10/17/2009

|  |
| --- |
| http://serviciipeweb.ro/iafblog  |



|  |  |
| --- | --- |
| Andrei Ignat | ASP.NET MVC TIPS AND TRICKS |

Contents

[Cui ii este adresata aceasta carte 4](#_Toc246290437)

[Despre mine 4](#_Toc246290438)

[Multumiri (in ordine aleatoare) 4](#_Toc246290439)

[De ce ASP.NET MVC si nu WebForms 5](#_Toc246290440)

[Pentru incepatorii absoluti 5](#_Toc246290441)

[Pentru cei care au lucrat cu ASP.NET MVC 5](#_Toc246290442)

[Primul site cu ASP.NET MVC 5](#_Toc246290443)

[Probleme comune: 6](#_Toc246290444)

[Nu merge pe Windows XP sau Windows 2003 (IIS 5.1 /IIS 6  ) 6](#_Toc246290445)

[Copiez site-ul la hosting si obtin : Runtime Error - could not load file or assembly System.Web.Mvc.dll) 7](#_Toc246290446)

[Nu imi vine valoarea parametrului din actiunea controllerului! 8](#_Toc246290447)

[Nu obtin nici o valoare in parametrul FormCollection ! 9](#_Toc246290448)

[Cum verifici ca actioneaza ErrorHandler 10](#_Toc246290449)

[Caractere speciale in adresa 11](#_Toc246290450)

[Best practices 12](#_Toc246290451)

[Nu folositi ViewData - nici macar pentru a trimite datele intre master si view-uri 12](#_Toc246290452)

[Folositi testarea automata 13](#_Toc246290453)

[Logati erorile 16](#_Toc246290454)

[Alte best practices 17](#_Toc246290455)

[Interceptati eroarea "HTTP 404 - Not Found" 17](#_Toc246290456)

[Nu folositi output caching – ci inmemorycache 18](#_Toc246290457)

[Accesorii 21](#_Toc246290458)

[Verificarea rutelor 21](#_Toc246290459)

[Testarea actiunii controlerelor 21](#_Toc246290460)

[Transmiterea datelor intre BusinessLogic si Controllere/View 24](#_Toc246290461)

[Cum sa faceti footer la un grid - sau sa adaugati alte rinduri 24](#_Toc246290462)

[Designul site-ului 27](#_Toc246290463)

[Cum sa aveti un sitemap 27](#_Toc246290464)

[Optimizarea   aplicatiei 27](#_Toc246290465)

[Accesul la baza de date 27](#_Toc246290466)

[Code protection from decompilers 27](#_Toc246290467)

[ilmerge si web deployment projects 28](#_Toc246290468)

[Ajax with jquery 28](#_Toc246290469)

[Optimizarea HTML 31](#_Toc246290470)

[Expires header pentru imagini 31](#_Toc246290471)

[Minimizarea html 32](#_Toc246290472)

[Minimizare incarcare  javascript 33](#_Toc246290473)

[Interfata grafica asemanatoare in diferite browsere 35](#_Toc246290474)

[Search Engine Optimization 35](#_Toc246290475)

[Additional helpers 36](#_Toc246290476)

[Ce urmeaza 40](#_Toc246290477)

[Linkuri utile care nu au incaput 40](#_Toc246290478)

[Tool-uri mentionate 41](#_Toc246290479)

[Postfata (Aurelian Popa) 43](#_Toc246290480)

[Nota Andrei Ignat: 44](#_Toc246290481)

Cui ii este adresata aceasta carte

Aceasta carte se adreseaza celor care vor sa lucreze cu ASP.NET MVC,http://www.asp.net/mvc/ , precum si celor care au o experienta minima cu ASP.NET MVC. In aceasta carte sunt problemele intilnite de mine in creearea site-urilor cu ASP.NET MVC.In orice caz, experienta cu HTML/JSCRIPT este necesara.

Pentru orice observatii va rog sa imi scrieti pe adresa ignatandrei@yahoo.com, cu subiectul "Asp.NET MVC Tips And Tricks".

