

ASP.NET Core i Entity Framework

G04Code

Marko Cvijanović
Mihailo Bubnjević

.NET

- Open source
- Cross-platform
- Primenjuje se na dosta različitih tipova aplikacija
- Podržava jezike C#, F# i VisualBasic



Web

Build web apps and services for Windows, Linux, macOS, and Docker.



Mobile

Use a single codebase to build native mobile apps for iOS, Android, and more.



Desktop

Create native apps for Windows and macOS or build apps that run anywhere with web technologies.



Microservices

Create independently deployable microservices that run on Docker containers.



Cloud

Consume existing cloud services, or create and deploy your own.



Machine learning

Add vision algorithms, speech processing, predictive models, and more to your apps.



Game development

Develop 2D and 3D games for the most popular desktops, phones, and consoles.



Internet of Things

Make IoT apps, with native support for the Raspberry Pi and other single-board computers.

Razlicite verzije platforme .NET-a

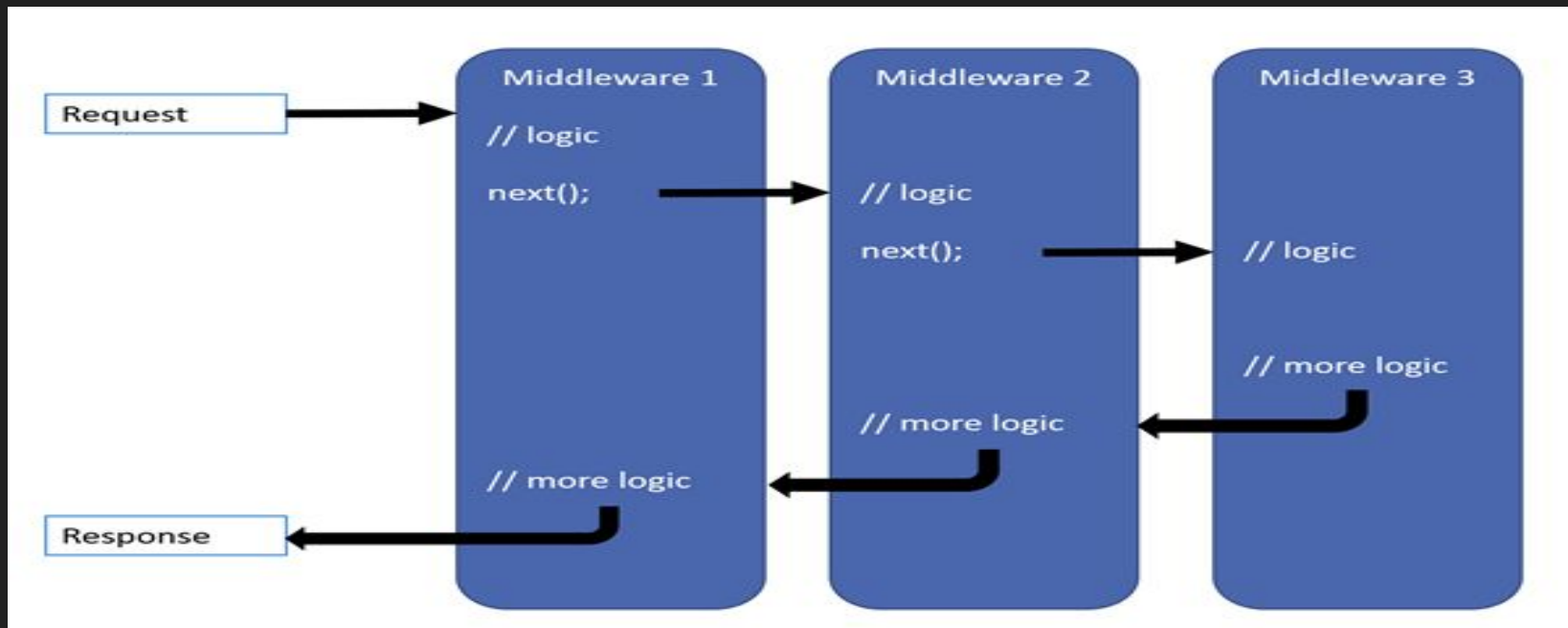
- .NET Framework - originalna verzija platforme koja je podržavala samo Windows operativni sistem
- .NET Core - moderna verzija platforme
- .NET Standard - specifikacija seta API-ja koju svaka implementacija .NET-a mora da definise

Program.cs

- Ulazna tačka projekta
- Pokreće server
- Radi dependancy injection registrovanih servisa
- Konfiguriše middleware (pipeline zahteva)

Middleware (pipeline zahteva)

- Izvršavaju se redom kojim su pisani u Program.cs



Dependency Injection

- Koristi se kako bi postigli slabije veze između objekata
- Ugrađen u .NET
- Postoje built-in mehanizmi za kontrolu životnog veka svih servisa za koje je definisan dependency injection kroz WebApplicationBuilder
- AddScoped
- AddTransient
- AddSingleton

Entity Framework

- ORM(Object Relational Mapper)
- Olakšava rad sa bazom
- Open source
- LINQ Podrška
- [EF Documentacija](#)

Entity Framework

- Potrebni alati i paketi za rad sa Sql server bazom:
 - [SQL Server](#)
 - [SQL Server Management Studio](#)
 - Microsoft.EntityFrameworkCore
 - Microsoft.EntityFrameworkCore.SqlServer
 - Microsoft.EntityFrameworkCore.Tools

Entity Framework

- **DbContext**

- Predstavlja sesiju sa bazom
- Koristi se da cuvamo i dobavljamo podatke iz baze
- Nas DbContext mora naslediti DbContext klasu
- Mora sadržati konstruktor
- Definišemo tabele kroz property tipa DbSet

Entity Framework

- DbContext primer

```
namespace ExampleDb.Context
{
    2 references
    public class ExampleDbContext : DbContext
    {
        0 references
        public DbSet<Person> People { get; set; }

        0 references
        public ExampleDbContext(DbContextOptions options) : base(options)
        {
        }
    }
}
```

```
builder.Services.AddDbContext<UniversityContext>(options =>
{
    options.UseSqlServer("Server=localhost;Database=UniversityDb;Trusted_Connection=True;TrustServerCertificate=True");
});
```

Entity Framework

- **Model**

- Modeli koji su property u DbContextu se smatraju entitetima
- Svaki model mora imati primary key (Id ili <naziv>Id)
- Konfiguracija putem Anotacija ili u Configure metodi
- Navigation property(many to one, one to many, one to one, many to many)

Entity Framework

- **Migracije**
 - Migracije izvršavamo kada želimo da promenimo entitete
 - Tools -> Nuget Package Manager -> Package Manager Console
 - add-migration - dodavanje migracije

Entity Framework

- **Korisne anotacije**
 - `[Table("imeTabele")]` - da damo naziv tabeli
 - `[Column("imeKolone")]` - za ime kolone u tabeli
 - `[MaxLength(500)]` - za ograničenje vrednosti
 - `[Required]` - za označavanje obaveznih polja
 - `[Column(Order = 0)]` - za redosled kolona u tabeli

Entity Framework

- IQueryable
 - Predvidjen za vršenje upita
 - Nasledjuje IEnumerable
 - Pri slanju upita na bazu šalje select upit sa svim filterima
 - LINQ upiti se putem njega izvršavaju direktno na bazi