|  |
| --- |
| ­­­Instruktion |

Räkna med ASP.NET MVC 4

Introduktionsuppgift

|  |
| --- |
| *Författare:* Mats Loock  *Kurs:* ASP.NET MVC  *Kurskod:*1DV409 |

**Upphovsrätt för detta verk**

Detta verk är framtaget i anslutning till kursen ASP.NET MVC (1DV409) vid Linnéuniversitetet.

**Du får använda detta verk så här:**

Allt innehåll i detta verk av Mats Loock, förutom Linnéuniversitetets logotyp, symbol och kopparstick, är licensierad under:

|  |  |
| --- | --- |
|  | Creative Commons Erkännande-IckeKommersiell-DelaLika 2.5 Sverige licens. <http://creativecommons.org/licenses/by-nc-sa/2.5/se/> |

**Det betyder att du i icke-kommersiella syften får:**

* kopiera hela eller delar av innehållet
* sprida hela eller delar av innehållet
* visa hela eller delar av innehållet offentligt och digitalt
* konvertera innehållet till annat format
* du får även göra om innehållet

Om du förändrar innehållet så ta inte med Linnéuniversitetets logotyp, symbol och/eller kopparstick i din nya version!

Vid all användning måste du ange källan: ”Linnéuniversitetet – ASP.NET MVC” och en länk till <https://coursepress.lnu.se/kurs/aspnet-mvc> och till Creative Common-licensen här ovan.

# Innehåll

Du ska följa ”steg-för-steg”-instruktionen i denna introduktionsuppgift och skapa en ASP-NET MVC-applikation som ska addera två heltal en användare matar in i två textfält. Du kommer därefter att modifiera applikationen så att användaren kan välja vilket räknesätt som ska användas.

