalhoz egy mögöttes kódot (code behind) a MainLayout.razor oldalhoz:

* Code behinde elkészítése a MainLayout.razor oldalhoz

public partial class MainLayout

{

bool \_drawerOpen = true;

void DrawerToggle()

{

\_drawerOpen = !\_drawerOpen;

}

}

### Sötét és világos színpaletta

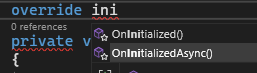
* [Implement MudBlazor Themes for Dark & Light Modes (Themes) - YouTube](https://www.youtube.com/watch?v=vVIYgXmf5T8)
* [Theming - MudBlazor](https://mudblazor.com/customization/overview#theme-provider)
* [Blazor Material UI - Configuration and Theme Customization - Code Maze (code-maze.com)](https://code-maze.com/blazor-material-ui-configuration-and-theme-customization/)

Lépések:

* Legyen egy változónk amely meghatározza, hogy melyik az aktuális paletta (sötét vagy világos)
* Legyen aktuális, sötét és világos palettánk
* Legyenek metódusaink, amelyek beállítják a világos és sötét palettát
* Legyen egy metódus, amely beállítja az aktuális témát

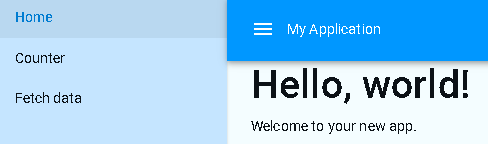
Írjuk felül az osztály OnInitializedAsync metódusát.

Az OnInitialized és OnInitializedAsync akkor hívódik meg, amikor a komponens inicializálódik, miután a SetParametersAsync-ben megkapta a kezdeti paramétereit.

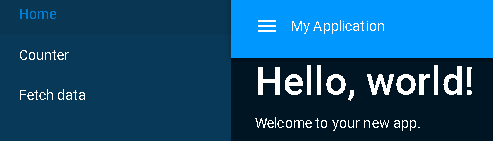


* Állítsuk be a világos és sötét témát és állítsuk be melyik az aktuális
* A razor oldalol a MudThemeProvdier kapja meg az aktuális témát paraméterként

Működik a világos téma:



És a sötét téma, ha azt állítjuk be induláskor:

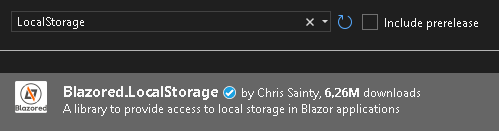


### Téma elmentése a böngésző tárhelyén

* Készítsünk metódust, amely visszaadja a téma nevét, amit el tudunk menteni a böngésző tárhelyén.

Használjuk a LocalStorage osztályt és a Blazored.LocalStorage csomagot:

* [Using Localstorage To Cache Data In Blazor Applications (c-sharpcorner.com)](https://www.c-sharpcorner.com/article/using-localstorage-to-cache-data-in-blazor-applications/)
* [Local Storage in Blazor WebAssembly .NET 6 - Blazor School](https://blazorschool.com/tutorial/blazor-wasm/dotnet6/local-storage-926535)



[Inject] public ILocalStorageService? LocalStorage { get; set; }

Ebben a kódrészletbe a dependency injection technikát alkalmazzuk a LocalStorage elérésére:

* [ASP.NET Core Blazor dependency injection | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/blazor/fundamentals/dependency-injection?view=aspnetcore-8.0)
* [Dependency Injection in Blazor Server .NET 7 - Blazor School](https://blazorschool.com/tutorial/blazor-server/dotnet7/dependency-injection-785774)

### Téma elmentése és témák közötti váltás

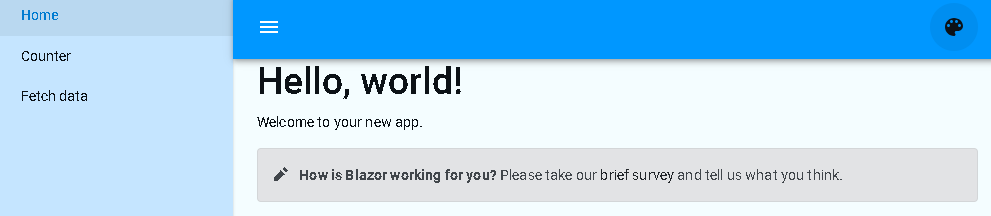
* Készítsünk egy metódust, amely témát vált és elmenti a LocalStorage-ba az aktuális témát

Ezután megvalósítjuk a témák közötti váltás egy ikon gombbal:

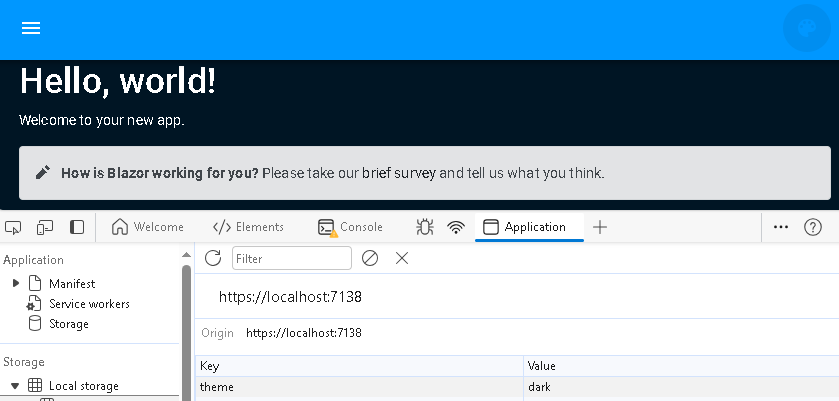
* [MudBlazor IconButton - Blazor Component Library](https://mudblazor.com/components/iconbutton)
* [Spacer - MudBlazor](https://mudblazor.com/components/spacer)

A LocalStorage egy service, a dependency injection akkor működik, ha a servicet felvesszük a Program.cs-ben.

A váltás a témák között működni kezd:



A téma neve a LocalStorage-ba mentésre került (a DevTools segítségével ellenőrizhető).



* Az aktuális témát kiolvassuk a LocalStorage-ból, ha ott megtalálható

Így induláskor az utoljára elmentett téma jelenik meg.