ASP.NET CORE ASOSLARI

Yusupov Mirsaidbek Abdulaziz o'g'li
Farg'ona davlat universiteti amaliy matematika va informatika kafedrasi o'qituvchisi
mirsaidbeky@gmail.com
Abdulhafizov Ibrohim Husanjon o'g'li
Farg'ona davlat universiteti 2-kurs talabasi
ibrohimabdulhafizov33@gmail.com

Annotatsiya: *Ushbu maqolada ASP.NET Core platformasining asosiy tushunchalari, arxitekturaviy yechimlari, ishlash mexanizmlari va real hayotdagi qo'llanilishi yoritilgan. ASP.NET Core yordamida cross-platformali (ko'p operatsion tizimda ishlovchi) yuqori samarali web ilovalar yaratish mumkin. Shuningdek, maqolada middleware, routing, dependency injection (DI), MVC va Web API konsepsiyalari texnik tafsilotlari bilan bayon etilgan. Ushbu texnologiya bugungi IT sanoatida katta ahamiyatga ega.*

Kalit so'zlar: *ASP.NET Core, cross-platforma, web API, MVC, .NET, Routing tizimi, Middleware, Dependency Injection (DI), Xavfsizlik.*

Аннотация: *В этой статье рассматриваются основные концепции, архитектурные решения, рабочие механизмы и реальные приложения платформы ASP.NET Core. С помощью ASP.NET Core можно создавать кроссплатформенные (работающие на нескольких операционных системах) высокопроизводительные веб-приложения. Кроме того, в статье подробно описаны концепции промежуточного программного обеспечения, маршрутизации, внедрения зависимостей (DI), MVC и веб-API. Эта технология имеет большое значение в современной ИТ-индустрии.*

Ключевые слова: *ASP.NET Core, кроссплатформенность, веб-API, MVC, .NET, система маршрутизации, промежуточное ПО, внедрение зависимостей (DI), безопасность.*

Annotation: *This article covers the basic concepts, architectural solutions, working mechanisms and real-life applications of the ASP.NET Core platform. With the help of ASP.NET Core, it is possible to create cross-platform (working on multiple operating systems) high-performance web applications. Also, the article describes middleware, routing, dependency injection (DI), MVC and Web API concepts in technical details. This technology is of great importance in today's IT industry.*

Keywords: *ASP.NET Core, cross-platform, web API, MVC, .NET, Routing system, Middleware, Dependency Injection (DI), Security*.

Kirish

Internet texnologiyalari taraqqiyoti web dasturlarni yaratish texnologiyalarining ham tez rivojlanishiga olib keldi. Microsoft tomonidan ishlab chiqilgan ASP.NET Core — bu zamonaviy, moslashuvchan, kross-platformali framework bo‘lib, u orqali professional darajadagi web ilovalarni ishlab chiqish mumkin. U .NET Core asosida ishlaydi va ochiq manbali hisoblanadi. ASP.NET Core dasturchilarga RESTful API, dinamik web sahifalar, real vaqtli aloqa (SignalR), Blazor yordamida SPA (Single Page Application) tuzish imkonini beradi.

Asosiy qism

1. ASP.NET Core tarixi va afzalliklari

ASP.NET Core dastlab 2016-yilda taqdim etilgan bo‘lib, klassik ASP.NET framework’iga nisbatan ko‘plab yangilanishlarga ega:

- Ochiq manbali (open-source) — GitHub’da mavjud.
- Kross-platformali — Windows, Linux va macOS’da ishlaydi.
- Engil va tezkor — Kichik resurslarda ham samarali ishlaydi.
- Modular — faqat kerakli komponentlar yuklanadi.
- Cloud-friendly — bulutli xizmatlar (Azure, AWS) bilan yaxshi integratsiyalashadi.

2. Dastur tuzilmasi va asosiy fayllar

ASP.NET Core loyihasi quyidagi asosiy fayllardan tashkil topgan:

- **Program.cs** – Dastur boshlanish nuqtasi. WebApplication.CreateBuilder() orqali konfiguratsiya qilinadi.
- **Startup.cs** (yoki zamonaviy versiyalarda Program.csning ichida yoziladi) – xizmatlar va middleware’lar ro‘yxatga olinadi.
- **appsettings.json** – Konfiguratsion ma’lumotlar: ulanish satrlari, log darajalari va h.k.
- **Controllers/, Models/, Views/** – MVC (Model-View-Controller) tuzilmasi asosidagi kataloglar.

