

This cheat sheet is for the course [Learn C# Full Stack Development with Angular and ASP.NET](#) by Jannick Leismann.

ENTITY FRAMEWORK CORE (DbContext)

The main class in Entity Framework Core (EF Core) that communicates with the database is called a DbContext. Serving as a link between your entity classes and the database, it is utilized for data queries and saves.

Components of DbContext

DbSet<TEntity> Properties:

These properties represent collections of entities in the context. Typically, there is one DbSet<TEntity> for each entity class in your model.

```
using EmployeeManagement.Models;
using Microsoft.EntityFrameworkCore;

namespace EmployeeManagement.Data
{
    5 references | jaysonsoi, 4 days ago | 1 author, 1 change
    public class AppDbContext : DbContext
    {
        0 references | jaysonsoi, 4 days ago | 1 author, 1 change
        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)
        { }
        5 references | jaysonsoi, 4 days ago | 1 author, 1 change
        public DbSet<Employee> Employees { get; set; }
    }
}
```

Configure Datasource

This method is used to configure the database (and other options) to be used for this context. Add this to your **Program.cs**

```
builder.Services.AddDbContext<AppDbContext>(
    options => options.UseSqlServer("YourConnectionString")
);
```

You have to make sure that you have installed the required NuGet package base on your datasource.



Usage:

```
using (var context = new AppDbContext())
{
    // Adding a new employee
    var employee = new Employee{ Name= "John Doe", Position = "Manager" };
    context.Employees.Add(employee );
    context.SaveChanges();

    // Querying products
    var employees= context.Employees.ToList();
}
```

In order to manage database connections, keep track of entity changes, and store data, an EF Core **DbContext** is necessary. It offers an extensive feature set for working with the database in an object-oriented, high-level manner.