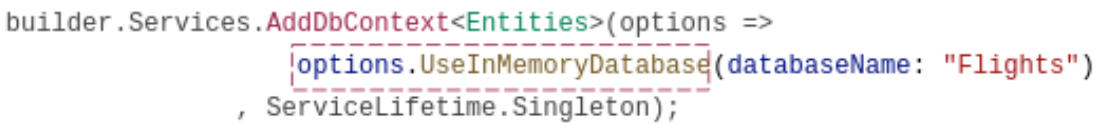
Day 16 - Angular & Asp in 21 days Handouts

Status

Day 16: Entity Framework in Memory Database



Entity Framework is an abstract interface that lets you connect to different database technologies by changing some configuration.

Rather than setting up and configuring connection to a SQL database upfront, you may want to start the development of your application with an in-memory database. And switch to a SQL Server database before the first release of your application.

To use Entity Framework, you should install the `Microsoft.EntityFrameworkCore.InMemory` package using dotnet CLI or Nuget.

```
dotnet add package Microsoft.EntityFrameworkCore.InMemory
```

Then derive your `Entities` class from the `Microsoft.EntityFrameworkCore.DbContext` class.

```
public class Entities : DbContext
```

You can then define your entity collections as `DbSet` s rather than `List` s.

```
public class Entities : DbContext
{
```

```
    public DbSet<Passenger> Passengers => Set<Passenger>();
    public DbSet<Flight> Flights => Set<Flight>();
}
```

You should also specify how the entities should be mapped to database. So you should override the `OnModelCreaing` method of the `DbContext` class for your `Entities` class.

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Customer>().HasKey(e => e.Email);
}
```

Finally you should configure the provider you want to work with and add it to the service collection.

```
builder.Services.AddDbContext<Entities>(options =>
                    options.UseSqlServer(
                        "Data Source=localhost,1433;" +
                        "Database=Flights;" +
                        "User id=SA;" +
                        "Password=1234!Secret;"
                    ));
```

You will learn how to configure a SQL Server provider later in this course.

For an in-memory provider, make sure to inject it as a singleton service.

```
builder.Services.AddDbContext<Entities>(options =>
                    options.UseInMemoryDatabase(databaseName:"Flights")
                    ,ServiceLifetime.Singleton);
```