Nota 1: pe tot parcursul cartii am folosit ca browser Mozilla Firefox si IIS 6, cu exceptia partii despre SEO.

Despre mine

Sunt un programator in .NET , trecut prin VB3 si ASP. Mai multe amanunte in CV la  http://www.infovalutar.ro/files/Cv.doc si pe blogul meu tehnic, http://serviciipeweb.ro/iafblog. Spre surprinderea (si placerea ) mea am fost numit MVP in C# pentru Romania pe anul 2009.

Multumiri (in ordine aleatoare)

Bogdan Maxim,http://www.bogdanmaxim.ro/ , pentru ca mi-a aratat greseli legate de poze/formatare si mi-a facut sugestii legate substanta cartii .

Mecu Sorin AKA Yoda, http://ronua.ro/CS/members/yoda/default.aspx - pentru aditionale sugestii legate de subiect.

Stefan Pirvu AKA Strofo , http://sharp-monkey.blogspot.com/ - pentru sugestii legate de formatare si de substanta cartii.

Alin Berce, http://alinberce.wordpress.com/ - pentru sugestii legate de formularea problematicii.

Catalin Gheorghiu AKA Mr.Smersh, http://itboard.ro/blogs/catalins\_blog/, pentru poze, cod si resurse.Precum si pentru o sugestie ( nefacuta) de a utiliza FxCop si a imi corecta codul ( rusine mie!)

Andrei Rinea, http://blog.andrei.rinea.ro/ , pentru detalii legate de carte.

Gabriel Enea , http://gabrielenea.blogspot.com/ , Senior Software Developer, fondator al serviciului [joobs.ro](http://www.joobs.ro/), pentru suport, review si sugestii.

Aurelian Popa, http://aurelian.ro/dasBlogCE/ , pentru continut, formatare/ si sugestii si pentru [Postfata](#Postfata) .

Timotei Dolean, http://timoteidolean.wordpress.com/, pentru sugestii legate de continut si formatare.

Tudor Turcu, http://www.turcu.name/ , pentru sugestii legate de continut.

De ce ASP.NET MVC si nu WebForms

Pentru ca nu vreau sa fiu silit sa re-testez paginile cind vreau sa optimizez site-ul.
Pentru ca imi place sa am controlul asupra codului HTML generat.

Pentru incepatorii absoluti

Pentru cei carora le place sa invete urmarind un video, urmariti [ASP.NET MVC Video Tutorials](http://www.asp.net/mvc/learn/) de la adresa <http://www.asp.net/mvc/learn/> .

Pentru cei carora le place sa invete citind, recomand tutorialul lui Scott Guthrie <http://nerddinnerbook.s3.amazonaws.com/Intro.htm> .

De asemenea v-ar ajuta <http://refcardz.dzone.com/refcardz/getting-started-aspnet-mvc-10>.

Pentru cei care au lucrat cu ASP.NET MVC

Continuati sa cititi ;-)

Primul site cu ASP.NET MVC

Asigurati-va ca aveti:

PC cu Windows >= XP . Instalati pe el IIS ( control panel, add/remove programs, windows components) . Asigurati-va ca aveti cd-ul original cu XP.

.NET framework . Instalati-va toate update-urile pentru Windows XP (SP-uri) si .NET framework se va instala automat. Sau (nerecomandat) puteti instala .NET 3.5 SP1 de la http://msdn.microsoft.com/en-us/netframework/default.aspx.

ASP.NET MVC . Downloadati de la http://www.asp.net/mvc/ si instalati.

