



**Code  
Academy**



**Dėstytojas**









**Vilmantas**

# **Prijungiame duomenų bazę**

**Data**



# Nuget'ai kurių prireiks

 	<b>Microsoft.EntityFrameworkCore</b> by Microsoft Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations. EF Core works with SQL Server, Azure SQL Database, SQLite, Azure Cosmos DB, MySQL, PostgreSQL, and other databases through a provider plugin API.	6.0.3
		6.0.5
 	<b>Microsoft.EntityFrameworkCore.SqlServer</b> by Microsoft Microsoft SQL Server database provider for Entity Framework Core.	6.0.3
		6.0.5
 	<b>Microsoft.EntityFrameworkCore.Tools</b> by Microsoft Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.	6.0.3 
		6.0.5 



# Susikuriame Context'o klasę

```
public class ApplicationDbContext : DbContext
{
    public DbSet<Account> Accounts { get; set; }

    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
    {
    }
}
```



# Priedame ConnectionString i appsettings.development.json

```
"ConnectionStrings": {  
  "Database": "Server=localhost;Database=Test;Trusted_Connection=True;"  
},  
"AllowedHosts": "*"}
```



## Užregistruojame ApplicationDbContext klasę kaip DbContext naudodami connection string iš appsettings

```
// This method gets called by the runtime. Use this method to add services to the container.
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllers();
    services.AddSwaggerGen(c =>
    {
        c.SwaggerDoc("v1", new OpenApiInfo { Title = "DefaultTemplate", Version = "v1" });
    });
    services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(Configuration.GetConnectionString("Database")));
}
```

Nepamirštame migracijų!



# Nepamirštame enkapsuliuoti konteksto repozitorijoje

```
public class AccountRepository : IAccountRepository
{
    private readonly ApplicationDbContext _context;

    public AccountRepository(ApplicationDbContext context)
    {
        _context = context;
    }
}
```



# Užregistruojame repozitorijos servisą.

Jeigu dvejojate kokį tipą naudoti serviso registravimui naudokite Scoped.

Transient/Singleton pasirinkimas turi būti gerai argumentuotas.

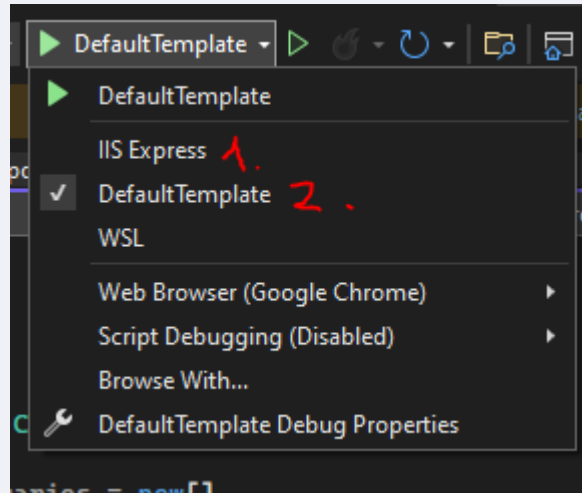
```
services.AddScoped<IAccountRepository, AccountRepository>();
```



# Loginimas

Paprščiausias būdas matyti ILogger serviso log'us konsolėje yra pasileisti programą ne kaip IIS Express, bet kaip aplikaciją.

DefaultTemplate yra mano projekto pavadinimas



Galima susikonguruti Loggerį su IIS Express: <https://stackoverflow.com/questions/40148660/how-to-get-a-console-output-in-asp-net-core-with-iis-express>



Asp .NET core Entity framework

# Loginimas

Susikuriame ILogger, Injectiname jį ir naudojame

```
[ApiController]
[Route("[controller]")]
public class WeatherForecastController : ControllerBase
{
    private static readonly string[] Summaries = new[]
    {
        "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering", "Scorching"
    };

    private readonly ILogger<WeatherForecastController> _logger;

    public WeatherForecastController(ILogger<WeatherForecastController> logger)
    {
        _logger = logger;
    }

    [HttpGet]
    public void Get()
    {
        _logger.LogInformation("Information message");
        _logger.LogWarning("Warning message");
        _logger.LogError("Error message");
    }
}
```

```
C:\Users\ITWORK\Documents\CodeAcademy\DefaultTemplate\DefaultTemplate\bin\Debug\net5.0\DefaultTemplate.exe
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\ITWORK\Documents\CodeAcademy\DefaultTemplate\DefaultTemplate
info: DefaultTemplate.Controllers.WeatherForecastController[0]
      Information message
warn: DefaultTemplate.Controllers.WeatherForecastController[0]
      Warning message
fail: DefaultTemplate.Controllers.WeatherForecastController[0]
      Error message
```



## Užduotis nr. 1

- Prijungiame duomenų bazę prie programos, kurią parašėte vakar.
- Pamėginkite naudoti ILogger, spausdinti informaciją, kuri pagal jus yra svarbi.