

Ez a dokumentum leírja, mik szükségesek egy DotNetNuke modul fejlesztéséhez.

## Vázlat

- Telepítés és beállítás (OK)
- Modulprojekt létrehozása és kezdeti beállítása
  - o MVC
  - o A telepített DotNetNuke könyvtárba kell tenni

## Előfeltételek

A fejlesztésre szánt gépre a következő szoftverek telepítése szükséges:

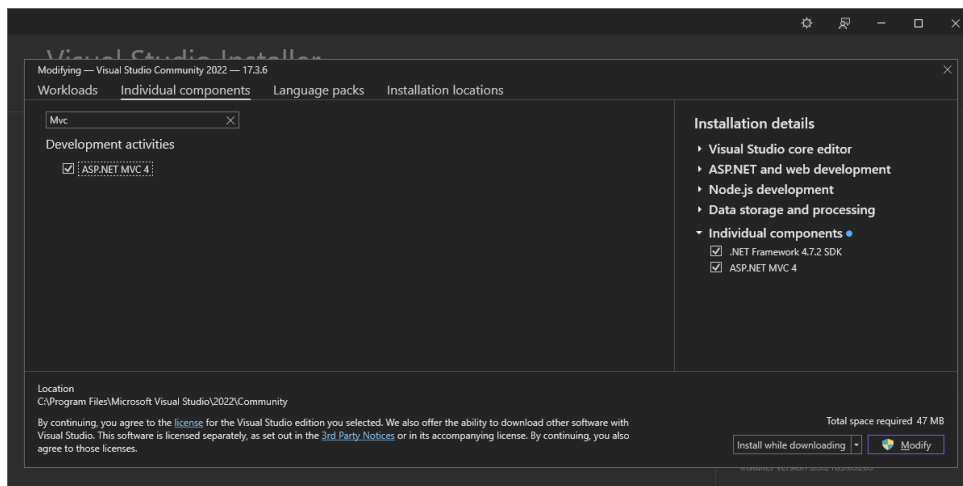
- Visual Studio 2022
- SQL Server Management Studio
- SQL Express
- Valamilyen GIT kliens (A VS2022 integrált kliense is megfelel)

A Visual Studio telepítést bővíteni kell a következő csomagokkal:

- .Net 4.7.2 SDK
- ASP.NET MVC 4

Optionálisan szükséges lehet a következő csomagok telepítése is:

- .Net Desktop Development payload



## Helyi DotNetNuke példány telepítése

A korábbi telepítést leíró dokumentum alapján telepíteni kell egy DotNetNuke példányt a fejlesztésre szánt számítógépen. A leíráshoz képest a fejlesztésre szánt DotNetNuke példány konfigurációja kis mértékben eltér.

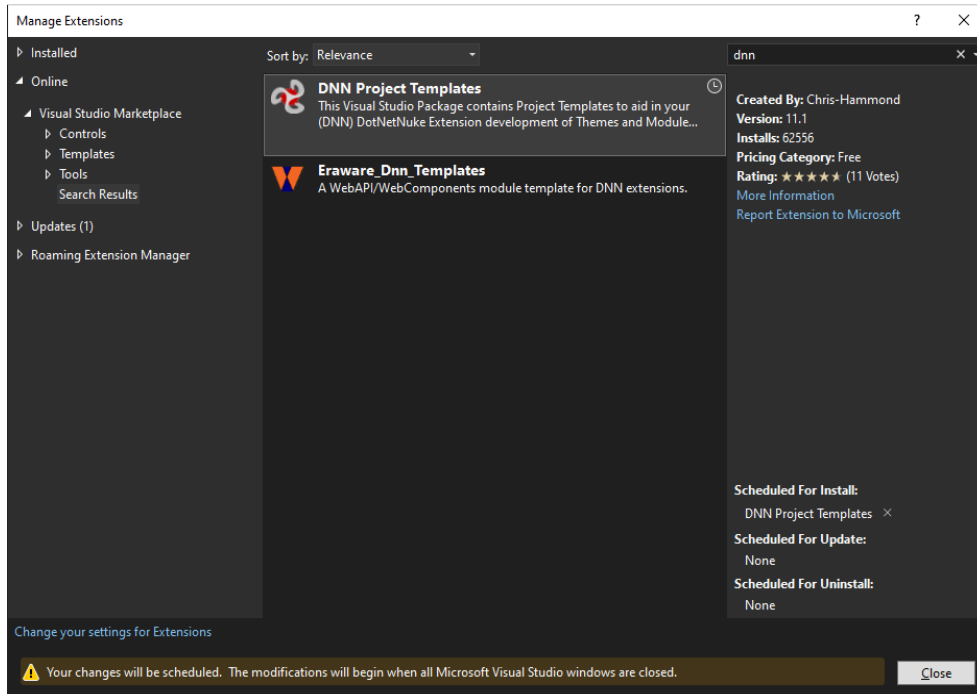
- Egy másik könyvtárba (pl. c:\DotNetNuke\Dev) hajtsuk végre a telepítést
- A neve legyen „dnn-dev”

