



## EF Code First

Vi skapar en lokal databas i vår miljö Visual Studio och genom att jobba code first så behöver vi inte modellera fram tabellerna och relationen mellan dem. Istället är det konfigurationen i C#-klasserna som är de viktiga att få till för att säkra upp kopplingen mellan klasserna. Dock jobbar vi väldigt enkelt, det finns mycket som skulle kunna göras annorlunda med stored procedure och hämtning från databasens tabeller med join osv. Det finns alltså mycket mer man kan göra med Entity Framework än vad vi gör i kursen.

Bilaga till denna uppgift är textfilen som innehåller testdatat ni ska använda er av.

### Modellens olika klasser

Vi har ju redan skapat en del klasser när vi började lägga till Models, nu ska vi justera dem något innan vi skapar databasen.

### Errand (Ärenden)

Typ	Namn	DataAnnotation
int	ErrandId	
String	RefNumber	
String	Place	Required
String	TypeOfCrime	Required
DateTime	DateOfObservation	Required, DataType(DataType.Date), DisplayFormat(DataFormatString="{0:yyyy-MM-dd}")
String	Observation	
String	InvestigatorInfo	
String	InvestigatorAction	
String	InformerName	Required
String	InformerPhone	Required, RegularExpression
String	StatusId	
String	DepartmentId	
String	EmployeeId	

**ErrandID** – ändrar vi nu från string till en int. Detta nummer kommer att skapas automatiskt i databasen och är inget vi behöver tänka på när vi sedan skapar ärenden från formuläret.

**RefNumber** – ärende id är något för databasen att hålla koll på, men Småstads kommun behöver ha ett referensnummer att jobba med och lämna ut till anmälare osv. De vill ha numret i formen *år-45-löpnummer* och det betyder att vi ska lägga in RefNumber hårdkodat när vi skapar ett ärende. Se nedan vid tabell Sequence för hur det ska gå till.



**Place** handlar om platsen där miljöbrottet observerades. **TypeOfCrime** är vilken typ av brott det handlar om. **DateOfObservation** är datumet när brottet uppdagades och **Observation** vad anmälaren sett (kan lämnas tomt). **InformerName** och **InformerPhone** är förstås anmälarens namn och telefon.

**InvestigatorInfo** och **InvestigatorAction** är knutet till handläggarens arbete, vilket sker i uppgift 4.

**StatusId** handlar förstås om vilket status ärendet har, även här måste vi hårdkoda när vi ska spara ner i databasen – ett nytt ärende har alltid **StatusId** = S\_A. **DepartmentId** är vilken avdelning som har fått ärendet till sig och **EmployeeId** handlar om vilken handläggare som har ansvar för ärendet.

---

De sista tre hämtar data från databasen. Alltså tre tabeller där vi lägger in data från start och sedan inte rör dem:

### ErrandStatus

Den här tabellen innehåller uppgifter om de olika statusar ett ärende kan ha. Och det är från den här tabellen data hämtas för att fylla de rullgardinslistor som visar status.

Typ	Namn	DataAnnotation
String	StatusId	Key
String	StatusName	

Innehållet i tabellen är:

StatusID	StatusName
S_A	Inrapporterad
S_B	Ingen åtgärd
S_C	Påbörjad
S_D	Klar

När ett ärende skapas så anses den vara inrapporterad och därför sätts status till *Inrapporterad* innan ärendet sparas i databasen. När en chef markerar ett ärende för Ingen Åtgärd får det ärendet status *Ingen åtgärd*, när en handläggare börjar sin utredning och spar ner information av något slag ska ärendet ändras till *Påbörjad* och när ett ärende är avslutat så markeras den med *Klar*.



### Department (Avdelning)

Typ	Namn
String	DepartmentId
String	DepartmentName

Denna tabell innehåller samtliga (nuvarande) avdelningar hos Småstads Kommun.

#### Innehållet i tabellen är:

<i>DepartmentID</i>	<i>DepartmentName</i>
D00	Småstads kommun
D01	Tekniska Avloppshanteringen
D02	Klimat och Energi
D03	Miljö och Hälsoskydd
D04	Natur och Skogsvård
D05	Renhållning och Avfall

**Observera** att Småstads Kommun inte ska användas när man fördelar ärenden. Det är endast samordnaren som tillhör den "avdelningen" och hen utreder inga ärenden.

