



---

---

---

---

---

---

---

---

Översikt

- Skapa MVC Models.
- Arbeta med information.

.eerec

---

---

---

---

---

---

---

---

Lektion 1: Skapa MVC Models

- Introduktion Model.
- Specifikation av egenskap.
- Validering av indata.
- Skräddarsydd validering.
- Model binder.
- Funktion för Model binder.

.eerec

---

---

---

---

---

---

---

---

---

---

## Introduktion Model

- MVC Model är en samling av .NET Framework-klasser.
- När Model skapas, definierar du egenskaper och metoder som bäst passar de objekt som den skall beskriva.
- Dessa egenskaper beskrivs i programkod, så att MVC kan bestämma hur dessa skall renderas till webbsida och hur indata skall valideras.
- Model beskriver funktionella krav för webbapplikationen, oftast refererade till user stories.

.eerec

---

---

---

---

---

---

---

---

## Introduktion Model (forts.)

- Om Agile utveckling används, skapas oftast klassen med några få egenskaper, efter diskussioner med användare kommer klassen att revideras ett antal gånger.

.eerec

---

---

---

---

---

---

---

---

## Specifikation av egenskap

- Modelklass specificerar tre attribut för varje egenskap:
  - Namn.
  - Datatyp.
  - Nivå av access (get, set).
- Ytterligare metadata kan anges, exempelvis hur egenskapsnamn skall visas. Detta görs med [DisplayName].

### Exempel:

```
[DisplayName("Skapad den:")]  
Public CreatedDate { get: set: }
```

.eerec

---

---

---

---

---

---

---

---

---

---

## Validering av indata

- Required, Range, StringLength och RegularExpression används för att validera indata från användare.
- ErrorMessage kan användas för att specificera felmeddelande om användare skriver in något felaktigt:

```
[Required(ErrorMessage="Skriv in ett namn:")]  
public string Namn { get; set; }
```

.eeec

---

---

---

---

---

---

---

---

## Skräddarsydd validering

- Arkitekturen för MVC har blivit designad för utbyggbarhet.
- Data annotations användas för att validera, ibland kan du behöva att skriva en egen skräddarsydd validering.
- Klass skapas som ärver från System.ComponentModel.DataAnnotations.ValidationAttribute.
- Override måste göras för metoden IsValid, för att kunna köra dina egna tester.

.eeec

---

---

---

---

---

---

---

---

## Skräddarsydd validering (forts.)

```
    }  
    public class CheckValidYear : ValidationAttribute  
    {  
        public CheckValidYear()  
        {  
            ErrorMessage = "Den tidigaste operan är Daphne, 1598 av Corsi, Peri och Rinuccini";  
        }  
        public override bool IsValid(object value)  
        {  
            int year = (int)value;  
            if (year < 1598)  
            {  
                return false;  
            }  
            else  
            {  
                return true;  
            }  
        }  
    }  
}
```

.eeec

---

---

---

---

---

---

---

---

---

---

## Model binder

- Model binder är en komponent som skapar instans av en modelklass, beroende på information som skickas i begäran från webbläsare.
- ASP.NET MVC innehåller en default model binder, som kan användas i de flesta situationer. Skräddarsydd binder kan skapas.
- Model binder ser till att rätt information skickas till en metods parametrar.

.eerec

---

---

---

---

---

---

---

---

## Funktion för Model binder

- Begäran från webbläsare:  
<http://plommonmannen.se/produkt/display/45>.
- Tre saker identifieras:
  - Klass som intresserar användare: produkt.
  - Händelse som skall ske: display.
  - Specifik instans av modelklass: produktid 45.
- Begäran tas emot av controller action invoker. Denna i sin tur kommer att kalla på händelse Display i controller Produkt och skickar 45 som en parameter till händelse.

.eerec

---

---

---

---

---

---

---

---

## Övning Lägga till Model



.eerec

---

---

---

---

---

---

---

---

---

---

## Lektion 2: Arbeta med information

- Ansluta till databas.
- Anslutningssträng.
- Entity Framework.
- Arbetsflöde i Entity Framework.
- Lägga till Entity Framework Context.
- För att använda Entity Framework behövs.
- LINQ to Entities.

.eeec

---

---

---

---

---

---

---

---

## Ansluta till databas

- ADO.NET stödjer ett stort antal databaser, genom att använda olika providers.
- Databaser i molnet kan användas för webbapplikationer som ligger i molntjänst.
- För att ansluta MVC-applikation till databas:
  - Lägg till referens till System.Data.
  - Lägg till anslutningssträng i filen web.config.

