

ASP.NET MVC

Grzegorz Caban
grzegorz.caban@gmail.com

20 stycznia 2009

Agenda

- Przyczyna powstania
- Co to jest ASP.NET MVC
- Architektura
- Hello World w ASP.NET MVC
- ASP.NET MVC vs ASP.NET WebForm
- Przyszłość framework'a

Idea ASP.NET WebForm

- Tworzenie aplikacji webowych równie łatwo jak desktopowych
- Http jest bezstanowym protokołem!
- Warstwa abstrakcji nad protokołem http
- Mechanizm przechowywania stanu strony/formularza pomiędzy zapytaniami (ViewState)
- Bogate w funkcjonalność kontrolki (RAD)

BROWSER

(ShopCart.aspx)

Shopping Cart

- Add
- Add
- Add

"Add" button submits to server...

SERVER

ShopCarts.aspx.cs

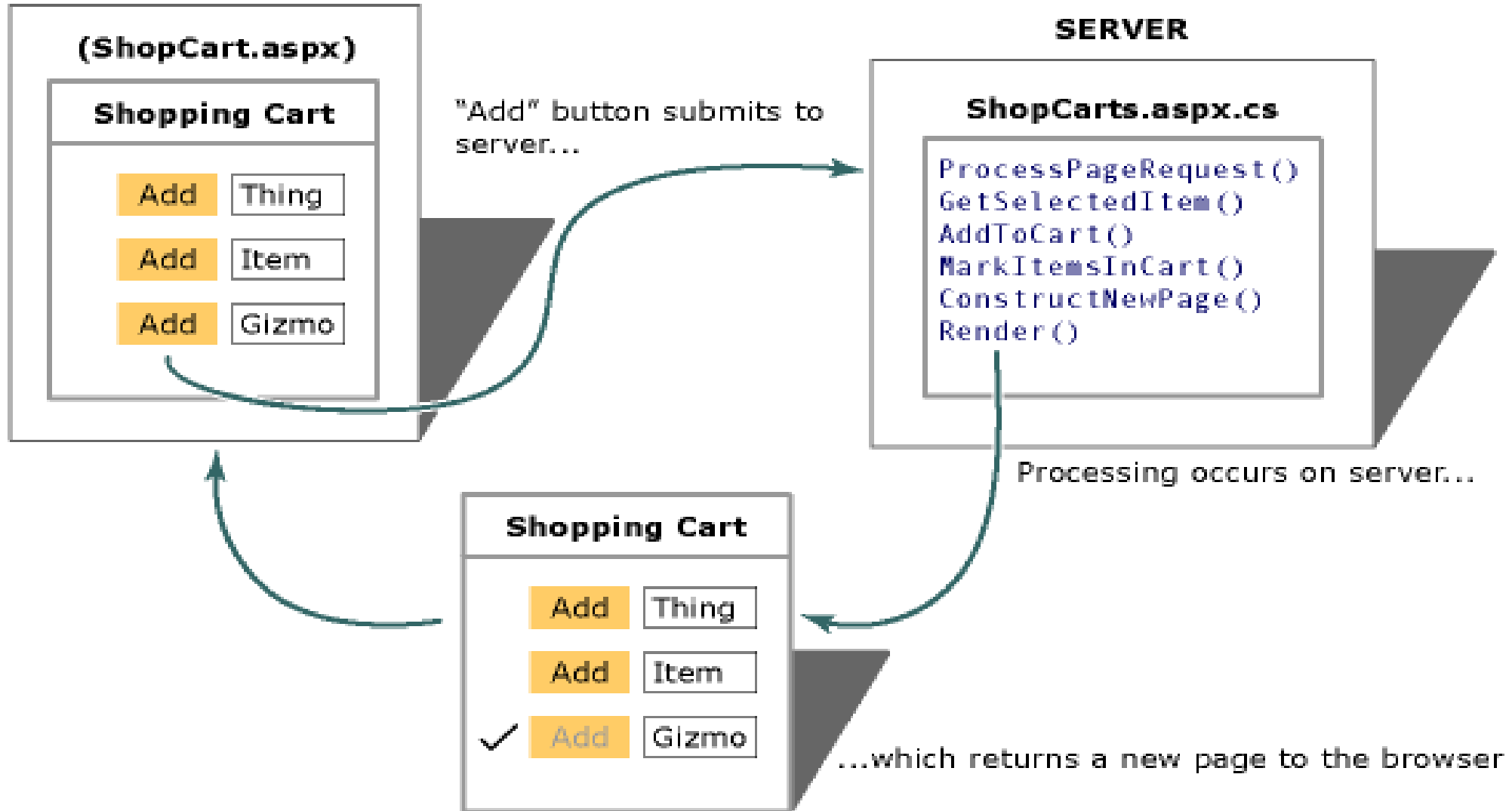
```
ProcessPageRequest()  
GetSelectedItem()  
AddToCart()  
MarkItemsInCart()  
ConstructNewPage()  
Render()
```

Processing occurs on server...