### Employee (Anställda)

Typ	Namn
String	EmployeeId
String	EmployeeName
String	RoleTitle
String	DepartmentId

De anställda är fördelade på de olika avdelningarna och en del av dem är chefer, andra handläggare och så har vi en samordnare.



Innehållet i denna tabell är:

<i>EmployeeID</i>	<i>EmployeeName</i>	<i>RoleTitle</i>	<i>Department</i>
E001	Östen Ärling	Coordinator	Småstads Kommun
E100	Anna Åkerman	Manager	Tekniska Avloppshanteringen
E101	Fredrik Roos	Investigator	Tekniska Avloppshanteringen
E102	Gösta Qvist	Investigator	Tekniska Avloppshanteringen
E103	Hilda Persson	Investigator	Tekniska Avloppshanteringen
E200	Bengt Viik	Manager	Klimat och Energi
E201	Ivar Oscarsson	Investigator	Klimat och Energi
E202	Jenny Nordström	Investigator	Klimat och Energi
E203	Kurt Mild	Investigator	Klimat och Energi
E300	Cecilia Unosson	Manager	Miljö och Hälsoskydd
E301	Lena Larsson	Investigator	Miljö och Hälsoskydd
E302	Martin Kvist	Investigator	Miljö och Hälsoskydd
E303	Nina Jansson	Investigator	Miljö och Hälsoskydd
E400	David Trastlund	Manager	Natur och Skogsvård
E401	Oskar Ivarsson	Investigator	Natur och Skogsvård
E402	Petra Hermansdotter	Investigator	Natur och Skogsvård
E403	Rolf Gunnarsson	Investigator	Natur och Skogsvård
E500	Emma Svanberg	Manager	Renhållning och Avfall
E501	Susanne Fred	Investigator	Renhållning och Avfall
E502	Torsten Embjörn	Investigator	Renhållning och Avfall
E503	Ulla Davidsson	Investigator	Renhållning och Avfall

---

Två tabeller som vi inte kommer jobba med förrän i uppgift 4 är de som rör handläggarens arbete. När en handläggare jobbar med ett ärende så kan det betyda att det görs provtagningar (Sample) och det kanske tas bilder (Picture).

### Sample (Prover)

Typ	Namn	DataAnnotation
Int	SampleId	
String	SampleName	
Int	ErrandId	

**SampleID** är det numret som filen får (automatiskt av EF), **SampleName** är namnet på filen och **ErrandID** vilket ärende denna fil är kopplad till.



### Picture (Bilder)

Typ	Namn	DataAnnotation
Int	PictureId	
String	PictureName	
Int	ErrandId	

**PictureID** är det numret som filen får (automatiskt av EF), **PictureName** är namnet på filen och **ErrandID** vilket ärende denna fil är kopplad till.

---

### Hantering av referensnummer

Ett ärendes referensnummer har formatet *år-45-löpnummer*, ex. 2018-45-0024, detta för att få ett unikt nummer som man kan använda för att söka efter. Löpnumret hämtas från en tabell som heter Sequence, och den innehåller bara en rad med nuvarande värde:

#### Sequence

Typ	Namn	DataAnnotation
Int	Id	
Int	CurrentValue	

**ID** är det numret som ges av databasen automatisk, vilket bara kommer att vara 1.

**CurrentValue** håller i det löpnummer som ska användas för att skapa referensnumret till ärendet.

Vi kommer att starta CurrentValue med talet 200.

### Stegen för att skapa ett ärende med löpnummer och ändra i Sequence-tabellen är:

Kolla om ärende-objektet har ett ErrandID

Om inte så betyder det att vi kan lägga till ett ärende

Hämta CurrentValue från Sequence-tabellen (gör på liknande sätt som när du hämtar ett ärende, men här är alltid id=1 då det bara finns en post i tabellen)

Lägg till refNumber på ärende-objektet = "2018-45-" + CurrentValue

Lägg till ärendet till databasen (glöm inte status)

Öka CurrentValue med 1 och uppdatera Sequence-tabellen med det nya värdet

Spara ändringarna i databasen

Returnera ärendets refnummer till tack-sidan i applikationen.