قسمت 6

در این قسمت لایه Domain را به قسمت Core اضافه می‌کنیم. در این لایه Entity های خودمان را قرار می‌دهیم. موجودیت‌های که قرار است جدول انها در دیتابیس ایجاد شود.

جلسه 7

ساخت BaseDomainEntity که این کلاس abstract شامل تمام قسمت های فیلد های مشترک داخل همه کلاس های Domain است. بعد ساخت کلاس بقسه کلاس ها از این کلاس ارث بری می کنند.این کلاس را توی فولدر common قرار میدهیم.



قسمت 8

در این قسمت لایه core رو میسازیم. این لایه بین دیتابیس و برنامه ما و هر چیزی که می خواهد به دیتابیس وصل شود قرار میگیرد.برای این اتصال از الگوی Repository استفاده می کنیم.این اینترفیس ها اجازه می دهند که ما برنامه هایی که قرار هست با دیتا بیس ارتباط برقرار کنند را کنترل کنیم. به عنوان یک گلوگاه این کار را انجام بدیم. در این جلس IGenericRepository را میسازیم.



قسمت9

دراین جلسه IGenericRepository را می‌سازیم و فیلدهای اه را پر می کنیم.



بعد از این کار برای هرکدام از کلاس های Domain یک ریپوزیتوری خاص می‌سازیم. که از کلاس IGenric ارث بری کرده است. و وردی ان خود کلاس مربوط به ان است.



قسمت 10

AutoMapper

در این قسمت با نصب این پکیج برای تبدیل داده ها از Dto و entity ها استفاده می کنیم



کلاس را هم در این قسمت ایجاد می کنیم. از profile که مربوط به Automapper هست ارث بری می کنیم.



فولدر DTOs (Data Transfer Object)که مربوط به مدل ویو ها هست را هم در لایه Application ایجاد می کنیم.برای ارتباط بین Entity ها و برنامه ها ما از مدلهای DTO استفاده می کنیم.Automapper مدل های که در DTOs ساختیم را به Entity های Domain مپ می کند.

درس 11

در این قسمت برای هر کدام از entity های Domain یک DTO میساریم. بیس و همه موجودیت ها. هدف از این بخش این است که فقط خود موجودیت ها با دیتا بیس ارتباط داشته باشندو برای ازتباط بیقه برنامه ها با موجودیت ها باید از DTO استفاده کنند.



بعد از این بخش Automapper رو هم کانفیگ می کنیم.



همانطور که داخل کدها مشخص هست از داخل constractor از createmap استفاده می کنیم و در انتها با ReverseMap این عمل را معکوس می کنیم که هم درجهت و معکوس کار کند.

درس 12

ساخت DTO های اختصاصی. به طور مثال برای یک ویو خاص نیاز به همه فیلد های موجودیت نیست. در این موارد به صورت اختصاصی برای ان ویو مدل dto میسازیم.



بعد ایجاد این DTO در قسمت َAutomapinng این قسمت رو اضافه می کنمیم تا در صورت نیاز استفاده کنیم.



در ادامه برای اینکه وابستگی ها مدیریت بشوند و پروژه مقصد برای وابستگی های سایر لایه ها سنگین نشود ، در هر لایه وابستگی ها و سرویس ها را در یک کلاس قرار میدهیم و در پروژه مقصد فقط اون رو فراخوانی می کنیم.

ساخت اکستنشن متد برای کانفیگورلایه Application برای این کار کلاس زیر را میسازیم. کلاس و متد باید staic باشند و و با کلمه کلیدی this این کلاس را به صورت اکستنشن متد می کنیم. حالا سرویس Automapper را تنظیم می کنیم. در کد زیر خط کامت شده فقط کلاس Mapping profile را به سرویس میشناساند ولی خط بعدی که با Assembly نوشته شده همه مواردی که با Profile هستند را انتخاب می کند. این برای زمانی هستکه قصد داریم برای Dto ها کلاس Automapper جدا بسازیم.

