Dynamiske kontroller - gængse fælder.

Kontrol forsvinder, ViewState problemer, event handler kører ikke... har trådt på alle disse miner selv:) Artiklen er møntet på ikke helt befarne brugere af ASP.NET

Skrevet den **03. Feb 2009** af **neoman** I kategorien **Programmering / ASP.NET** | ★★★☆☆

Det, som jeg beskriver her, har voldt mig mange problemer, og så vidt jeg kan se også for ret mange andre. Derfor synes det nærliggende at dele erfaringerne.

1. Hvorfor forsvinder min kontrol ved postback ?

Man kunne få indtrykket af, at når kontrollen engang er loaded, at den bliver der (er det ikke hvad vi har ViewState til?). Men dette er ikke rigtigt - alt hvad du loader dynamisk lever kun indtil den næste postback.

Hvis du ønsker at se din(e) kontrol(er) igen, så skal du reloade dem også på hver postback. Du kan jo altid lægge en værdi ind et sted, f.eks. ViewState, Session, eller et hidden field, tjekke værdien ved PostBack, og reloade den/de kontroller som nødvendigt, i hht. den gemte information.

Meen - selve kontrollens ViewState overlever - selv for dynamiske kontroller. Se punkt 2.

2. Den vil ikke reloade kontrollen - brokker sig over at ViewState ikke matcher.

Frameworket anvender et komposit-ID på klient-siden til at identificere en kontrol, og matcher den med ViewState ved postback. Dette komposit-ID kan du se ved at kigge på den i sourcen i din .aspx side, som ligger på klienten. Det består af en masse ctlXX'er, hvor der er en ctlxx for hvert niveau: dvs din kontrol, dennes container, containerens container osv, indtil vi når toppen af kontroltræet.

Hvis du f.eks. har en user control, som du ikke selv har givet et ID, så får den automatisk et ID som f.eks. ctl01. Hvis du nu, efter postback, loader en anden user control, som heller ikke har noget tilskrevet unikt ID, så får den også ctl01. Og nu er der problemer med ViewState - som kun matcher i fald de to kontroller er 100 % ens.

For at det skal gå godt, så skal hver kontrol have et unikt ID som du skal tilskrive den. Så er det dét ID, som frameworket bruger (tilsat de sædvanlige ID'er for containeren osv).

Hvis du nu ikke ønsker at tilskrive ID selv, så skal du sørge for, at alle dynamisk tilføjede kontroller reloades i nøjagtigt samme rækkefølge som første gang. F.eks., hvis du dynamisk tilføjer en hel række tekstboxe, så skal du ved postback tilføje dem i samme rækkefølge.

Hvis du fjerner en af dem (fjerner=undlader at reloade den efter postback), og den fjernede kontrol ikke er den sidste i rækken af dynamisk tilføjede kontroller, så vil den efterfølgende dynamiske kontrol per automatik få den forudgående (og altså nu fjernede) kontrols ID ved postback. Frameworket vil forsøge at matche viewstate med den. Hvis kontrollen er af nøjagtigt samme type som den forudgående (og som nu er fjernet), så vil denne match lykkes, og du kan få "forkert" viewstate, hvilket kan være

uheldigt. Hvis typen afviger, så får du en fejlmeddelelse og et crash.

3. Hjælp - min eventhandler kører ikke (eller kører, men ikke på første klik).

Du har en eller anden dynamisk tilføjet kontrol, du husker at reloade den ved postback, men hvorfor h... kører det event, som en af kontrollens knapper skulle trigge, ikke ? Du har nøje læst det som jeg har skrevet lige ovenover, og har givet knappen et unikt ID, men det virker stadig ikke!!

Igen - kig på ID på klienten ved første load og ved postback:

1. Efter kontrollen er loaded, find i sourcen på klienten den knap som skulle trigge noget og se dens kompositte ID
2. Efter du har trykket på knappen og fået postback, kig igen i sourcen, og se hvad, om noget, har ændret sig i knappens kompositte ID.

Ovenstående skulle gerne vise, hvilket element som har fået ændret sit ID - enten knappen, eller en af parent-containerne.

Det er hele den kompositte client-ID som skal matche. Det kan ske, at tingene bliver loaded i en lidt anden rækkefølge første gang og ved postback. Hvis din knap sidder i en user control, eller i en anden container, som ikke har fået noget ID af dig, og load rækkefølgen ændrer sig, så kan frameworket ikke finde eventhandleren, da den er bundet til det ID som knappen havde, den gang der blev trykket på knappen.