- A webalkalmazást mindenképpen a „dnndev.me” domain-re állítsuk be

Ezekkel a beállításokkal indítsunk és telepítsünk egy lokális fejlesztői környezetet.

## Visual Studio beállítása

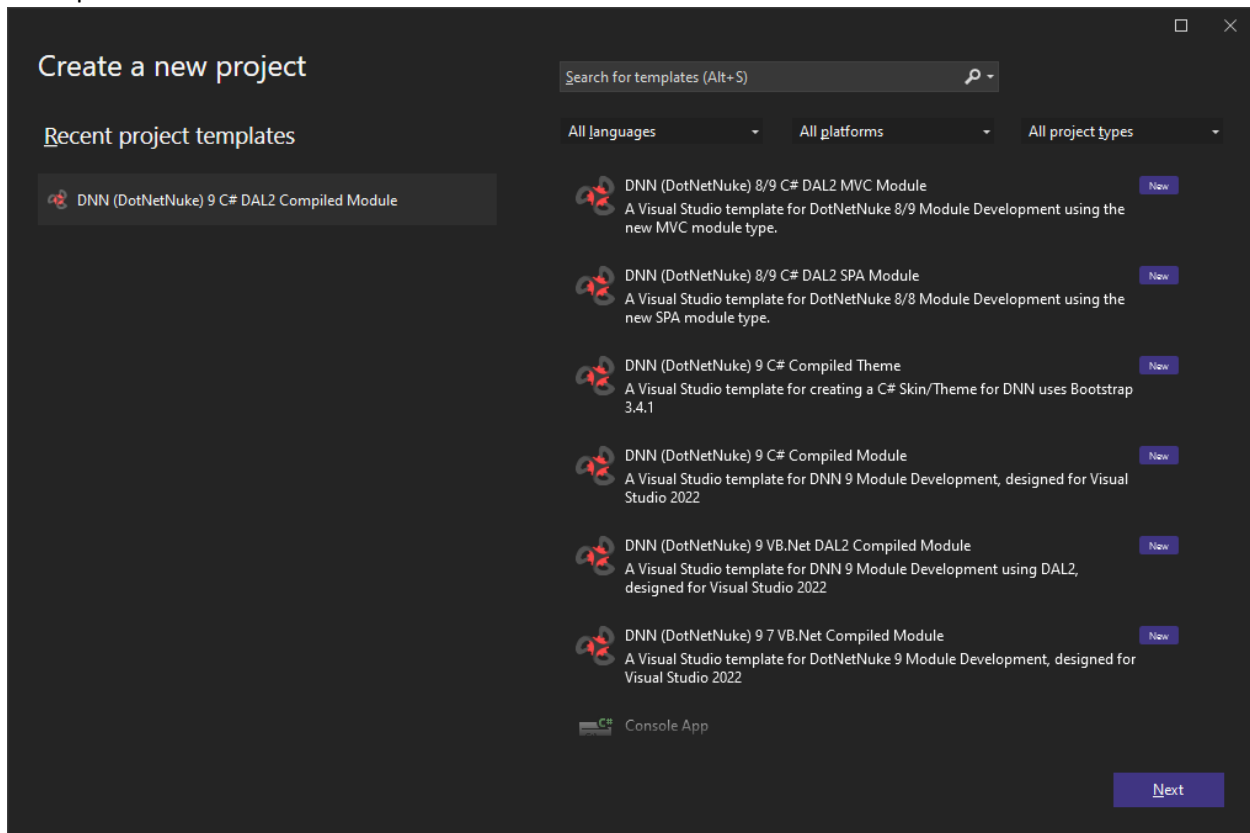
Telepítsük a DotNetNuke projekt sablonokat a Visual Studio-hoz (Extensions/Manage Extensions menü):



A sablonok linkje:

<https://christoc.com/Products/DNN-Development-Templates>

A telepítést követően elérhetővé válnak a DotNetNuke sablonok:



Ezek közül a „DNN (DotNetNuke) 8/9 C# DAL2 MVC Module” sablont fogjuk használni.

