

Ursprung

I den första versionen skapade jag två **Mock** klasser för att lägga till och hämta information från. Jag skapade ingen koppling mellan **Artister** och **Paintings** klasser. Eftersom jag inte visste vad kraven är i den andra versionen eller att jag behöver bygga en annan klass för kategorin. Men i den andra versionen tänkte jag att varje artist har egna tavlor, vilket ersätter självklart kategorin.

Jag skapade tre **Controller** Painting, Artister och även HomeController för att skapa **index** och visa tydligt information om innehållet. Jag skapade en **ViewModel** där man kan skapa en lista av tavlor med artistnamn. I det här steget har jag bara en artist och namnet var hårdkodad i Paintings Controller. Allt var gott och jag var väldigt nöjd med att uppdatera projektet och jag kände inte att det finns några svårigheter faktiskt, kanske för att det var min kod och jag planerade väl.

Resultat

I den här versionen skapade jag en **AppDbContext** klass där jag har lagt till all information istället för **Mock** klasserna. 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 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, det vill säga att det är typ av **Details** om man vill göra det. Jag skapade även en ny View där man kan sortera och visa tavlorna för varje artist.

I början tänkte jag koppla Artister med sina egna tavlor för att uppfylla kraven, jag gjorde det men ändå skapade jag två ytterligare **ActionResult** **GetAllType** och **GetPaintingByType**. **GetAllType** samlar alla typer av tavlor som finns i listan och därefter returnerar bara typer utan upprepning. Medan **GetPaintingByType** tar en parameter av string (**PaintingType**) som kommer från GetAllType och samlar alla tavlor som har samma type. Det vill säga att nu har två olika sortering, en genom Artister och en genom typ. Jag tror att det kommer att bli mycket bättre att skapa en annan klass för typ eller kategori och gör någon koppling. Men jag ville träna mer och gör något som inte finns på materialet.

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 någon 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.

Enhetstester

TestingNumberOfAristsAndPaintings

Ett 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 elements som finns i original listor och lägger till de i listorna som jag har skapat. Med hjälp av **Count** skulle resultatet vara det samma.

TestPaintingsFilter

Ett annat test som var ett krav var att testa om jag kan filtrera tavlorna. Jag skapar två listor en för Artist **Sonja** och annan för Artist **Marco**. Jag loppar 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.

TestGetArtistDetailsById

Det här testet handlar om att visa om det finns möjlighet att visa en artists kontaktinformation. Jag skapar en list för att lägga till en artist information och i testet valde jag Sonja. Genom Sonjas id hämtar jag information och tar också hennes **firstNamn** som variable. Jag börjar med att kika om jag tog det rätta namnet genom att jämföra **firstNamn** med original listan och därefter kollar jag med hjälp av count om jag fick info för bara Sonja. Och därefter kollar jag om det finns något annat id och kör **NotEqual**. Alla tester i de tre stegen är godkända.