
Entity Framework Core

Introduction to [Entity Framework Core](#)

What is EF?

- Object-relational mapper (O/RM)
 - Supports many different DB providers ([list](#)) examples:
 - MS SQL Server
 - SQLite
 - PostgreSQL
 - MySQL
 - In-Memory (for testing)
 - NuGet (example): [Microsoft.EntityFrameworkCore.SqlServer](#)
-

Getting Started

- Follow [installing docs](#) to add EF to project
 - Make sure to install global EF tools: `dotnet tool install --global dotnet-ef`
 - Work through [tutorials](#) in EF docs
 - [Tutorial for ASP.NET Core](#)
 - Tips:
 - Run local [SQL Server using Docker](#): `docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=yourStrong(!)Password' -e 'MSSQL_PID=Express' -p 1433:1433 -d mcr.microsoft.com/mssql/server`
 - Install and use free [SQL Server Developer Edition](#) or [SQL Server Express LocalDB](#)
 - Use [In-Memory DB](#) for simple test scenarios
-

Building a Model

```
public class Blog
{
    public int BlogId { get; set; }
    [Required]
    public string Url { get; set; }
}
```

- Read more about [creating a Model](#)

Setting up the Context

```
public class BloggingContext : DbContext
{
    public BloggingContext(DbContextOptions<BloggingContext> options)
        : base(options)
    { }

    public DbSet<Blog> Blogs { get; set; }
}
```

Writing Data

```
using System.Threading.Tasks;

namespace EFIntro
{
    partial class Program
    {
        async static Task WriteToDB(AddressBookContext db)
        {
            db.Persons.AddRange(new [] {
                new Person() { FirstName = "Tom", LastName = "Turbo" },
                new Person() { FirstName = "Foo", LastName = "Bar" }
            });
            await db.SaveChangesAsync();
        }
    }
}
```

```
    }  
  }  
}
```

- Read more about [writing data](#)
-

Writing Data to Related Sets

```
using Microsoft.EntityFrameworkCore;  
using Newtonsoft.Json;  
using System;  
using System.Threading.Tasks;  
  
namespace Joins  
{  
    partial class Program  
    {  
        static async Task AddWithJoinAsync(OrderContext context)  
        {  
            var order = new Order { Product = "Bike",  
                                    Customer = new Customer { Name = "John" } };  
  
            // Note SINGLE call to `Add` and `SaveChangesAsync`  
            context.Orders.Add(order);  
            await context.SaveChangesAsync();  
        }  
    }  
}
```

- Note *single* call to Add and SaveChangesAsync
-

Querying Data

```
using Microsoft.EntityFrameworkCore;  
using System.Threading.Tasks;  
using System;  
using System.Linq;
```

```
namespace EFIntro
{
    partial class Program
    {
        async static Task ReadFromDB(AddressBookContext db)
        {
            await foreach(var person in db.Persons
                .Where(p => p.LastName.StartsWith("B"))
                .AsAsyncEnumerable())
            {
                Console.WriteLine($"{person.LastName},
↪ {person.FirstName}");
            }
        }
    }
}
```

- Read more about [querying data](#)

Querying Data

```
using Microsoft.EntityFrameworkCore;
using Newtonsoft.Json;
using System;
using System.Threading.Tasks;

namespace Joins
{
    partial class Program
    {
        static async Task QueryWithJoinAsync(OrderContext context)
        {
            var result = await context.Orders
                .Include("Customer")
                .FirstAsync();
            Console.WriteLine(JsonConvert.SerializeObject(result));
        }
    }
}
```

-
- Include: Getting Order with related Customer
-

EF Core + ASP.NET Core Cheat Sheet

[Entity Framework Cheat Sheet](#)

Further Readings and Exercises

- Readings
 - [Entity Framework Documentation](#)
 - [EF Tutorial with ASP.NET Core](#)
- Videos
 - Want to know more details? Watch [Entity Framework Core](#) on *Channel 9*