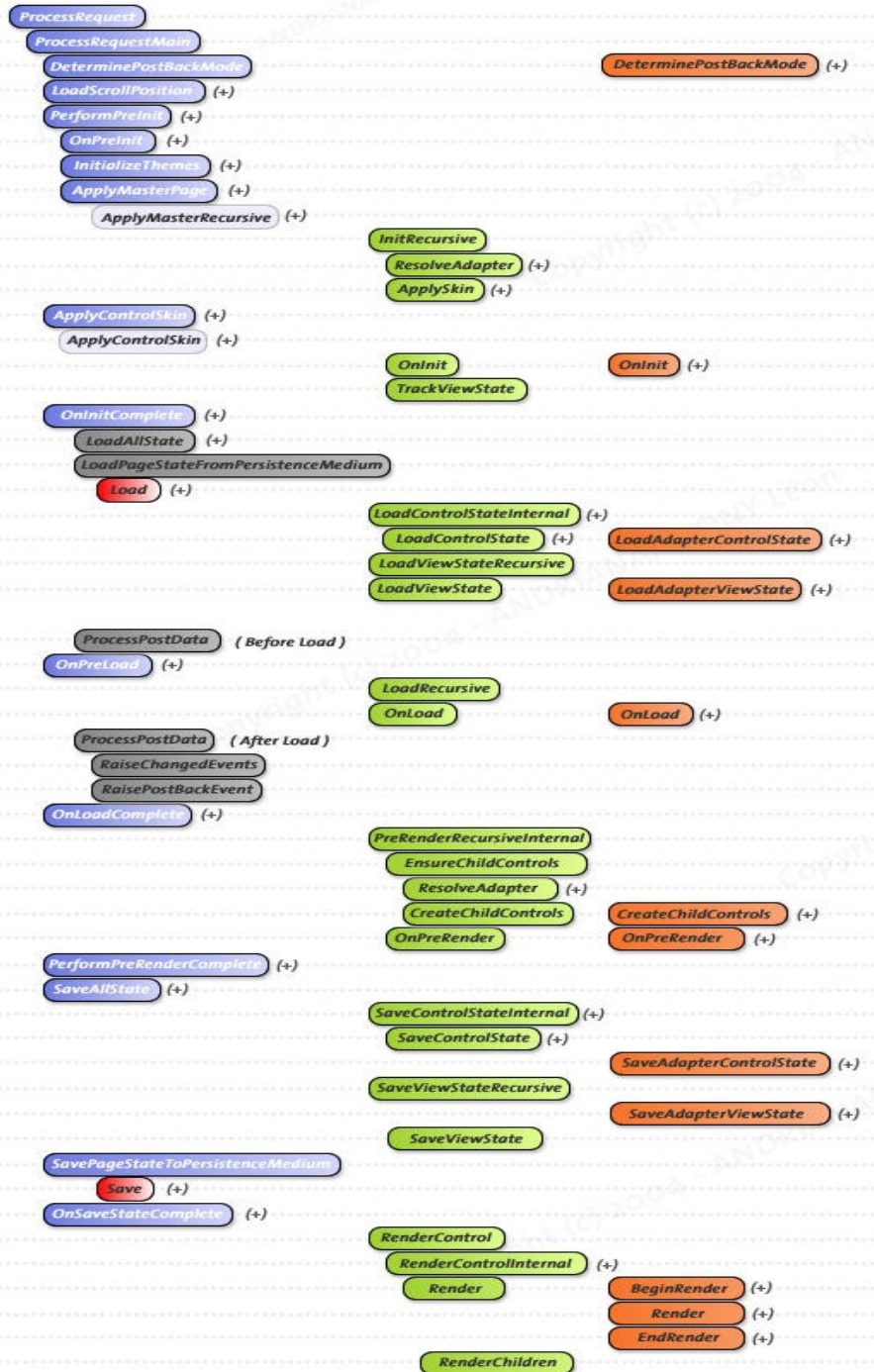
Shopping Cart

- Add
- Add
- ✓ Add

...which returns a new page to the browser



ASP.NET Page LifeCycle



Legend

- Page
- Control
- Adapter
- PostBack
- Page Persister

(+) means added in V2.0 since V1.1

Copyright (c) 2004 - Léon Andrianarivony

Przyczyna powstania MVC

Czyli co jest nie tak z ASP.NET WebForm?:

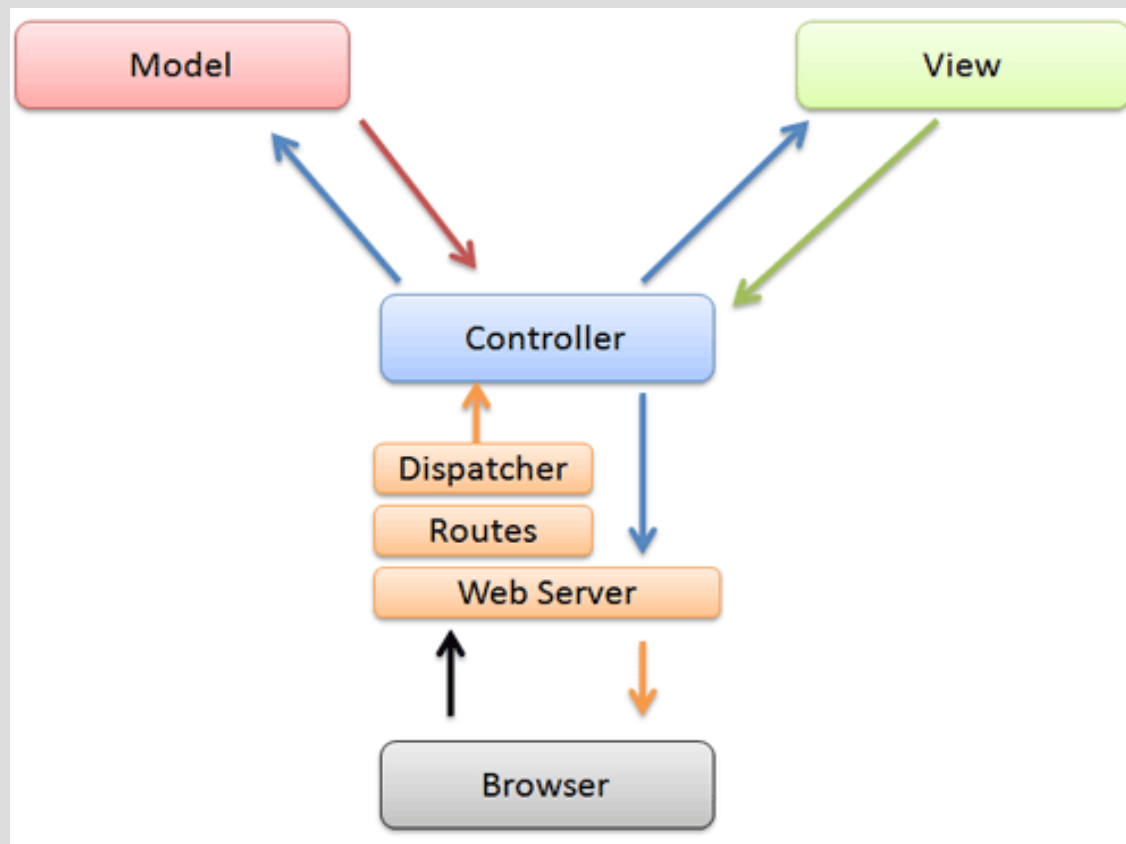
- View State – często kilkaset kB
- Mechanizm Postback bywa skomplikowany i mało intuicyjny
- Brak kontroli nad renderowanym html
- Brak podziału na logikę i prezentację
- Page Life Cycle

Co to jest ASP.NET MVC

- Odpowiedź na powyższe problemy..
- Nowy framework do tworzenia aplikacji webowych na platformie .NET
- Oparty o bardzo popularny wzorzec MVC (wiele rozwiązań dla PHP, Javy, Pythona) ale też dla samego .NET np. MonoRails, nStruts

Architektura

- Klasyczna implementacja wzorca MVC



Architektura

- Akcje użytkownika (zapytania http) przekładają się tylko na wywołania metod Controllera i zwrócenie Widoku
- Stan aplikacji zapisywany tylko w Modelu
- Widoki służą tylko do generowania prezentacji, brak logiki
- Luźno powiązane wyspecjalizowane elementy