Cum faceti primul vostru site cu ASP.NET MVC integrat in IIS :

1. Hotariti-va unde va fi site-ul. De exemplu structura mea pentru un proiect, sa zicem Devin , este un folder Devin in care am
	* + - Docs
			- Application
			- UnitTests
2. Porniti Visual Studio ( sau Visual Web Developer) , aveti template-ul de ASP.NET MVC Web Application si salvati-l in folder-ul respectiv(Devin)
3. Adaugati un virtual dir in IIS ( control panel, Administrative Tools , Internet Information Services, expandati pina la Default Web Site, click dreapta, New=>Virtual Directory,introduceti un alias, introduceti apoi folder-ul de la punctul anterior.
4. Inchideti Visual Studio ( sau Visual Web Developer) , deschideti din nou si File=> Open Web Site => Local IIS=> numele de la virtual dir
5. CTRL+F5 si ar trebui sa va spune "Welcome "
6. Daca va da eroare la pasul anterior , dati-mi un email la ignatandrei@yahoo.com

E adevarat ca nu aveti nevoie neaparata de IIS si ca puteti folosi server-ul din Visual Studio. Totusi, anumite probleme (cum ar fi cea de donut caching, vezi mai jos) nu apar in server-ul de Visual Studio.

Probleme comune:

Nu merge pe Windows XP sau Windows 2003 (IIS 5.1 /IIS 6  )

Rezolvare : Mapati toate extensiile (.\*) la  asp.net Mergeti la control panel => administrative tools => Internet Information Services, click dreapta pe site, properties, tab "virtual directory", configuration, edit pe ascx, copiati ce executabil este , de obicei C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet\_isapi.dll, cancel, add, paste la executabil, iar la extension puneti .\* si debifati "check that file exists" , ca in poza alaturata  :

Copiez site-ul la hosting si obtin : Runtime Error - could not load file or assembly System.Web.Mvc.dll)

Asigurati-va ca aveti .NET 3.5 ( contactati hostingul ) si ca aveti Copy local la true pentru System.Web.Mvc ca in figura :

 Mai puteti citi http://haacked.com/archive/2008/11/03/bin-deploy-aspnetmvc.aspx

Nu imi vine valoarea parametrului din actiunea controllerului!

Asigurati-va ca parametrul se cheama la fel ca in routes.MapRoute .
De exemplu, aveti structura default  :

Daca puneti un controller ProductController si scrieti actiunea :

routes.MapRoute("Default",// Route name

                "{controller}/{action}/{id}",// URL with parameters

                new { controller = "Home", action = "Index", id = "" }  // defaults

            );

parametrul ProductName nu o sa contina nici o valoare pentru linkul : http://<site>/product/view/100

 public ActionResult View(string ProductName)

Ce trebuie sa scrieti este :

public ActionResult View(string id) // cel de la "{controller}/{action}/{id}"

Nu obtin nici o valoare in parametrul FormCollection !

Verificati ca aveti Html.BeginForm.
De asemenea verificati ,in cazul controalelor, ca aveti si atributul name, nu doar id, definit pe controlul HTML.
De exemplu, pentru ca

<select id="ddlAnswer">

<option value="0">No</option>

<option value="1">Yes</option>

</select>

sa intoarca o valoare in FormCollection["ddlAnswer"], trebuie sa ii puneti atributul name :

<select id="ddlAnswer"***name***="ddlAnswer">

Cum verifici ca actioneaza ErrorHandler

Puteti face o actiune in Home, numita GenerateError, in care sa treceti throw new ArgumentException("verify error")

Daca nu vi se activeaza, verificati in Web.config daca aveti la customErrors mode="RemoteOnly" sau mode="On" (eu recomand RemoteOnly ):

        [HandleError()]

        public class HomeController : Controller

        {

            public ActionResult error()

            {

                //return View("Error", new HandleErrorInfo(new ArgumentException("asd"), "home", "error" ));

                throw new ArgumentException("error  ");

            }

        }

<customErrors mode="RemoteOnly" >

</customErrors>

Caractere speciale in adresa

Am avut o problema aparent minora cu ASP.NET MVC . Aveam un path de forma /client/view//client/view/http://localhost/<numevirtualdir>/client/view/<nume client>

Foarte bine si frumos – dar stiati ca nu accepta caractere ciudate in path ( de exemplu, ampersant :, A&D Servicii SRL) . Eroarea este :

This error (HTTP 400 Bad Request) means that Internet Explorer was able to connect to the web server, but the webpage could not be found because of a problem with the address.

For more information about HTTP errors, see Help.