.eeec

---

---

---

---

---

---

---

---

## Anslutningssträng

- Anslutningssträng definieras i tag <connectionString> under <configuration>, efter <configSections>.

```
<<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?linkid=21468 -->
    <section name="entityframework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=7b4132bf1d428d2f" />
  </configSections>
  <connectionStrings>
    <add name="OperasDB" connectionString="Data Source=(localdb);attachdbfilename=|DataDirectory|kaspnet-OperasDBSite-20170522110524.mdf;Initial Catalog=kaspnet-OperasDBSite-20170522110524;integrated security=true" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="WebPages/Version" value="3.0.0.0" />
  </appSettings>
</configuration>
```

.eeec

---

---

---

---

---

---

---

---

## Entity Framework

- Utvecklare skriver programkod som arbetar med klasser och objekt, i kontrast mot att databas lagrar information i tabeller, kolumner och rader.
- ADO.NET inkluderar Entity Framework , en teknologi som ger möjlighet till att läsa och skriva information till och från databas genom att använda klasser och objekt.
- Ramverket knyter tabeller och kolumner i databas till objekt och dess egenskaper.

.eeec

---

---

---

---

---

---

---

---

## Arbetsflöde i Entity Framework

- Entity Framework erbjuder följande arbetsflöde:
  - Database First, används om databas redan existerar.
  - Model First, modell skapas först i designverktyget, när programkod körs kommer databas att skapas.
  - Code First, .NET Framework klasser skapas tillsammans med klasser för modell, när programkod körs kommer databas att skapas.

.eeec

---

---

---

---

---

---

---

---

## Lägga till Entity Framework Context

- När du arbetar med arbetsflödet Code First, måste du försäkra dig om att ramverket skapar rätt databas och rätt tabell, för att lagra dina klasser.
- Skapa klass som ärver från klassen Entity Framework DbContext.
- Denna klass används av controller när du vill manipulera information i databasen.
- I denna klass, skapar du `DbSet<>` property.

.eeec

---

---

---

---

---

---

---

---

---

---

## Lägga till Entity Framework Context (forts.)

- Lägga till Entity Framework context till din model:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;

namespace OperasWebSites.Models
{
    public class OperasDB : DbContext
    {
        public DbSet<Opera> Operas { get; set; }
    }
}
```

- Entity Framework kommer att leta efter anslutningssträng med namnet OperasDB.
- Databas kommer att ha tabell med namnet Operas, i exemplet ovan.

.eeec

---

---

---

---

---

---

---

---

## För att använda Entity Framework behövs

- För att använda Entity Framework involverar du:
  - Användning av context i Controllers.

```
public class OperasController : Controller
{
    private OperasDB db = new OperasDB();

    // GET: Operas
    public ActionResult Index()
    {
        return View(db.Operas.ToList());
    }
}
```

- Användning av Initializers för att populärisera databasen.

```
public class OperasInitializer : DropCreateDatabaseAlways<OperasDB>
{
    protected override void Seed(OperasDB context)
    {
        base.Seed(context);
        var operas = new List<Opera>
        {
            new Model<>().Opera
            {
                Title = "Vind från Tureta",
                Year = 1799,
                Composer = "Holzner"
            }
        };
        operas.ForEach(o =>
        {
            context.Operas.Add(o);
            context.SaveChanges();
        });
    }
}
```

.eeec

---

---

---

---

---

---

---

---

## LINQ to Entities

- LINQ to Entities är en version av LINQ som fungerar med Entity Framework.
- Ger möjlighet att skriva queries för att lokalisera specifik information, använda join för att hämta information från flera tabeller, uppdatera informationen eller andra åtgärder på din information.

.eeec

---

---

---

---

---

---

---

---

---

---

## Övning Arbeta med Entity Framework



---

---

---

---

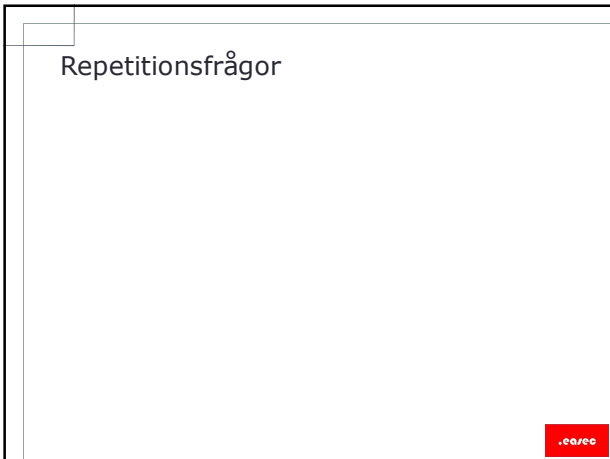
---

---

---

---

## Repetitionsfrågor



---

---

---

---

---

---

---

---

---

---