|  |  |
| --- | --- |
|  | Starta Visual Studio 2012. |
|  | Du ska skapa ett nytt projekt för en webbapplikation, välj därför File ► New ► Project....  C:\Users\mats\AppData\Local\Temp\SNAGHTML14e5f534.PNG  Figur . |
|  | Dialogrutan New Project visas.    Figur .   1. Under Installed, Templates, Visual C# markera Web. 2. Kontrollera så att .NET Framework 4.5 visas i den nedrullningsbara listrutan. 3. Markera ASP.NET MVC 4 Web Application. 4. Vid Name skriver du in projektets namn (MyValuableCalculator). 5. Ange vid Location en lämplig katalog där projektet ska sparas (C:\Projects). 6. Se till att kryssrutan vid Create directory for solution är markerad. |
|  | Dialogrutan New ASP.NET MVC 4 Project visas.  C:\Users\mats\AppData\Local\Temp\SNAGHTML14e01207.PNG  Figur .   1. Under **Select a Template** markera **Basic**. 2. Kontrollera så att View engine är satt till Razor. |
|  | Projektet Visual Studio skapar innehåller filer och kataloger.    Figur .  /App\_Data  Används för att lagra applikationsdata, som t.ex. SQL-databaser, XML-filer.  /App\_Start  Denna katalog är bara en konvention och den separerar en del kod från Global.asax.  /Content  Här placerar du CSS-filer, bilder och övrigt icke dynamiskt innehåll; Javascript placerar i en egen katalog.  /Controllers  Här placerar du Controller-klasser som hanterar URL-begäran som skickas till webbapplikationen.  /Models  Här placerar du klasser som representerar och manipulerar data och affärsobjekt.  /Scripts  Här placerar du JavaScript-bibliotek och -skript (.js).  /Views  Här placerar du vy-filer som ansvar för rendering av t.ex. HTML. |
|  | Högerklicka på katalogen Models och välj Add ► Class….  C:\Users\mats\AppData\Local\Temp\SNAGHTML25cd5fc2.PNG  Figur . |
|  | Dialogrutan Add New Item visas.  C:\Users\mats\AppData\Local\Temp\SNAGHTML25cf2c34.PNG  Figur .   1. Ange namnet på klassen (ElementaryMath) vid Name. 2. Klicka på Add. |
|  | Klassen ElementaryMath,i namnområdet MyValuableCalculator.Models, ansvarar för att två heltal ska adderas. De två autoimplementerade egenskaperna Op1 och Op2 kommer att innehålla de värden användaren matar in i formulärets textfält. Metoden Compute utför additionen av Op1 och Op2 och summan lagras i ”read-only”-egenskapen Result.    Figur . |
|  | För att Visual Studio ska bli fullt medveten om att en ny klass lagts till projektet bygger du projektet genom att trycka på Skift+F6. Du kan också välja Build ► Build MyValuableCalculator.  C:\Users\mats\AppData\Local\Temp\SNAGHTML25d2dbc4.PNG  Figur . |
|  | Högerklicka på katalogen Controller och välj Add ► Controller….  C:\Users\mats\AppData\Local\Temp\SNAGHTML25d4657c.PNG  Figur . |
|  | Dialogrutan Add Controller visas.  C:\Users\mats\AppData\Local\Temp\SNAGHTML25d6484c.PNG  Figur .   1. Vid Controller Name skriver du in namnet, HomeController, på kontrollern. ***OBS!*** *Det är mycket viktigt att namnet på kontrollern avslutas med Controller*. 2. Klicka på Add. |
|  | Kontrollern HomeController skapas.    Figur . |
|  | Komplettera metoden Index för att skapa en modell, en instans av klassen ElementaryMath, som skickas till vyn.    Figur . |
|  | Högerklicka någonstans i metoden Index och välj Add View….  C:\Users\mats\AppData\Local\Temp\SNAGHTML29711dc6.PNG  Figur . |
|  | Dialogrutan Add View visas.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29727354.PNG  Figur .   1. Det är viktigt att vyn har samma namn som metoden i kontrollern, d.v.s. Index. 2. Kontrollera så att Razor är valt under View engine:. 3. Markera Create a strongly-typed view. 4. Vid Model class: välj klassen ElementaryMath (MyValuableCalculator.Models). 5. Under Scaffold template ska Empty vara valt. 6. Reference script libraries ska vara markerad. 7. Klicka på Add. |
|  | Filen Index.cshtml skapas i katalogen \Views\Home.  Vyer som tillhör kontrollern HomeController måste placeras i katalogen \Views\Home. Det är vanligt att vyernas namn är detsamma som kontrollmetoden som använder dem, men det behöver inte vara så.    Figur . |
|  | Med Html-hjälpmetoder skapas ett formulär. Den första raden talar om att vyn är starkt typad och dess modell är av typen ElementaryMath.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a63ca2f.PNG  Figur . På raderna 10, 12 och 14 refererar m till objektet som utgör modellen. |
|  | Webbapplikationen är nu färdig för att provköras för första gången. Enklast provkör du den genom att trycka på Ctrl+F5.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a663a5f.PNG  Figur .  Den renderade HTML-koden innehåller i princip inget du inte skulle skrivit för hand. Textfälten namnges automatiskt med namnet egenskaperna, Op1 respektive Op2, har i klassen ElementaryMath. Någon HTML för egenskapen Result renderas inte eftersom den har värdet null.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29816264.PNG  Figur . |
|  | För att kunna behandla formulärdatat behöver klassen HomeController kompletteras.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29854218.PNG  Figur .  Metoden med attributet HttpPost tar hand om URL-begäran av typen POST. Formulärdatat används för att initiera ElementaryMath-objektet parametern elementaryMath refererar till. Egenskaperna Op1 och Op2 kommer att innehålla de värden användaren matat in i respektive textfält i formuläret. |
|  | Provkör webbapplikationen med Ctrl+F5. Skriv in värden 25 och 17. Klicka på kommandoknappen och konstatera att den beräkna summan helt korrekt blir 42.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a6707bc.PNG  Figur .  Radera innehållet i textfälten och klicka på kommandoknappen. Konstatera att textfälten nu formateras annorlunda och att någon summa inte presenteras. Varför? Någon validering är i egentlig mening inte implementerad!  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a6b16e8.PNG  Figur . |
|  | För att implementera validering måste klassen ElementaryMath kompletteras så att validering sker med hjälp av ”*data annotations*”.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29a4249a.PNG  Figur . |
|  | I kontrollermetoden måste modellens status undersökas. Om objektet uppfyller de krav valideringen ställer ska en beräkning ske och objektet, innehållande resultatet skickas vidare till vyn. Klarar inte objektet valideringen skickas objektet utan beräkning gjord men innehållande felmeddelanden tillbaka till vyn.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29b2f580.PNG  Figur . |
|  | Även vyn Home/Index.aspx måste modifieras så valideringsmeddelanden visas.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a6f6cd4.PNG  Figur . |
|  | Nu visas felmeddelande om användaren inte matar in något i textfälten och klickar på kommandoknappen.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a708e1a.PNG  Figur .  Felmeddelanden visas även då användaren matar in något som inte kan tolkas som ett heltal.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a712bfe.PNG  Figur . |
|  | Nu sker all validering på servern. För att valideringen även ska ske på klienten måste Shared/\_Layout.cshtml kompletteras.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a72fc57.PNG  Figur . |
|  | Nu sker valideringen omedelbart när användaren skriver i textfälten.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a73c772.PNG  Figur .  Klickar användaren på kommandoknappen postas inte formuläret men felmeddelanden visas ändå.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2a7436a8.PNG  Figur . |
|  | Som service till användaren av webbapplikationen skulle det vara trevligt om det första textfältet har fokus när sidan laddat klart. Det kan ordnas enkelt med jQuery. Vyn Home/Index.aspx behöver modifieras.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29ee2a27.PNG  Figur .  (Fungerar skriptet verkligen? Det gör det eftersom en funktion som skickas som ett argument till jQuery-konstruktorn binds automatiskt till händelsen ready, varför $(document).ready() inte behöver anges explicit.) |
|  | Nu är webbapplikationen nästan klar. Punkterna som följer visar hur en nedrullningsbar listruta kan användas för att användaren ska kunna välja räknesätt. |

