

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Teze diplomové práce

Vytvoření webové aplikace s využitím .NET 5 a

MVC6

Bc. Martin Brabec

© 2016 ČZU v Praze

Vytvoření webové aplikace s využitím .NET 5 a MVC6

Souhrn

Hlavní náplní této práce je návrh a implementace moderní webové aplikace v prostředí ASP.NET Core 1.0, skládající se z prezentačního webu a administrační části. V rámci práce je také navržena vlastní obecná architektura datového zdroje (CRUDQ) libovolné aplikace, jejíž implementace je dále předvedena ve vyvíjené webové aplikaci. Jako databázový systém je zvolena dokumentová databáze MongoDB, do které je přístupováno právě pomocí autorem navržené architektury. Mezi další významné aspekty práce patří implementace efektivního propojení architektur MVC a MVVM, které je docíleno pomocí vlastního generátoru JavaScriptového datového modelu. Tento generovaný datový model je využíván frameworkem KnockoutJS pro zajištění funkčnosti uživatelského rozhraní. Výsledná webová aplikace je plně funkční jednoduchý CMS, spustitelný na třech nejpoužívanějších desktopových operačních systémech, využívající nejmodernější technologie, nástroje a návrhové vzory.

Klíčová slova: ASP.NET, Core, C#, CMS, MongoDB, CRUDQ, KnockoutJS, MVC, MVVM

Creating web application using .NET 5 and MVC6

Summary

The main concern of this thesis is the design and implementation of advanced Web application in ASP.NET 1.0 Core, consisting of a presentation web and administration section. A part of the thesis is also design of a custom architecture for a data source (CRUDQ) suitable for any application, whose implementation is further demonstrated in the developed web application. Selected database system is document database called MongoDB, which is accessed using the author designed architecture approach. Other important aspects of the work includes for example effective interconnection between MVC and MVVM architectures, which is achieved by custom generator of JavaScript data model. The generated data model is used by KnockoutJS framework to ensure the functionality of user interface. The web application is fully functional simple CMS, executable on three most widely used desktop operating systems, using the latest technology tools and design patterns.

Keywords: ASP.NET, Core, C#, CMS, MongoDB, CRUDQ, KnockoutJS, MVC, MVVM

1 Úvod

Rozmanitost technologií a platforem na tvorbu webových stránek se v posledních letech značně zvýšila. Nabízí možnosti použití od statické webové prezentace až po velmi komplexní webové aplikace. To vše většinou v příjemném a uživatelsky přívětivém prostředí.

Tato práce obsahuje přiblížení technologií, nástrojů, architektur a návrhových vzorů, ze kterých je následně sestaven plně funkční jednoduchý CMS systém, využívající celou řadu moderních a v praxi často využívaných technologií a nástrojů. Kromě již existujících architektur a návrhových vzorů tento CMS systém využívá autorem navrženou architekturu datového zdroje a vlastní implementaci propojení dvou běžně používaných architektur webových stránek.

2 Cíl práce a metodika

Cílem práce je s využitím nejnovějších nástrojů, postupů a vzorů v oblasti vývoje webových stránek navrhnout a implementovat moderní webovou aplikaci, sestávající z webové prezentace a související administrační části. Administrační část aplikace bude sloužit pro správu webového obsahu a pro základní nastavení vzhledu webové prezentace. Výsledná webová aplikace bude implementována v multiplatformním prostředí Microsoft ASP.NET 5 v jazyce C# s použitím MVC 6. Databázový systém, který bude pro webovou aplikaci zvolen, musí vyhovovat jejím požadavkům a musí být také multiplatformní. Mezi dílčí cíle patří návrh vlastní architektury datové základny a vytvoření efektivního propojení klientské a serverové části webové aplikace.

Metodika práce je založena na analyticko-syntetickém přístupu. Bude provedena analýza odborných informačních zdrojů a na základě syntézy zjištěných poznatků budou popsány vlastnosti nových technologií a způsob jejich použití. Těchto poznatků bude dále využito při návrhu a implementaci webové aplikace, jejich dílčích součástí, a také při návrhu architektury datové základny. Na závěr bude popsán a zhodnocen postup návrhu, implementace, testování a nasazení.

3 Aplikace Scms

Aplikace Scmc je jednoduché CMS, implementované v jazyce C# na platformě ASP.NET Core 1.0. Verze běhového prostředí je 1.0.0-rc2-15546. Zdrojové soubory aplikace jsou publikovány jako jeden balíček obsahující předkompilované soubory. Tento balíček, společně se zdrojovými kódy, je k této práci přiložen jako příloha.