Am cochetat cu ideea sa schimb denumirea – si sa pun codul lor – dar supriza : codul era non-numeric : A&DS …

Am inlocuit, fara sa ma gindesc prea mult, & cu &amp; - aceeasi eroare, normal!

In cele din urma, dupa cautari amarnice( 2 ore…) pe internet, am dat de un fisier .reg cu 2 rinduri – sper sa va fie de folos :

Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\ASP.NET]

"VerificationCompatibility"=dword:00000001

Best practices

Nu folositi ViewData - nici macar pentru a trimite datele intre master si view-uri

Motivatie minora : Nu aveti Intellisense pentru ViewData
Motivatie majora : Orice schimbare a textului in controller va poate strica functionarea site-ului - chiar si schimbarea de o litera in numele variabilei ( de exemplu, ViewData["Prenume"] versus ViewData["Prenumele"]  )
In loc de ViewData, creati-va o clasa , ca in acest exemplu pentru master :

namespace your\_Namespace

{

    public class DataMasterModel

    {

        public DataMasterModel()

        {

            User = HttpContext.Current.User.Identity.Name;

        }

        publicstring User;

    }

}

iar in master :

si undeva mai jos :

<%@MasterLanguage="C#"Inherits="System.Web.Mvc.ViewMasterPage<your\_Namespace.DataMasterModel>"%>

<%=ViewData.Model.User %>

Observatie 1: atentie la erori - view-ul de erori e derivat din acelasi master . Cel mai bine faceti un alt master pentru erori

Observatie 2: Orice view va avea modelul derivat din clasa de master:

 public class DataExcelModel : DataMasterModel

    {

    }

Folositi testarea automata

Pentru testarea automata a View-urilor puteti folosi un engine de Mock (cu ObjectFactory pentru initializare) - iar pentru testarea paginilor in site puteti folosi [Selenium](http://seleniumhq.org/).
Pentru [RhinoMock](http://www.ayende.com/projects/rhino-mocks.aspx) am folosit codul lui Scott Hanselman , <http://www.hanselman.com/blog/ASPNETMVCSessionAtMix08TDDAndMvcMockHelpers.aspx>
Codul arata cam asa:

//in controller :

 public ActionResult ExportDate(string id)

{

    IDateExport fe = ObjectFactory.GetInstance<IDateExport>();

    fe.id = id;

    export exp = new export(Server.MapPath("~/bin/Templates"));

    FileContentResult fcr = new FileContentResult(exp.Export(fe), "application/ms-word");

    fcr.FileDownloadName = fe.Number + ".doc";

    return fcr;

}

Pentru a testa paginile site-urilor am folosit  [selenium](http://seleniumhq.org/),www. seleniumhq.org (downloadati selenium-server-1.0.1 , rulati java -jar selenium-server.jar , adaugati referinta la ThoughtWorks.Selenium.Core ) si puteti incepe sa scrieti asa ceva:

//in test :

exportController i = new exportController();

MockRepository mocks = new MockRepository();

using (mocks.Record())

{

//MvcMockHelpers.SetFakeControllerContext(mocks, i);

    mocks.SetFakeControllerContext(i);

                    SetupResult.For(i.ControllerContext.HttpContext.Server.MapPath(null)).IgnoreArguments().Return(@"c:\program\templates");      // cod pentru chemarea Server.MapPath

    mocks.ReplayAll();

}

using (mocks.Playback())

{

    ObjectFactory.Initialize(x =>

    {

        x.ForRequestedType<IExport>().TheDefaultIsConcreteType<FactFind>();// FactFind nu atinge BD

     });

     FileContentResult fcr = i.exportdate("865", "A") as FileContentResult;

     fcr.ShouldNotBeNull();

     fcr.FileContents.Length.ShouldBeGreaterThan(0);

}

[TestFixture]

    public classTestWeb

    {

        private ISelenium selenium;

        private StringBuilder verificationErrors;

        [SetUp]

        public void SetupTest()

