Linnéuniversitetet
Kalmar Växjö

Laborationsanvisning

Hur många versaler?

Steg 1, laborationsuppgift 1

Författare: Mats Loock

Kurs: ASP.NET Web Forms

*Kurskod:*1DV406

Linnéuniversitetet
Kalmar Växjö

Upphovsrätt för detta verk

Detta verk är framtaget i anslutning till kursen ASP.NET Web Forms (1DV406) vid Linnéuniversitetet.

Du får använda detta verk så här:

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

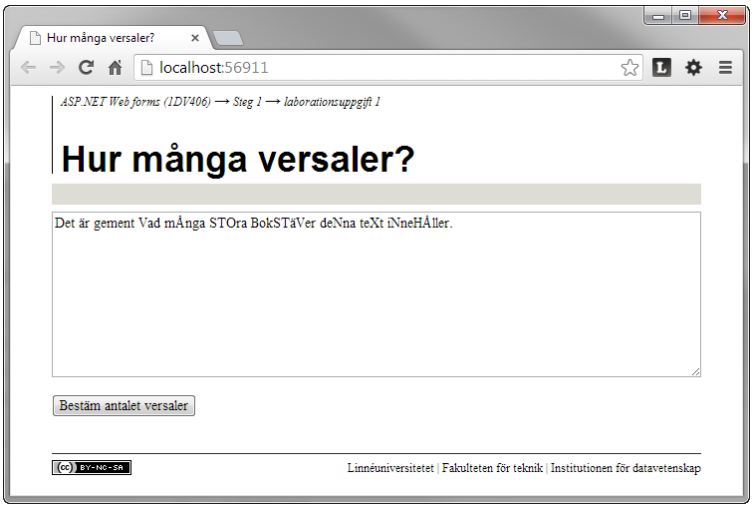
Om du förändrar innehållet så ta inte med Linnéuniversitetets logotyp, symbol och/eller kopparstick i din nya version!

Innehåll

Uppgift

Inledning

Skriv en webbapplikation bestående av en sida där användaren kan mata in en text. Texten ska analyseras och antalet inmatade versaler (stora bokstäver) ska bestämmas och presenteras.



Figur 1. Sida där användaren matar in text.

Användaren ska kunna mata in vilken text som helst i ett textfält. Då användaren klickar på en kommandoknapp ska formuläret skickas tillbaka (”*postback*”) och texten analyseras för att bestämma antalet versaler. Svaret skickas tillbaka och antalet versaler presenteras på lämpligt sätt.

Figur 2. Samma sida som figur 1 men här visas resultatet efter analysen av texten.

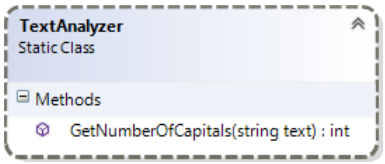
Då svaret presenteras ska det inte vara möjligt att ändra i textfältet. Den ska vara ”grå-markerad”, vilket görs genom att sätta egenskapen `Enabled` till `false`. För att göra en ny bestämning av antalet versaler i en text måste användaren klicka på en kommandoknapp som skickar tillbaka sidan till servern som returnerar sidan som den såg ut då den visades första gången. Textfältet ska alltid initialt vara tom då det är möjligt att mata in text i det.

Inkapsling (*Separation of Concern*)

Du ska separera hanteringen av data från kod som har med presentationslogiken att göra så långt det är möjligt. Koden som har med presentationslogiken att göra blir då så mycket enklare att hantera. Du ska därför placera all kod som har med bestämning av antalet versaler i en text i ett så kallat

affärslogiklager bestående av en separat C#-klass som placeras i katalogen `Model`, som du själv måste skapa.

