



Översikt

- MVC-applikation.
- Introduktion till MVC.

.eerec

Lektion 1: MVC-applikation

- MVC.
- Model.
- Views.
- Controller.
- Delade ASP.NET funktioner.

.eerec

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## MVC

- MVC är en programmeringsmodell tillgänglig i ASP.NET.
- Karaktärsdrag för MVC är en stark separation mellan företagslogik, tillgång till information och gränssnitt för användare.
- Applikation separeras i Models, Controllers och Views.

.eeec

---

---

---

---

---

---

---

---

## Model

- Beskriver site, genom skräddarsydda klasser och sparar programkod i .cs-filer.
- Exempel:
  - Site för ett bokförlag, presenterar information om böcker och författare.
  - Bok innehåller egenskaper såsom titel, summeringsinformation och antalet sidor.
  - Författare, innehåller egenskaper såsom namn och en kort biografi.
- Två klasser skapas om model används, en klass för bok och en klass för författare.

.eeec

---

---

---

---

---

---

---

---

## Model (forts.)

- Klasserna inkluderar de egenskaper som beskrivs tidigare och kan innehålla metoder för att köpa bok eller kontakta författare.
- Om information sparas i databas, kan model inkludera programkod som kan läsa eller förändra poster.
- Models är skräddarsydda .NET-klasser och sparar programkod i cs-filer.

.eeec

---

---

---

---

---

---

---

---

## Views

- Varje webbsite måste rendera HTML-sidor som visas i webbläsare.
- I vårt exempel kommer View hämta information från Model och rendera den.
- Views skapar användargränssnitt.
- Lagrar programkod i både HTML och C# i .cshtml-filer.

.eeec

---

---

---

---

---

---

---

---

## Controller

- Controller interagerar med användare, t ex vid klicka på knapp eller länk.
- Controllers svarar på användarehändelser, laddar information från model och skickar vidare till View.
- Controllers implementerar logik för indata och knyter Model till rätt View.
- Controllers är .NET-klasser som ärver från System.Web.Mvc.Controller och lagrar programkod i .cs-filer.

.eeec

---

---

---

---

---

---

---

---

## Delade ASP.NET funktioner

- Inkluderar ett antal funktioner oavsett programmeringsmodell.
- *Configuration*, web.config-fil, XML-baserad, innehåller konfiguration för applikationen.
- *Authentication*, ASP.NET har stöd för ett antal autentiseringsmetoder.
- *Membership och roles*, membership providers är komponenter som implementeras funktioner för användarehantering, SQL databas kan användas. Flera användare kan grupperas i roller.

.eeec

---

---

---

---

---

---

---

---

---

---

### Delade ASP.NET funktioner (forts.)

- *State Management*, webbservrar och webbläsare kommunicerar med HTTP. Är ett stateless protokoll där request separeras. Funktioner här hanterar mellanlagring av information.
- *Caching*, ASP.NET byggs dynamiskt av ASP.NET runtime på webserver. Hastighet kan höjas om mellanlagring används.



---

---

---

---

---

---

---

---

### Lektion 2: Introduktion till MVC

- Views och Controllers
- Razor view - exempel.
- Livscykel för begäran.
- Default route.
- Funktioner i ASP.NET MVC.



---

---

---

---

---

---

---

---

### Models, Views och Controllers

- Model har .NET-klasser som representerar hur objekt hanteras av webbsite.
- Flesta webbsiter lagrar information i databas, i en MVC-applikation inkluderar model programkod som läser och skriver poster i databas.
- ASP.NET MVC fungerar med många olika ramverk för tillgång till databas, Entity Framework vanligast.



---

---

---

---

---

---

---

---

---

---

## Models, Views och Controllers (forts.)

- I en MVC-applikationer, är controller .NET-klasser som besvarar begäran från webbläsare.
- Vanligtvis en controllerklass, för varje modelklass.
- Controllers inkluderar händelser, dessa är metoder som körs för att besvara begäran från användare.
- Controller ärver från basklassen System.Web.Mvc.Controller, händelser ärvs från System.Web.Mvc.ActionResult.

.eeec

---

---

---

---

---

---

---

---

## Razor view -exempel

```
<h2>Detaljer</h2>
<fieldset>
<legend>Kommentar</legend>
<div class="display-label">
    @Html.DisplayNameFor(model => model.Subject)
</div>
<div class="display-field">
    @Html.DisplayFor(model => model.Subject)
</div>
<div class="display-label">
    @Html.DisplayNameFor(model => model.Body)
</div>
<div class="display-field">
    @Html.DisplayFor(model => model.Body)
</div>
</fieldset>
```

.eeec

---

---

---

---

---

---

---

---

## Livscykel för begäran

- Begärn om detaljer för produkt med id = 1:
  - Användare begär <https://plommonmannen.se/produkt/display/1>.
  - MVC routing engine undersöker begäran och räknar ut hur begäran skall vidarebefordras till produktcontroller och vad som skall visas.
  - Händelsen display (i produktcontroller) skapar ny instans av Produkt model class.
  - Produkt model class skickar förfrågan om information om produkt med ID = 1.
  - Händelsen display skapar ny instans av Produkt Display view och skickar in Produkt Model till den.
  - Razor view engine kör serversidekod i Produkt Display view för att rendera HTML, lägger till Beskrivning, katalognummer och pris till HTML.
  - Komplet HTML-sida returneras till webbläsare.

.eeec

---

---

---

---

---

---

---

---

---

---

## Default route

- Standardväg för val av view.

```
routes.MapRoute(  
    name: "Default",  
    url: "{controller}/{action}/{id}",  
)
```

.eerec

---

---

---

---

---

---

---

---

## Demonstration se under Video



.eerec

---

---

---

---

---

---

---

---

## Funktioner i ASP.NET MVC

- *Web API*, gör det enkelt att skapa web service som svarar på begäran genom enkla HTTP verb som GET, POST och DELETE.
- *Funktioner för mobila användare*, genom att involvera jQuery Mobile, kan site utvecklas för mobila användare.
- *Display mode*, gör det enkelt att välja view efter bildskärm, kommer att skriva över standardinställningarna.
- *Asynchronous controllers*, normalt synkront, men asynkrona controllers kan användas.

.eerec

---

---

---

---

---

---

---

---

---

---

### Funktioner i ASP.NET MVC (forts.)

- Stöd för OAuth och OpenID, tredje part hanterar användare.
- Sammanfogning och minimering, begärn resultat i HTML-sida, CSS och JavaScript från server, dessa kan sammanfogas och minimeras.



---

---

---

---

---

---

---

---

### Övning Skapa MVC-applikation



---

---

---

---

---

---

---

---

### Repetitionsfrågor



---

---

---

---

---

---

---

---

---

---