## GIT alapfogalmak

A fejlesztés során a verziókövetésre GIT-et használunk majd. Ebben a dokumentumban a következő GIT és verziókövetési alapfogalmak ismeretét feltételezzük:

- Working copy
- Commit és delta
- Branch
- Repository
- Remote és Local repository
- Push/pull/fetch műveletek
- .gitignore fájl

Amennyiben ezek a fogalmak és műveletek nem teljesen világosak, erősen ajánlott a folytatás előtt a GIT ismeretek frissítése.

Mivel többféle GIT verziókövető is elérhető, és a verziókövető eszköz kiválasztása sokszor egyéni preferenciák mentén történik, a GIT műveleteket parancssori parancsokkal illusztráljuk.

## Hello world modul elkészítése

A „Hello World” modul egy ún. „POC” – proof-of-concept modul, ami azt jelenti, hogy az egyetlen célja, hogy teszteljük, hogy létre tudunk-e hozni egy DotNetNuke modult. Mag a „proof-of-concept” azt jelenti, hogy az elméletileg helyesnek tűnő megoldás gyakorlati működőképességét igazoljuk egy példával.

Ebben a projektben a következők a céljaink:

- Fejlesztőkörnyezet teljességének a tesztelése
- Projektstruktúra kialakítása a fejlesztőeszközök segítségével
- Egyszerű modul létrehozása
- Modul hibakereshetőségének („debuggolhatóságának”) tesztelése
- Verziókövetési stratégia kezdeti kialakítása

## Munkakönyvtárak

A DNN sajátosságai miatt a projekteket a DNN telepítés könyvtárában kell létrehozni. A modulok a DesktopModules könyvtárban találhatóak, ezért nekünk is ide kell dolgozunk. A DNN és az általunk használt template van olyan rugalmas, hogy az összetartozó moduljainkat ezen belül saját könyvtárba gyűjthetjük. Ez utóbbi a verziókövetés miatt fontos. (Pl.:

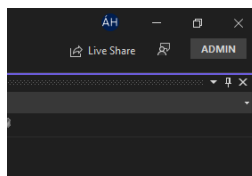
c:\DotNetNuke\Dev\DesktopModules\YourRFProject)

A gyökérkönyvtárban csak egy .gitignore fájl legyen. A .gitignore fájl lehet valamilyen .Net fejlesztéshez előre elkészített fájl, vagy kézzel is létrehozhatjuk. Utóbbi esetben a szabályok kialakításánál figyeljünk, hogy a következő kategóriákba eső fájlokat *NE* kövessük:

- Generált forrásfájlok
- Fordítás eredménye (obj és bin könyvtárak)
- Harmadik féltől származó, csomagkezelő segítségével telepített csomagok (pl. NuGet vagy NodeJS modulok)
- Ideiglenes könyvtárak tartalma (pl. log fájlok, vagy alkalmazás tesztelése közben létrehozott fájlok)

## Projekt létrehozása

**!! FONTOS: DotNetNuke modulfejlesztés során a Visual Studio-t MINDIG „Administrator” jogosultságokkal kell indítani, mivel fejlesztés közben az IIS-t csak ilyen jogkörrel indítva tudja vezérelni a Visual Studio.**



*Ha a bal felső sarokban megjelenik az „ADMIN” címke, akkor a Visual Studio rendszergazdai jogkörrel fut.*