Kuren er igen at sørge for at give ID til kontrollen og dens container (og dennes container..hele vejen til toppen), eller at sørge for at alt reloades i nøjagtigt samme rækkefølge.

Tilføjelse 12/09-07
En alternativ måde at få det til at gå i ged er at ID ændres ved postback, ikke fordi frameworket laver fejl, men fordi man selv ændrer ID. For at debugge: igen, kig på ClientID i sourcen ude på klienten, og check om det ændrer sig ved postback. Hvis den pågældende kontrol sidder inde i et Ajax UpdatePanel, og sourcen ikke opdateres "ordentligt", pil kontrollen ud derfra midlertidigt for tests.

Hvis du tilføjer din kontrol for sent i sidens life-cycle **(for sent=efter PageLoad ved postback)**, så vil din event handler heller ikke køre. (For beskrivelse af rækkefølgen find "ASP.NET Page Life Cycle Overview" i Google). Networket foretager matching mellem ViewState (og ControlState) med kontrollerne, men hvis kontrollen endnu ikke er defineret, så vil en event handler ikke køre, da networket ikke kan finde den kontrol som handleren tilhører.

Rækkefølgen i sidens life-cycle er (som kopieret fra ASP.NET Page Life Cycle Overview, som den så ud i skrivende stund):

Page request
Start
Page initialization
Load
During load, if the current request is a postback, control properties are loaded with information recovered from view state and control state.
Validation
During validation, the Validate method of all validator controls is called, which sets the IsValid property of individual validator controls and of the page.
Postback event handling

If the request is a postback, any event handlers are called.
Rendering
Before rendering, view state is saved for the page and all controls. During the rendering phase, the page calls the Render method for each control, providing a text writer that writes its output to the OutputStream of the page's Response property.
Unload

4. Reloade? Ja men i hvilket page event ?

De dynamiske kontroller jeg tilføjer, reloader jeg i PageLoad og det går godt for mig. Ude på nettet ser man mange råd om at man skal udføre dette i PreInit - det har noget at gøre med på hvilket tidspunkt i sidens life cycle viewstate matches med kontrollerne. I mange tilfælde skyldes problemerne det som jeg beskrev i punkterne 1,2 og 3.

Jeg ved ikke nok om emnet til at kunne give gode råd - har blot beskrevet det som virker for mig, og formentligt for mange andre. Hvis tiltagene beskrevet her ikke virker for dig, så kan det være, at du skulle kigge på reloads i PreInit.

5. Hvordan er det så lige jeg skal gøre, når jeg bruger AJAX?

Når der er events som skal opdatere din side, så er det kun sidens **præsentation i browseren** som opdateres partielt (altså, AJAX opdaterer din UpdatePanel f.eks). Selve din code-behind udføres ganske normalt: Page Load eksekveres, samt, efterfølgende, alle de event handlers du nu måtte have trigget.

Ovenstående er ret logisk, men det tog mig en lille stund at forstå, at det var sådan, det forholdt sig. Det medfører jo også, at dine dynamiske kontroller skal tilføjes på nøjagtigt samme vis, som når du kører uden AJAX.

Den lille fælde der kan forekomme er, at din kontrol nu sidder inde i endnu en container - f.eks. UpdatePanel1 .. og pludselig virker din applikation ikke mere. Det er fordi du skal huske, for at tilgå/sætte værdier på kontroller i en container, at finde controllen først ved at bruge metoder som f.eks. Navnet_på_containeren.FindControl metode, eller Navnet_på_containeren.Controls(controlIndex)

Lidt tricky kan være det, at få styr på triggers og UpdatePanels. Mange synes at blive overrasket at UpdatePanels opdaterer, eller rydder sit indhold, selv om de ikke er blevet trigget direkte. Bemærk parameteren UpdateMode: som default value har den "Always", hvilket bevirker, at en sådan UpdatePanel vil blive opdateret ved alle async events på siden.

Husk endeligt også at holde styr på kontrollernes ID'er, jvf. punkterne 2 og 3.

Konstruktive kommentarer/rettelser modtages gerne - så kan de evt. blive indlagt i artiklen.

Kommentar af prof2 d. 12. Jun 2007 | 1

Kommentar af -xyz- d. 18. Jun 2007 | 2

Mange tak gav mig en meget bedre forståelse for dynamiske kontroller

Kommentar af dr_chaos d. 17. Sep 2007 | 3

Kommentar af adamski3004 (nedlagt brugerprofil) d. 25. Jun 2007 | 4

tak men det gav mig ikke mange svar men jeg har fået bedre forståelse for dynamiske kontroller