Controller

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using System.Web.Mvc.Ajax;
7
8 namespace MvcApplication1.Controllers
9 {
10     public class TestController : Controller
11     {
12         public ActionResult Index()
13         {
14             // Add action logic here
15             return View();
16         }
17     }
18 }
19
```

Controller

- Mapowanie URL na publiczne metody kontrolera, inaczej Akcje

np. `http://localhost/Test/Index/1`

odpowiada wywołaniu metody `TestController.Index(1)`

- Sposób mapowania ustawiany w RouteTables w `Global.asax`
- Akcje kontrolera pobierają dane z modelu, realizują logikę biznesową i zwracają wyniki do właściwego widoku
- Po przekazaniu danych rozpoczyna generowanie widoku

View

```
1 <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Index.aspx.cs" Inherits="MvcApplication1.Views.Test.Index" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" >
6 <head runat="server">
7     <title></title>
8 </head>
9 <body>
10 <div>
11
12     <h1>Lista zamówień</h1>
13
14     <table>
15
16
17     <% foreach (Order order in (IEnumerable)ViewData.Model) {%>
18
19         <tr>
20             <td><%=order.ID %></td>
21             <td><%=order.BARCODE %></td>
22             <td><%=order.STATUS %>"</td>
23             <td><a href="/Data/OrderDetails?orderBarcode=<%=order.BARCODE %>">Wybierz</a> </td>
24         </tr>
25
26     <% } %>
27     </table>
28 </div>
29 </body>
30 </html>
31
```

View

Plik umiejscowiony w odpowiednim folderze np Views/Test/Index.aspx posiadający zawartość:

- Statyczną : Html, javascript itp.
- Dynamiczną : skrypt w C# (vb.net etc)
- HtmlHelper – klasa ułatwiająca generowanie html
- Master Page i UserControls

Model

- Przechowuje stan aplikacji pomiędzy akcjami kontrolera
- Model najczęściej przechowywany w bazie danych
- Integracja z Linq, Entity Framework, NHibernate, SubSonic, LLBLGen Pro i itp.
- Możliwość generowania Widoków do Edycji, Listy itp. na podstawie klasy Modelu

Przykładowa aplikacja

Co jest potrzebne aby zacząć :

- Visual Studio 2008 lub Visual Web Developer 2008 Express (SP1)
- Microsoft ASP.NET MVC framework
- Do dzieła..

Zalety ASP.NET MVC

- Rozdzielenie elementów odpowiedzialnych za poszczególne role
- Bardziej pasujący do aplikacji webowych (brak sztucznych „stanów” w Http)
- Pełna kontrola nad renderowanym html
- Łatwa integracja z popularnymi frameworkami js(np. jQuery)
- Wsparcie dla zewnętrznych rozwiązań generowania html (np. nVelocity)
- Wsparcie dla Unit Testow
- URL „przyjazne wyszukiwarka”

Zalety ASP.NET WebForm

- Intuicyjne dla lubiących WinForms (model zdarzeniowy, pseudo stanowość w Http)
- Rozbudowane kontrolki zapewniające wiele funkcjonalności „out-of-box”
- RAD
- Znana sprawdzona technologia, wiele przykładów komponentów itp

Release history

- 2007-12-10 ASP.NET MVC Framework launched as a Community Technology Preview
- 2008-03-05 ASP.NET MVC Preview 2 released
- 2008-05-01 ASP.NET MVC Preview 3 released
- 2008-07-16 ASP.NET MVC Preview 4 released
- 2008-08-28 ASP.NET MVC Preview 5 released
- 2008-10-16 ASP.NET MVC Beta released
- 2009-01-?? ASP.NET MVC Version 1 full release

Przyszłość ASP.NET MVC

- Aktualnie wersji Beta
- Dalsze rozwijanie tego projektu, spotkał się z dobrym odbiorem społeczności
- Ale też ma to być alternatywa dla WebForm, a nie następca

Linki

- <http://www.asp.net/mvc/>
- <http://weblogs.asp.net/scottgu/default.aspx> - blog ScottaGu
- <http://weblogs.asp.net/shijuvarghese/archive/2008/07/09/asp-net-mvc-vs-asp-net-web-form.aspx> - porównanie MVC i WebForm



Pytania?

Dziękuję za uwagę