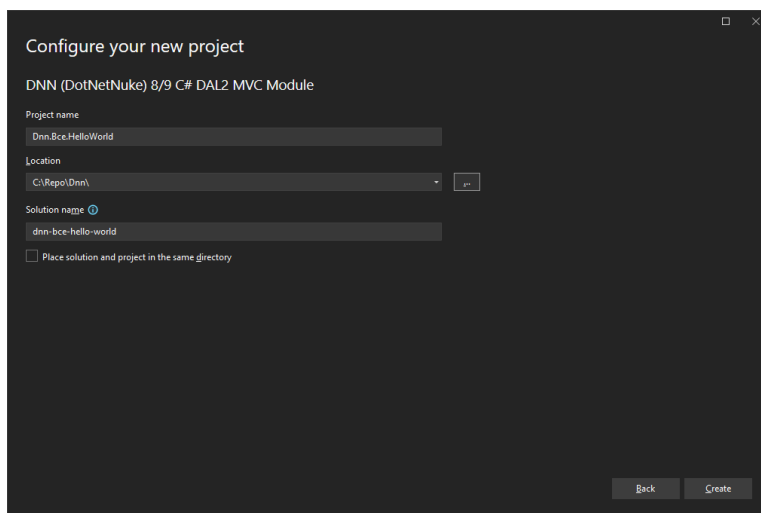
Kezdjük létrehozni egy új projektet. Ehhez a „DNN (DotNetNuke) 8/9 C# DAL2 MVC Module” sablont válasszuk.

- A projekt neve legyen Dnn.ProjektNeve.HelloWorld
- A solution neve legyen „module-helloworld”
- A projektet a telepítés DesktopModules könyvtárában (c:\DotNetNuke\Dev\DesktopModules\YourRFProject) hozzuk létre.

A projekt elnevezése során olyan névkonvenciót használjunk, amivel egyértelműen el tudjuk dönteni egy projektről, hogy DNN modult tartalmaz. Így például használhatjuk az „RFProjekt.Dnn.ValamilyenModul” konvenciót, ahol az „RFProjekt” előtag jelenti a projektünk kódnevét, a „Dnn”, hogy DNN modult ír le, a „ValamilyenModul” pedig a modul által megvalósított funkcionalitást.

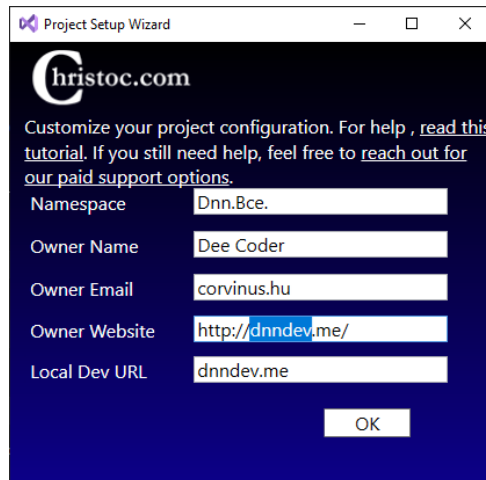
*Sajnos a projekt szót kell használnunk két, tartalmilag eltérő fogalomra. Az egyik projekt fogalmunk a tárgy során az esettanulmány teljes megvalósítása, a másik projekt fogalmunk a Visual Studio projekt. Az esettanulmány során elkészített kódbázist egyetlen Visual Studio megoldásba (solution) célszerű szervezni, de érdemes több rész-projektre bontani.*

*Mivel ez egy minta projekt a feladatmegoldáshoz, ezért helyőrzőként a „Bce” rövidítést használjuk. A beadandó feladatok megoldásánál az esettanulmányban megnevezett cég neve alapján kell létrehozni projektnevet. (Pl. SaudKruger.Dnn.HelloWorld)*



A sablon kér saját beállításokat. Ezek a mostani projekten a következők legyenek:

- Namespace: ProjektNeve.Dnn. (ponttal a végén!)
- Owner: Fejlesztő csapat neve
- Owner e-mail: A csapat kapcsolattartásra használt e-mail címe
- Owner website: Az esettanulmányban megadott cég webhelye
- Local Dev Url: A fejlesztőkörnyezet kialakításakor beállított név (dnndev.me)



## Első build és futtatás

A projekt létrehozása során a Visual Studio számos változtatásokat hajt végre az IIS-re telepített fejlesztői DNN-en, ami miatt az részben vagy teljesen működésképtelenné válhat. A projekt létrehozása után ezért mindenképpen tesztelni kell, hogy milyen állapotban van a DNN telepítésünk, és amennyiben szükséges, a „Problémák és megoldások” fejezetben leírtak szerint javítani.

A projekt tesztelése előtt a modult egy alkalommal telepíteni kell, Ehhez először „DEBUG”, aztán „RELEASE” módban is kell egy-egy build-et futtatni. A „DEBUG” build fogja létrehozni a tárgykódot, a „RELEASE” pedig a telepítendő csomagot. Az elkészült telepítőcsomag a projekt könyvtárán belül az „install” könyvtárban található.

