

Ursprung och Resultat

I den första versionen skapade jag två Mock klasser för att lägga till och hämta information från. Men i den här versionen skapade jag en **DbContext** istället. Där jag har lagt till all information. Jag har raderat **Mock** klasserna för att jag inte behövde dem längre. Jag har lagt till en koppling mellan Painting klass och Artist genom ArtistId. Jag gjorde inte det i den första versionen för att jag inte visste om jag kommer att bygga upp en annan klass som kategori eller inte. Men jag tänkte att varje artist har egna tavlor, vilket ersätter självklart kategorin.

Vi har också möjlighet att bygga en funktion där vi sorterar tavlorna beroende på tavlans typ, i klassen Painting finns redan en variabel PaintingType. Jag har raderat funktionen **GetPaint** från **IPainting** klass och skapade istället en **Action** i kontrollen **Painting** där kan jag hämta och visa tavlans bild med full storleken med hjälp av en ny vy. Jag skapade även en ny View där man kan sortera och visa tavlorna för varje artist. I Artist kontrollen finns två nya Action en för att visa **Details** för varje Artist och en annan för att visa alla artister och två knappar en för att visa detalj för varje artist och tavlor. Man kan nå detaljer för varje artist genom att trycka på artistens namn när man visar tavlorna.

I den första versionen använde jag Actionlink i **index** men den här gången använde samma sätt som finns i materialet. Jag vet faktiskt inte om det finns en skillnad. Jag använde den första för att jag gjorde så tidigare men inte i core. Jag har använt **UseInMemory**. Det finns en fil för Migrationen i projektet men den här filen är bara för mig och påverkar inte på något sätt projektet.

Allting var bra och jag kände ingen svårighet när jag jobbade, kanske lite med Linq lambda.

Enhetstester

Jag har skapat två tester. En test för att kontrollera om vi har information om **Artister** och **painting** i **AppDbContext** eller inte. Det här testet är jätteviktigt tror jag för att vi kan inte jobba vidare och göra något om det finns något fel eller brist i information som finns Db. Det kan också påverka våra lösningar och hämta fel resultat. Jag skapade två listor en för Artister och en för Tavlorna. Jag loppar på elemnets som finns i original listor och lägga till de i listorna som jag har skapat. Med hjälp av **Count**, resultat skulle vara det samma.

En annan test som var ett krav var att testa om jag kan filtrera tavlorna. Jag skapar två listor en för Artist **Sonja** och en för Artist **Marco**. Jag loappar och letar efter **Id** och **firstName**. Om till exempel **id** var ett och **firstName** **Sonja**. Jag lägger tavlorna till **Sonjas** list. Om **Id** var **2** och **firstName** **Marco** går de till **Marco** list. Jag räknar hur många tavlor det finns i varje lista och jämför resultat med original listan med hjälp av **Where** och **Count** funktioner som finns i Linq, de skulle bli det samma.