Вступ. Шаблон MVC. Middleware.

№ уроку: 1 **Курс:** ASP.NET Core

Засоби навчання: Visual Studio 2019, .NET Core SDK

Огляд, мета та призначення уроку

Вивчити шаблон MVC (Model View Controller). Ознайомитися з можливостями Middleware для створення конвеєра обробки запиту у застосунках ASP.NET Core. Навчитися налаштовувати порожній застосунок ASP.NET Core, додаючи компоненти, які необхідні для роботи MVC.

Вивчивши матеріал цього заняття, учень зможе:

- Додавати MVC до порожнього проєкту ASP.NET Core.
- Виконувати основні налаштування маршрутизації.
- Створювати контролери та представлення, використовувати основні можливості методів дій.
- Створювати користувацькі представлення.

Зміст уроку

1. Що таке MVC.
2. Що таке Middleware.
3. Як налаштувати застосунок ASP.NET Core для використання шаблону MVC.
4. Робота зі статичними файлами.
5. Приклад простого застосування з моделлю та кількома представленнями.

Резюме

- **MVC (Model View Controller)** – один із шаблонів для побудови застосунків з користувацьким інтерфейсом. Зараз поширений у технологіях, які пов'язані з побудовою веборієнтованих застосунків.
- **Основне завдання шаблону MVC** – відокремити користувацький інтерфейс (view) від бізнес-логіки (controller) і водночас винести окремо бізнес-сутності (model).
  Такий підхід дає велику гнучкість, надаючи можливість змінювати інтерфейс, не торкаючись поведінки та даних, і змінювати дані та поведінки, не вносячи зміни до інтерфейсу.
- **Model** – клас, який зберігає стан застосунку. Містить дані, які потрібно зобразити.
- **Controller** – клас, який відповідальний за виконання бізнес-логіки, оновлення моделі та вибір правильного представлення для показу даних користувача.
- **View** – HTML-розмітка, у якій зображується стан моделі.
- Якщо дотримуватися правил MVC, то виходить код, який тестується, оскільки написати Unit-тести на класи-контролери, які не пов'язані з інтерфейсом, набагато простіше, ніж на класи, які тісно пов'язані з HTML-розміткою та елементами управління.
- **Для використання MVC в застосунку ASP.NET Core потрібно додати два фрагменти коду:**
  - services.AddMvc() у класі Startup у методі ConfigureServices;
  - app.UseEndpoints(endpoints => endpoints.MapControllers()) у методі Configure.
- Під час іменування класу контролера потрібно використовувати закінчення **Controller**,

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

наприклад, Home**Controller**, Account**Controller**, Catalog**Controller**.

- Клас контролера має успадковуватись від базового класу Controller або ControllerBase.
- Контролери мають бути розташовані у директорії Controllers (але це є необов'язковим правилом).
- **Відкриті методи у контролері** називаються методами дії, або **Action Method**. Такі методи можна запустити за допомогою правильно надісланого запиту HTTP.
- Зазвичай методи дії повертають тип даних **IActionResult**, який може представляти розмітку, перенаправлення користувача або відповідь з певним статус-кодом.
- Під час використання у методі дії методу **View** без передання параметрів користувачу у відповідь буде повертатися представлення, яке називається так само як і поточний метод дії. Водночас подання буде розташовуватися у теці Views, у вкладеній теці, яка називається так само як і контролер, в якому було запущено метод дії.
- Для того, щоб застосунок міг обслуговувати запити до статичних файлів (наприклад, до HTML-сторінки або файлу зі стилями), необхідно розмістити файл у директорії **wwwroot** і додати в middleware pipeline app.UseStaticFiles. Водночас цей middleware має бути доданий перед middleware MVC.
- **Middleware** – блок програмного коду, який є частиною конвеєра та виконує обробку HTTP-запиту та відповіді.
- **Middleware може**: визначити, чи потрібно передавати запит наступному middleware в конвеєрі, виконати роботу до і після наступного middleware, який знаходиться в конвеєрі.
- **Методи визначення Middleware в коді конфігурації middleware pipeline:**
    - **Run** – найпростіший спосіб додати до конвеєра middleware. Middleware, який доданий через метод Run, не передає запит далі по конвеєру.
    - **Map** – застосовується для зіставлення шляху запиту з певним конвеєром, що складається з middleware.
    - **Use** – додає middleware до конвеєра обробки запиту, даючи можливість будувати ланцюжок middleware, який буде обробляти запит послідовно.

- **Для створення власного Middleware необхідно використовувати один із 2 підходів:**
    - Middleware, які ґрунтуються на угоді:
        - створити клас для Middleware;
        - визначити конструктор із параметром типу RequestDelegate;
        - визначити метод із сигнатурою public async Task Invoke(HttpContext context) або public async Task InvokeAsync(HttpContext context).
    - Middleware, який ґрунтується на фабриці middleware:
        - створити клас для Middleware;
        - реалізувати інтерфейс IMiddleware.

Закріплення матеріалу

- Навіщо потрібний шаблон MVC? Поясніть, що означає M, V, C.
- Що таке Middleware?
- Як додати власний Middleware до конвеєра обробки запиту у застосунку ASP.NET Core?
- Як додати MVC у застосунок ASP.NET Core?
- Яких правил потрібно дотримуватися під час створення контролера?
- Що таке Action Method?
- Навіщо потрібний метод View в контролері?
- Звідки береться метод View у контролері?
- Що таке статичні файли та як з ними працювати в застосунку ASP.NET Core?

Додаткове завдання

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Створіть порожній застосунок ASP.NET Core. Внесіть до нього потрібні зміни для використання MVC. Зробіть необхідні зміни у проєкті, щоб під час запиту /Test/Message зображувалася сторінка з повідомленням «Hello world», а під час запиту List/Info зображувався список <ul> з трьома елементами та довільним текстом.

Самостійна діяльність учня

Завдання 1

Доопрацюйте застосунок SimpleApp. До файлу data.txt додайте додаткову інформацію про продукт – опис продукту, кількість одиниць на складі.

Додайте в представлення Details опис продукту та кількість одиниць на складі.

У представленні List зробіть так, щоб якщо продукту на складі немає, зображувалося повідомлення навпроти продукту «Немає в наявності». А якщо кількість до 5 одиниць на складі – «Закінчується», якщо понад 5 одиниць – «У наявності».

Завдання 2

Зробіть так, щоб у застосунку SimpleApp за адресою Home/Index поверталося представлення. Зробіть у цьому представленні посилання «Завантажити опис уроку». Під час натискання на посилання цей файл має завантажуватися.

Рекомендовані ресурси

Загальні відомості про ASP.NET Core MVC

https://docs.microsoft.com/ru-ru/aspnet/core/mvc/overview?view=aspnetcore-2.1

Довідник із синтаксису Razor

https://docs.microsoft.com/ru-ru/aspnet/core/mvc/views/razor?view=aspnetcore-2.1

Model View Controller

https://ru.wikipedia.org/wiki/Model-View-Controller

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com