# L03\_ E-shop v ASP.NET - Základ - Migrace a první spuštění

V dnešním ASP.NET Core tutoriálu si vygenerujeme databázi pomocí **migrací**. A také aplikaci poprvé spustíme.

V reálných aplikacích je potřeba neustále měnit databázová schémata s tím, jak se mění modely aplikace. To se navíc musí provést tak, aby se z databáze neztratila žádná data. **Migrace** v Entity Framework Core za nás tento problém řeší.

Jedná se o třídy, které můžeme automaticky generovat pomocí příkazu Add-Migration. Výsledkem je nová třída odvozená od Migration, obsahující metody Up() a Down() (pokud se od poslední migrace model nezměnil, jsou prázdné). Tato třída je uložena v samostatném souboru ve složce Data/Migrations/ v **datovém projektu**.

Metoda Up() popisuje změny, které se v databázi vykonají při **aplikaci** migrace příkazem Update-Database. Metoda Down() naopak umožňuje tyto změny vzít zpět. Ve složce Data/Migrations/ nalezneme už scaffoldnutou úvodní migraci pod názvem CreateIdentitySchema.

Pokud v metodě Up() příslušné migrace nalezneme chybu, upravíme **datový model**, nikoli samotnou migrační třídu. Po úpravě datového modelu odebereme chybnou migraci příkazem Remove-Migration. Poté přidáme novou migraci příkazem Add-Migration a zase zkontrolujeme obsah metody Up() příslušné migrační třídy. Pokud je obsah metody Up() už v pořádku, aplikujeme migraci příkazem Update-Database.

Přesuneme se do projektu Eshop.Data.

### Migrace Identity

Nyní si vytvoříme první vlastní **migraci**. Spusťme si Package Manager Console z menu Tools → NuGet Package Manager. Jako Default project vybereme náš datový projekt Eshop.Data.

V Package Manager Console přidáme novou migraci příkazem Add-Migration. **Název migrace**, pod jejímž jménem se nám vygeneruje partial migrační třída, si zvolíme např. Identity. Celý příkaz je tedy Add-Migration Identity:

Potom aplikujeme migraci příkazem Update-Database.

### Kontrola databáze

Nyní si zkontrolujeme, že se databáze opravdu vytvořila. Z menu View vybereme SQL Server Object Explorer. V nově otevřeném okně rozklikneme naši databázi a poté složku Tables:

Název databáze u nás bude aspnet-Eshop-6-0.

Do tabulek se můžeme podívat přes pravé tlačítko myši a výběrem View Data.

## Aplikační vrstva

Přesuneme se do projektu Eshop.

### Lazy loading

Abychom mohli využít načítání některých, pro to vhodných dat až v případě jejich potřeby zobrazení, přidáme do souboru Program.cs volání metody UseLazyLoadingProxies():

...

// Add services to the container.

var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");

builder.Services.AddDbContext<ApplicationDbContext>(options =>

options.UseSqlServer(connectionString)

// přidali jsme tento řádek:

.UseLazyLoadingProxies());

...

### Dokončení konfigurace

Po provedení všech těchto úprav se nám při pokusu o zkompilování projektu možná zobrazí pár chyb. Pravděpodobně se bude jednat převážně o chyby stejného typu - je potřeba dopsat příkazy using.

Pro rychlé odhalení chyb klikneme v okně Solution Explorer pravým tlačítkem na Solution a vybereme možnost Rebuild Solution. Tím se nám celý projekt sestaví kompletně znovu. Případné nalezené chyby se zobrazí klasicky v okně Error List. Kliknutím na chybu nás to přesune na místo jejího vzniku do příslušného souboru:

Nezapomeneme i na přidání usingu datové vrstvy using Eshop.Data; v souboru Program.cs a dalších, kde je to potřebné.

## První spuštění aplikace

Jelikož databázi máme již připravenou, můžeme aplikaci spustit. Spustí se hlavní (aplikační) projekt, tedy projekt Eshop, který by měl být nastaven jako Startup Project.

Pokud není, nebo bychom někdy Startup Project potřebovali změnit, docílíme toho kliknutím pravým tlačítkem na daný projekt v okně Solution Explorer a vybráním Set as Startup Project:

Po spuštění se nám zobrazí pohled Index ze složky Views/Home/:

Máme tedy připravený základ úvodní stránky a navigační lištu s textovým polem pro vyhledávání produktů a několika navigačními tlačítky, jako například tlačítko pro nákupní košík.

To je pro dnešek vše.

**Pozor**: Budete-li chtít přiloženou vzorovou aplikaci spustit, je potřeba nejprve vygenerovat databázi s daty! Testovací databázi vygenerujeme tak, že v Package Manager Console nejprve vybereme jako Default project Eshop.Data. Poté zadáme příkaz Update-Database. V SQL Server Object Explorer se nám vygeneruje nová databáze se stejným názvem, jako je hodnota v Database z připojovacího řetězce DefaultConnection uloženého v souboru appsettings.json.

# Zdroj: https://www.itnetwork.cz/csharp/asp-net-core/eshop-zaklad/e-shop-v-aspnet-zaklad-migrace-a-prvni-spusteni