Mezi hlavní pomocné nástroje, které velmi výrazně zefektivnily práci a zkrátily implementační čas, patří především framework KnockoutJS v kombinaci s autorem navrženým generátorem JavaScriptového modelu. Ve spojení s autorem navrženou architekturou datové základny webové aplikace tak vytvořili kombinaci, jejíž implementační doba byla zkrácena na minimum, avšak i přes to byla zachována přehlednost a relativní jednoduchost kódu.

Díky tomu, že aplikace využívá MongoDB, je připravena na splnění požadavku na spustitelnost na různých operačních systémech. Bohužel, ovladač databáze pro MongoDB, kompatibilní s .NET Core frameworkem, je ještě ve vývoji. Z toho důvodu v současné době aplikaci není možné spustit na jiném operačním systému než Windows. Všechny ostatní nástroje a frameworky, které byly použity, jsou již multiplatformní a mají takové licence, které nebrání využití Scms bez jakýchkoliv poplatků. Tímto byl splněn i požadavek na finanční nenáročnost.

4 Architektura CRUDQ

Součástí této práce je návrh vlastní architektury datového přístupu, využitelného v libovolném projektu. Tato architektura je ukázkově využita v implementované aplikaci Scms.

Architektura CRUDQ rozděluje veškeré datové operace na pět základních typů, přičemž každý typ operace lze aplikovat na libovolnou entitu z datového modelu. Všechny operace mají pevně dané místo v objektové struktuře, i ve struktuře souborového systému. Díky použití typu Interface, jako základního stavebního prvku, je navržená architektura nejen velmi dobře testovatelná a vhodná pro použití v kombinaci s Dependenci Injection, ale především umožňuje snadnou záměnu databázového systému, čímž splňuje jeden z hlavních požadavků na tuto architekturu.

5 Propojení MVC a MVVM

V rámci této práce autor navrhl mimo jiné vlastní řešení pro spojení dvou dobře známých architektur – MVC a MVVM. Každá z nich nalezne uplatnění na jiných místech, avšak lze je velmi efektně a efektivně propojit. Jejich propojení docílil autor vytvořením vlastního generátoru JavaScriptového datového modelu. Díky tomuto generátoru je možné na serverové i klientské straně používat naprosto stejný datový model entit, přičemž ten na klientské straně je obohacen ještě o takzvaný Dependency Tracking, zajišťující velmi snadné provázání s uživatelským rozhraním.

6 Závěr

Hlavní cíl práce se podařilo splnit návrhem a implementací plně funkčního CMS systému Scms, skládajícího se z webové prezentační části a administrační části. Jak se uvádí v kapitole 4.1, administrační část je složena ze správy stránek, správy navigace a z nastavení vzhledu, čímž splňuje požadovanou funkcionalitu. Dle zadání je tento systém vytvořen v prostředí ASP.NET Core 1.0 (dříve ASP.NET 5, viz. 3.1.1), s využitím frameworku MVC 6. Požadavek na zvolení vhodného, multiplatformního databázového systému byl splněn v kapitole 5.3 výběrem dokumentové databáze MongoDB.

Navržením a implementací vlastní architektury CRUDQ v kapitole 5.5 byl splněn dílčí cíl návržení architektury datového zdroje. Naplnění druhého dílčího cíle, a sice implementaci vlastního řešení pro efektivní propojení serverové a klientské strany webové aplikace, bylo docíleno v kapitole 6.3 vytvořením vlastního generátoru JavaScriptového datového modelu, obsahujícího funkcionalitu pro použitý MVVM framework KnockoutJS.

7 Vybrané zdroje

- [1] S. Hanselman, „ASP.NET 5 is dead,“ 19. 1. 2016. [Online]. Dostupné: <http://www.hanselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10AndNETCore10.aspx>.
- [5] S. Smith, „Introducing .NET Core,“ 2015. [Online]. Dostupné: <https://docs.asp.net/en/latest/conceptual-overview/dotnetcore.html>.
- [6] Community, „GitHub - CoreFX,“ Microsoft, 2016. [Online]. Dostupné: <https://github.com/dotnet/corefx>. [Přístup získán 2016].
- [18] Microsoft, „Building your first MVC 6 application,“ Microsoft, 2016. [Online]. Dostupné: <http://docs.asp.net/en/latest/tutorials/first-mvc-app/index.html>. [Přístup získán 6. 2. 2016].
- [31] DevIQ, „Explicit Dependencies Principle,“ DevIQ, 2014. [Online]. Dostupné: <http://deviq.com/explicit-dependencies-principle/>.
- [20] O. Joe a M. Jason, Pablo's SOLID Software Development, 2009.