

21 Návrh webové aplikace ASP.NET

Wednesday, 19 January 2022 09:05

Razor Pages, MVC, Page, PageModel, Razor syntaxe, Startup.cs, služba, databáze, Entity Framework, Identity

- ASP.NET (*Active Server Pages*)
 - základem je .NET
 - Microsoft framework
 - může běžet i na macOS a Linuxu
 - C#, F#, VB.NET
 - několik stylů tvorby aplikací:
 - i. Razor Pages
 - novější architektura
 - Razor syntax
 - ◆ HTML s možností vkládání C# kódu
 - PageModel
 - modifikace MVC
 - spojení Modelu a Controlleru
 - ◆ třída přiřazená k Page
 - ◆ připojení obecných služeb (db, session, cookies) a dat pro zobrazení ve View
 - ◆ bindování pomocí [BindProperty] a asp-for
 - konfigurace ve Startup.cs
 - ◆ od .NET 6 spojen s Program.cs
 - ◆ ConfigureServices
 - ◇ připojení db, controllerů
 - ◇ autentizace, autorizace, identita
 - ◇ dependency injection
 - ◆ Configure
 - ◇ konfigurace aplikace v závislosti na http requesty
 - ii. MVC
 - klasická architektura
 - oddělení logiky od vstupu
 - ◆ vhodnější pro robustnější aplikace
 - např. ASP.NET MVC, Ruby on Rails, Django
 - a. Model
 - data a logika aplikace
 - b. View
 - zobrazení výstupu uživateli
 - měl by obsahovat minimum logiky
 - c. Controller
 - propojení Modelu a View
 - příjem vstupu od uživatele a převod na příkazy pro Model
 - metody, které se dají volat skrze REST nebo routování, se nazývají akce
 - kvůli routování přes Controller do Modelu a přes Controller zpátky do View může být pomalý
 - iii. Blazor
 - client-side C#
 - kompilace přes WebAssembly
 - přímé navázání na backend přes TCP oproti HTTP
 - prostředky:
 - i. služba
 - pomocná část aplikace

- lze připojit do aplikace a využívat jejích prostředků
- např. služba na nahrávání obrázků
- a. AddSingleton
 - platnost po celý běh aplikace
- b. AddScoped
 - platnost po celý request
- c. AddTransient
 - každý controller a služba dostanou novou instanci
- ii. databáze
 - systém souborů s pevnou strukturou záznamů
 - připojení:
 - a) přímo jako objekt v projektu
 - b) connection string v appsettings.json
 - ◇ umožňuje mít víc databází zároveň
 - pracuje se s ní pomocí relace DbContext
 - migrace
 - u code-first databází
 - provádění změn i po vytvoření databáze
- iii. Entity Framework
 - ORM (*Object Relational Mapping*) pro ASP.NET
 - umožňuje práci s databází skrze virtuální objekty
 - generování SQL dotazů
 - syntax LINQ
- iv. Identity
 - hotové řešení pro přihlašování uživatelů
 - dostupné přes NuGet Package Manager
 - potřebuje Entity Framework
 - rozšíření databáze o další tabulky (Users, Roles)
 - scaffoldování Identity stránek