3. Middleware tushunchasi

Middleware — bu HTTP so‘rovlari va javoblarini qayta ishlovchi komponentlar zanjiri. Misollar:

- UseRouting() – so‘rovlarni marshrutlash.
- UseAuthentication() – autentifikatsiyani tekshirish.
- UseAuthorization() – foydalanuvchining huquqlarini tekshirish.
- UseEndpoints() – yakuniy manzillarga yo‘naltirish.

Middleware’lar Program.cs faylida navbat bilan yoziladi. Tartib muhim!

4. Dependency Injection (DI)

DI — xizmatlar va komponentlar o‘rtasidagi bog‘liqlikni boshqarish tizimi. ASP.NET Core ichki DI mexanizmini qo‘llab-quvvatlaydi. Masalan:

```
services.AddScoped<IEmailService, EmailService>();
```

Ushbu xizmatni controller ichida quyidagicha chaqirish mumkin:

```
public class HomeController : Controller
{
   private readonly IEmailService _emailService;

   public HomeController(IEmailService emailService)
   {
      _emailService = emailService;
   }
}
```

5. Routing tizimi

Routing — foydalanuvchi kiritgan URL'ni kerakli controller va action’ga yo‘naltirish tizimi. ASP.NET Core’da ikki turdagi routing mavjud:

- **Conventional routing** — umumiy yo‘l qoidalari asosida.
- **Attribute routing** — controller va action ustida [Route] atributlari bilan belgilangan.

```
[Route("api/[controller]")]
public class ProductsController : ControllerBase
```

```
{
    [HttpGet("{id}")]
    public IActionResult GetProduct(int id) { ... }
}
```

6. MVC va Web API

ASP.NET Core MVC arxitekturasi:

1. **Model** – ma’lumotlar sinfi, validation va biznes qoidalari.
2. **View** – HTML va Razor sintaksisida yozilgan ko‘rinishlar.
3. **Controller** – foydalanuvchi so‘rovini qabul qiluvchi va qayta ishlovchi sinf.

Web API esa front-end ilovalar yoki boshqa xizmatlar bilan muloqot uchun mo‘ljallangan. JSON formatdagi ma’lumotlarni yuboradi/qabul qiladi.

7. Blazor texnologiyasi

Blazor yordamida siz C# tilida browser ichida ishlaydigan ilovalarni yaratishingiz mumkin. Bu texnologiya orqali JavaScript o‘rniga C# ishlatiladi:

1. **Blazor Server** – server orqali real vaqtda SignalR bilan ishlaydi.
2. **Blazor WebAssembly** – browser ichida to‘liq mustaqil ishlaydi.

8. Xavfsizlik: Autentifikatsiya (Authentication) va Avtorizatsiya (Authorization)

ASP.NET Core xavfsizlik uchun quyidagi imkoniyatlarni beradi:

1. **Identity** – foydalanuvchi ro‘yxatga olish, login, parolni tiklash.
2. **JWT (JSON Web Token)** – API’lar bilan xavfsiz aloqalar uchun.
3. **Role-based access** – foydalanuvchining roliga qarab kirishni cheklash:

```
[Authorize(Roles = "Admin")]
public IActionResult AdminPanel() { ... }
```

9. SignalR – real vaqtli aloqa

SignalR yordamida chat ilovalari, onlayn o‘yinlar yoki monitoring tizimlari yaratish mumkin. U WebSocket, Server-Sent Events yoki Long Polling orqali ishlaydi:

```
services.AddSignalR();
```

10. ASP.NET Core va bulut texnologiyalari

ASP.NET Core to‘liq **Azure**, **AWS**, **Google Cloud** xizmatlariga integratsiyalashgan. Bu esa ilovalaringizni osongina global miqyosga olib chiqish imkonini beradi.

Xulosa

ASP.NET Core — bu zamonaviy dasturchilar uchun kuchli, kengaytiriladigan va moslashuvchan web platformadir. Uning modullar asosidagi yondashuvi, kross-platformaliligi, xavfsizlik choralari, SignalR va Blazor kabi texnologiyalari orqali web ilovalarni ishlab chiqish ancha qulaylashgan. Ushbu platformani o‘rganish orqali dasturchilar bugungi kunning talablari asosida innovatsion loyihalarni yaratish imkoniga ega bo‘ladilar.

Foydalanilgan adabiyotlar:

1. Microsoft Docs – https://docs.microsoft.com/aspnet/core
2. Freeman A. *Pro ASP.NET Core MVC*, Apress, 2020.
3. TutorialsTeacher – https://www.tutorialsteacher.com/core

4. Udemy: *ASP.NET Core Web API Development*
5. Microsoft Learn – https://learn.microsoft.com
6. YouTube: “ASP.NET Core Full Course – 2024”
7. GitHub – ASP.NET Core Open Source Repositories
8. w3schools.com