        {

            //de verificat ca e pornit server-ul java -jar selenium-server.jar

            //firefox

            //selenium = new DefaultSelenium("localhost", 4444, @"\*firefox c:\Program Files\Mozilla Firefox\firefox.exe","http://localhost/");

            selenium = new DefaultSelenium("localhost", 4444, @"\*iexplore", "http://localhost/");

            selenium.Start();

            verificationErrors = new StringBuilder();

        }

        [TearDown]

        public void TeardownTest()

        {

            try

            {

                selenium.Stop();

            }

            catch (Exception)

            {

                // ignora erorile la oprire

            }

            Console.WriteLine(verificationErrors.ToString());

            Assert.AreEqual("", verificationErrors.ToString());

        }

        [Test]

        [Category("WEB")]

        public void InvoiceFind()

        {

            selenium.Open("/siteulmeu/");

            selenium.Focus("");

            selenium.WindowMaximize();

            selenium.Type("txtName", "Andrei Ignat");

            selenium.Click("search");

            selenium.WaitForPageToLoad("5000");

            try

            {

                //daca lucrati cu firefox , se poate captura pagina

                //selenium.CaptureEntirePageScreenshot(@"C:\a.jpg","");

                //daca lucrati cu firefox , se poate captura desktopul

                //selenium.CaptureScreenshot(@"C:\a.jpg");

            }

            catch (AssertionException e)

            {

                verificationErrors.Append(e.Message);

                Console.WriteLine(e.Message);

            }

            try

            {

                Assert.IsTrue(selenium.IsTextPresent("Andrei Ignat"));

            }

            catch (AssertionException e)

            {

                verificationErrors.Append(e.Message);

                Console.WriteLine(e.Message);

            }

        }

Logati erorile

Derivati din errorhandler si puteti loga erorile . Eu am urmatoarea clasa ajutatoare, ce foloseste log4net :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace TargetWeb.classes

{

    public class HandleErrorAndLog : HandleErrorAttribute

    {

        private static readonly log4net.ILog \_logger = log4net.LogManager.GetLogger(System.Reflection.MethodBase.GetCurrentMethod().DeclaringType);

        public override void OnException(ExceptionContext filterContext)

        {

            if (filterContext != null && filterContext.Exception != null)

            {

                string controller = filterContext.RouteData.Values["controller"].ToString();

                string action = filterContext.RouteData.Values["action"].ToString();

                string loggerName = string.Format("Error in : {0}Controller.{1}", controller, action);

                if (\_logger.IsErrorEnabled)

                {

                    \_logger.Error(loggerName, filterContext.Exception);

                }

            }

            if (filterContext != null && filterContext.Exception == null)

            {

                if (\_logger.IsErrorEnabled)

                {

                    \_logger.Error("unknown Error in Controller");

                }

            }

            base.OnException(filterContext);

        }

    }

}

Alte best practices

http://weblogs.asp.net/rashid/archive/2009/04/01/asp-net-mvc-best-practices-part-1.aspx
http://weblogs.asp.net/rashid/archive/2009/04/03/asp-net-mvc-best-practices-part-2.aspx
http://www.lostechies.com/blogs/jimmy\_bogard/archive/2009/04/24/how-we-do-mvc.aspx
http://codeclimber.net.nz/archive/2009/04/17/how-to-improve-the-performances-of-asp.net-mvc-web-applications.aspx
http://blog.maartenballiauw.be/post/2009/05/06/More-ASPNET-MVC-Best-Practices.aspx

Interceptati eroarea "HTTP 404 - Not Found"

Vedeti discutia de la http://stackoverflow.com/questions/318886/net-mvc-routing-catchall-not-working .
In principiu batalia se da intre \*catchall

si interceptarea erorilor in Application\_Error :

routes.MapRoute("Error", "{\*catchall}", new { controller = "Base", action = "Error", id = "404" });

 protected void Application\_Error(object sender,EventArgs e)

{

    Exception exception =Server.GetLastError();

    HttpException httpException = exception as HttpException;

    if(httpException !=null)

    {

        RouteData routeData =newRouteData();

        routeData.Values.Add("controller","Error");

        routeData.Values.Add("action","HttpError500");

        if(httpException !=null)