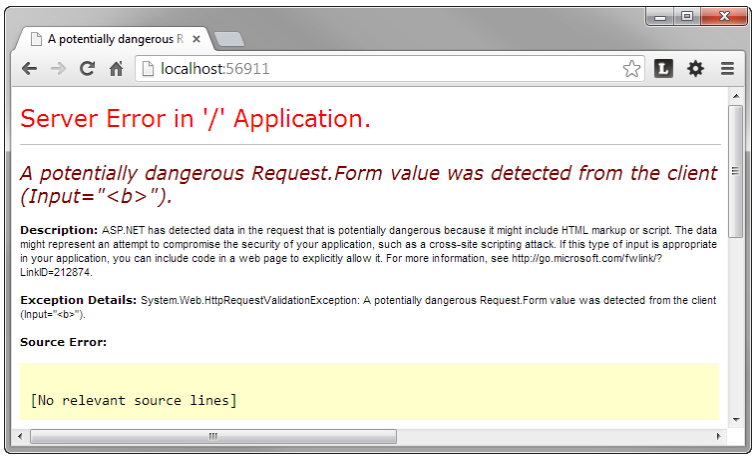
Implementera den statiska klassen `TextAnalyzer` enligt klassdiagrammet i figur 3. Klassen har bara en statisk metod som returnerar antalet versaler som finns i strängen som skickas som argument till metoden.



Figur 3. Den statiska klassen TextAnalyzer.

Hantering av fel på servern

Du behöver inte använda dig av någon validering av textfältet men du ska se till att användaren slipper se sidor med automatgenererade felmeddelanden enligt figur 4 om nu användaren t.ex. skriver en tagg som `<b>`.



Figur 4. ”Gul-ful” sida som visas vid fel innan redigering av Web.config.

Istället för de automatgenererade felmeddelandena ska du istället visa en sida med ett användarvänligare(?) meddelande. Figur 5 visar ett exempel på en sådan sida.

Figur 5. Anpassad sida som visas vid fel efter redigering av Web.config.

På sidan *customErrors Element (ASP.NET Settings Schema)*, http://msdn.microsoft.com/en-us/library/h0hfz6fc(v=vs.100).aspx, beskrivs vad du måste ändra på i `Web.config` för att åstadkomma detta.

Undersök och prova vad som händer då du i `Web.config` för elementet `<customErrors>` ändrar på attributet `mode` då det får de olika värdena `On`, `Off` och `RemoteOnly`.

Mål

- Förstå hantering av kontroller och hur dessa behåller sina värden med hjälp av ”view state”.
- Du ska veta att serverkontrollen `TextBox` renderas ut som ett textarea-element då egenskapen `TextMode` till `MultiLine`.
- Kunna hantera det en `TextBox`-kontroll innehåller med hjälp av egenskapen `Text`.
- Förstå att då användaren klickar på en knapp sker en ”postback” och en händelse skapas som kan tas om hand i ”code-behind”-filen.
- Kunna ta hand om händelsen `Click` som inträffar då användaren klickar på en knapp.
- Använda `Label`-kontrollen för att placera ut text.
- Förstå vikten av att, och hur man, presenterar användarvänliga felmeddelanden.

Läsvärt

- På sidan ”*ASP.NET Page Life Cycle Overview*”, http://msdn.microsoft.com/en-us/library/ms178472.aspx, beskrivs händelser som inträffar och i vilken ordnng de utlöses då en sida genereras på servern.
- Hur du knyter en händelse till en `Button`-kontroll hittar du på sidan ”*Button.Click Event*”, http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.click(v=vs.110).aspx.
- Använd en stilmall för att utforma dokumentet, gör det inte ”inline”. På sidan *Working with CSS Overview* , http://msdn.microsoft.com/en-us/library/bb398931.aspx, hittar du den information du behöver.
- Du kan bestämma antalet versaler i en sträng på flera sätt. Du kan t.ex. med hjälp av klassen `Regex` och ett reguljärt uttryck rensa bort alla tecken i en sträng som inte är versaler för att sedan ta reda på längden strängen har. Du kan också stega igenom strängen tecken för tecken och använda den statiska metoden `Char.Isupper` för att undersöka om ett tecken är versal eller inte.
- Du behöver bara använda en `Button`-kontroll. Byt bara texten på knappen med hjälp av egenskapen `Text`.
- En kontroll kan ”gömmas” genom att sätta egenskapen `Visible` till `false`. Faktum är att kontrollen inte ens renderas ut då.