Az elkészült telepítőcsomagot telepíteni kell a DNN admin felületén keresztül. Telepítést követően a további módosításokhoz elég csak DEBUG módban fordítani, és azonnal érvényre jutnak a változtatások.

## Debug

A DNN architektúráis jellemzői miatt a szokásos debug módok nem használhatóak, így az „alkalmazás” elindítása (hiszen nincs is alkalmazás), illetve az „Attach to Process” funkció.

FONTOS! A DNN minden alkalommal újraindul, amikor azt detektálja, hogy a binárisok megváltoztak (például újrafordítottuk a modulunkat)

Ahhoz, hogy a modul debuggolható legyen, mindenképpen „Debug” konfigurációban kell fordítani, illetve a kód releváns részére a következő részletet kell írni:

```
System.Diagnostics.Debugger.Launch();
```

Ez az utasítás a .Net keretrendszeren belül megpróbál csatlakoztatni egy elérhető debugger szolgáltatást – ez praktikusán vagy egy futó Visual Studio példány (vagy egy újonnan indított Visual Studio) Fontos, hogy a debug célokra használt Visual Studio példányt mindig rendszergazdai jogosultságokkal indítsuk el.

Ezt a sort természetesen el kell távolítani a hibakeresés befejeztével!

## Következő lépések

A sablonnal létrehozott egyszerű modul tartalmaz a weboldalba azonnal beépíthető minta modult. A minta modulhoz tartozik konfigurációs felület és adatbázis elérés is.

## Problémák és megoldások

### Jelenség

A projekt/solution létrehozása után a weboldal „Application Error”-t jelenít meg.

### Megoldás

A projekt létrehozása során a Visual Studio felülbírálja a „DesktopModules” könyvtárat egy virtuális könyvtárral. Az IIS Manager segítségével meg kell keresni ezt, ellenőrizni kell, hogy virtuális könyvtár-e. Amennyiben az, törölni kell.

A törlést követően újra elérhetővé válik a fizikai DesktopModules könyvtár, és így helyreáll a DotNetNuke működése.

Ellenőrizni kell azt is, hogy a DesktopModules könyvtáron belül az „mvc” könyvtár virtuális könyvtár-e. Amennyiben az, ezt is törölni kell.

### Jelenség

Nem fordul a frissen létrehozott és beállított DotNetNuke modul projekt

### Megoldás

Előfordulhat, hogy a template-ban előre beállított útvonalhoz képest a DotNetNuke.dll assembly más elérési útvonalon található. Ebben az esetben ellenőrizzék a .csproj fájlban a szerelvényhez tartozó HintPath tulajdonságát. A mintamegoldásban ezt állítottuk be:

```
<Reference Include="DotNetNuke, Version=9.8.0.0, Culture=neutral,  
processorArchitecture=MSIL">  
  <SpecificVersion>False</SpecificVersion>  
  <HintPath>..\..\bin\DotNetNuke.dll</HintPath>  
</Reference>
```

Előfordulhat, hogy több DNN assembly is hivatkozva van, ebben az esetben a többivel is ugyanígy kell eljárnotok.

### Jelenség

Hiába frissítjük a fájlokat, nem látszik változás az alkalmazásban.

### Megoldás

Indítsuk újra a webhelyet az IIS-ben. (Javasolt a „Stop”-„Start” a „Restart” helyett)

### Jelenség

Fordítás közben nem tud a Visual Studio minden fájlt írni.

### Megoldás

A fájlok egy részét az IIS zárolhatja futás közben. Ahhoz, hogy elengedje, le kell állítani a webhelyet, és fordítás után újra kell indítani.

Select Users or Groups

Select this object type:  
Groups Object Types...

From this location:  
VR-DNN Locations...

Enter the object names to select [\(examples\)](#):  
VR-DNN\IIS\_IUSRS Check Names

Advanced... OK Cancel

Permissions for dnn-bce-hello-world

Security

Object name: C:\Repo\Dnn\dnn-bce-hello-world

Group or user names:

- Authenticated Users
- IIS\_IUSRS (VR-DNN\IIS\_IUSRS)
- SYSTEM
- NETWORK SERVICE
- Administrators (VR-DNN\Administrators)
- Users (VR-DNN\Users)

Add... Remove

Permissions for IIS\_IUSRS

	Allow	Deny
Full control	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modify	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read & execute	<input checked="" type="checkbox"/>	<input type="checkbox"/>
List folder contents	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>

OK Cancel Apply