

РОЗРОБКА ВЕБ-ОРІЄНТОВАНИХ ДОДАТКІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ASP.NET 5

Вінницький національний технічний університет

Анотація

Дана стаття містить інформацію про використання технології ASP.NET 5 при розробці веб-орієнтованих додатків, цілі, переваги та перспективи використання даної технології.

Ключові слова:

ASP.NET 5, MVC, WEB API, Web Pages, кросплатформність, IIS, self-hosting, CLR.

Abstract

This article contains information about the involvement of the technology ASP.NET 5 in development of web-oriented applications, goals, advantages and using prospects of this technology.

Keywords:

ASP.NET 5, MVC, WEB API, Web Pages, Cross-platform, IIS, self-hosting, CLR.

ASP.NET 5 — це новий кросплатформний фреймворк для побудови сучасних хмарних веб-додатків з використанням .NET. У порівнянні з попередніми версіями, він був повністю перебудований, для того щоб забезпечити оптимізовану та надійну платформу для розробки додатків, які будуть розгорнуті в хмарі або працюватимуть на власних серверах. Для підтримки гнучкості при побудові рішень дана платформа складається із модульних компонентів з мінімальними накладними витратами [1].

ASP.NET 5 включає в себе наступні можливості:

- Нове гнучке середовище виконання.
- Новий модульний конвеєр для HTTP-запитів.
- Конфігурація готова для використання в хмарі.
- Уніфікована програмна модель, яка поєднує MVC, Web API и Web Pages.
- Можливість побачити зміни без повторної побудови проекту.
- Використання декількох версій .NET Framework пліч-о-пліч.
- Можливість self-hosting або хостингу на IIS.
- Використання нових інструментів в Visual Studio 2015.
- Відкритий вихідний код в GitHub.

Усі зміни, які були зроблені в для ASP.NET 5 були основані на запитих клієнтів і зворотному зв'язку. Ці зміни спрощують розробку, хостинг і обслуговування і орієнтовані на сучасні веб-додатки [2].

ASP.NET 5 пропонує велику гнучкість, маючи можливість працювати в трьох варіантах кросплатформного середовища виконання:

- Full .NET CLR.
Full .NET CLR — середовище виконання за замовченням в Visual Studio. Воно забезпечує весь набір API и це найкращий вибір для зворотної сумісності.
- Core CLR (оптимізоване для хмари середовище виконання, cloud-optimized runtime).
Core CLR — середовище виконання для проектів ASP.NET 5. Воно позбавлене від усього лишнього и являється повністю модульним. Це CLR було перероблено на компонентну модель, де потрібно окремо додавати ті компоненти, що необхідні для розробки. Core CLR займає біля 11 мегабайт замість біля 200 мегабайт для повної версії .NET CLR.
- Кросплатформне середовище CLR.
Середовище виконання, що дозволяє розробляти і запускати .NET-додатки на Mac та Linux пристроях. До релізу використовується Mono CLR для кросплатформної розробки.

ASP.NET 5 дозволяє розміщувати додаток на IIS або в режимі self-hosting. Якщо використовується Core CLR, додаток можна розгорнути зі всіма залежностями, зібраними в пакет розгортання. Таким чином, додаток і його залежності повністю автономні і більше не залежать від установки .NET в системі. Додаток може працювати на будь-якому типі пристрою або хостингової платформи.

В минулому MVC, Web API та Web Pages включали різні реалізації схожого функціоналу. Наприклад, MVC і Web API реалізують маршрутизацію, але класи маршрутизації MVC знаходяться в просторі імен System.Web.Mvc.Routing, а схожі класи Web API знаходяться в System.Web.Http.Routing. Або, Web Pages та MVC використовують синтаксис Razor, але деякі NuGet пакети сумісні тільки з одною або другою реалізацією. В ASP.NET 5, MVC, Web API та Web Pages об'єднані в єдиний фреймворк під назвою MVC 6. Це злиття усуває дублювання із фреймворку і робить розробку додатків легшою. Щодо фреймворку ASP.NET Web Forms, то він не був включений в ASP.NET 5 і вважається застарілим.

ASP.NET 5 надає новий, легкий спосіб управляти залежностями в проектах. Більше не потрібно додавати посилання на збірки в проект, замість цього програміст управляє залежностями, посилаючись на NuGet пакети. Існує можливість або додати NuGet пакети за допомогою менеджера NuGet пакетів або відредагувати файл JSON (project.json), в якому перераховані NuGet пакети і версії, що використовуються в проекті. Для додавання інших залежностей, в файл project.json дописується назва і номер версії пакета NuGet.

Файл project.json включає тільки NuGet пакети, які були безпосередньо добавлені в проект. Якщо додати NuGet пакет, який залежить від інших пакетів, ці вторинні залежності завантажуються, але не перераховані у файлі project.json. Цей підхід дозволяє зберегти файл project.json не загроможденим і простим в керуванні. Якщо видалити NuGet-пакет із project.json, вторинні залежності видаляються також, якщо немає інших пакетів, яким вони потрібні.

Завдяки формату JSON керувати залежностями досить легко, навіть якщо немає установленної Visual Studio. Так як можна відкрити файл project.json в будь-якому текстовому редакторі і внести зміни, наприклад, оновити залежності додатку, розгорнутого в хмарі.

Visual Studio 2015 надає досить легкий підхід в розробці ASP.NET додатків. При внесенні змін в код і їхньому збереженні, достатньо оновити сторінку в браузері. Всі зміни будуть відображені без повторної перебудови проекту [3].

Visual Studio використовує Roslyn для динамічної компіляції. Програмісту як і раніше доступна вся потужність компілюемого фреймворку, але при розробці складається враження використання інтерпретованої мови програмування.

Весь код для ASP.NET 5 доступний на GitHub, де розробники можуть побачити, які саме зміни були внесені, завантажити код і внести свої зміни. Це значно полегшує розуміння коду і спрощує створення власних реалізацій ASP.NET.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ / REFERENCES

1. Introduction to ASP.NET 5 [Електронний ресурс] — Режим доступу: <http://docs.asp.net/en/latest/conceptual-overview/aspnet.html>.

2. Обзор ASP.NET 5 [Електронний ресурс] — Режим доступу: <https://habrahabr.ru/post/243667/>.

3. Top 10 Changes in ASP.NET 5 and MVC 6 [Електронний ресурс]. — Режим доступу: <http://stephenwalther.com/archive/2015/02/24/top-10-changes-in-asp-net-5-and-mvc-6>.

Череватов Василь Володимирович, студент групи 2ПІ-126 Факультету інформаційних технологій і комп'ютерної інженерії Вінницького національного технічного університету, м. Вінниця, e-mail: cherevatov5@mail.ru.

Науковий керівник – **Кательніков Денис Іванович**, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, E-mail: fuzzy2dik@gmail.com.

Cherevatov Vasyly Volodymyrovich, student of the group 2PE-12b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: cherevatov5@mail.ru.

Research director – **Katelnikov Denys Ivanovych**, PhD, Associate Professor of Software Engineering Department, Vinnytsia National Technical University, Vinnytsia, E-mail: fuzzy2dik@gmail.com.