|  |  |
| --- | --- |
|  | Modellen, d.v.s. klassen ElementaryMath, måste modifieras så att den kan hantera de fyra räknesätten.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29f4ed37.PNG  Figur . |
|  | Skapa en ny katalog, ViewModels, under projektroten. I denna katalog placerar du filer med klasser, innehållande vyspecifikt data som inte har med modellen att göra, controllrar kan skapa instanser av för att skicka till vyer.  Lägg till en ny klass med namnet HomeIndexViewModel i katalogen ViewModels.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2fa77c6a.PNG  Figur . |
|  | Klassen ElementaryMathViewModel innehåller två publika egenskaper. Egenskapen ElementaryMath refererar till objektet som utgör själva modellen. ”Read-only”-egenskapen ArithmeticOperations returnerar en lista med de fyra räknesätten som den nedrullningsbara listrutan i vyn ska visa.  C:\Users\mats\AppData\Local\Temp\SNAGHTML29fba511.PNG  Figur . |
|  | Vyn Home/Index.aspx måste modifieras så den använder vymodellklassen HomeIndexViewModel istället för modellklassen ElementaryMath. Dessutom ska +-tecknet ersättas med en nedrullningsbar listruta.  C:\Users\mats\AppData\Local\Temp\SNAGHTML3f6defd7.PNG  Figur .  Lägg märke till att Op1, Op2 och Result nu måste förgås av ElementaryMath, som ju egenskapen i vymodellklassen heter som refererar till modellobjektet av typen ElementaryMath. |

|  |  |
| --- | --- |
|  | Även metoderna i klassen HomeController måste modifieras att använda klassen HomeIndexViewModel.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2fa06fdc.PNG  Figur .  Det enda som behöver ändras är att objektet som skickas till vyn ska vara av typen HomeIndexViewModel. ASP.NET MVC kan fortfarande binda formulärdatat till objektet av typen ElementaryMath. Detta fungerar eftersom parametern (elementaryMath) har samma namn som egenskapen har i vymodellen (ElementaryMath). |

|  |  |
| --- | --- |
|  | Nu kan användaren välja om de två talen ska summeras, subtraheras, multipliceras eller divideras.  C:\Users\mats\AppData\Local\Temp\SNAGHTML2f9b69d3.PNG  Figur .  Valideringen fungerar fortfarande trots förändringarna som gjorts. Det är bara att prova…  C:\Users\mats\AppData\Local\Temp\SNAGHTML2f9ce7e8.PNG  Figur . |
|  | Nu är webbapplikationen klar. Förhoppningsvis har du nu fått en introduktion till ASP.NET MVC och förstår grundläggande koncept, även om det är mycket mer att lära. |