        {

            if(httpException.GetHttpCode()==404)

            {

                routeData.Values["action"]="HttpError404";

            }

        }

        Server.ClearError();

        Response.Clear();

        IController errorController =newErrorController();

        errorController.Execute(newRequestContext(newHttpContextWrapper(Context), routeData));

    }

}

Nu folositi output caching – ci inmemorycache

De ce?  Pentru ca, daca vreti sa faceti caching la tot view--ul(vezi Donut Caching, http://haacked.com/archive/2008/11/05/donut-caching-in-asp.net-mvc.aspx) si vreti sa si comprimati si sa faceti si zip - o sa va dea , in loc de html, niste coduri absconse.
Puteti verifica la adresele :
http://infovalutar.ro/example/index  -daca dati refresh, se schimba toate cele 3 date
http://infovalutar.ro/example/inmemorycachetest -daca dati refresh, se schimba doar data din inmemorycache
http://infovalutar.ro/example/outputcachetest -apar gindacei

InMemorycache este luat de la http://stackoverflow.com/questions/343899/how-to-cache-data-in-a-mvc-application

Codul din controler este urmatorul :

Codul din view este mai simplu :

 public class exampleController : Controller

        {

            public ActionResult Index()

            {

                return View("Index", newViewDate());

            }

            **[CompressFilter(Order = 1)]**

            [WhitespaceFilter(Order = 2)]

            [OutputCache(Order = 3)]

            public ActionResult OutputCacheTest()

            {

                return View("Index", newViewDate());

            }

            **[CompressFilter(Order = 1)]**

            [WhitespaceFilter(Order = 2)]

            public ActionResult InMemoryCacheTest()

            {

                InMemoryCache c = new InMemoryCache();

                ViewDate v = c.Get<ViewDate>("cache\_viewdate", () => { returnnewViewDate(); });

                return View("Index", v);

            }

        }

Concluzia : Nu folositi outputcaching !

<%@PageTitle=""Language="C#"MasterPageFile="~/Views/Shared/Site.Master"Inherits="System.Web.Mvc.ViewPage<InfovalNew.Controllers.ViewDate>"%>

<%@ImportNamespace="InfovalNew.Controllers"%>

<asp:ContentID="Content3"ContentPlaceHolderID="MainContent"runat="server">

    <h2>View Date Example</h2>

    <br/>

<h1>Date from Controller : <%=ViewData.Model.date.ToString("yyyyMMMMdd\_HHmmss") %></h1>

<br/>

<h1>Current date in HTML : <%=DateTime.Now.ToString("yyyyMMMMdd\_HHmmss")%></h1>

<br/>

<h1>Donut caching : <%= Html.Substitute(c => DateTime.Now.ToString("yyyyMMMMdd\_HHmmss"))%></h1>

</asp:Content>

<asp:ContentID="Content4"ContentPlaceHolderID="title"runat="server">View Date Example</asp:Content>

<asp:ContentID="Content5"ContentPlaceHolderID="metaname"runat="server">

</asp:Content>

Accesorii

     Verificarea rutelor

E usor sa verificati rutele - daca downloadati de la http://haacked.com/archive/2008/03/13/url-routing-debugger.aspx dll-ul.Adaugati referinta la el,si adaugati

    RouteDebug.RouteDebugger.RewriteRoutesForTesting(RouteTable.Routes);

De exemplu acesta este un standard Application\_Start la mine

Testarea actiunii controlerelor

 protected void Application\_Start(object sender, EventArgs e)

        {

            RegisterRoutes(RouteTable.Routes);

            RouteDebug.RouteDebugger.RewriteRoutesForTesting(RouteTable.Routes);

            System.IO.FileInfo s = new System.IO.FileInfo(Server.MapPath("~/log4net.config"));

            log4net.Config.XmlConfigurator.ConfigureAndWatch(s);

        }

Dependency Injection ne ajuta sa testam automat codul -fara sa atingem alte componente( ca de exemplu BD). Gasiti o lista la adresa http://en.wikipedia.org/wiki/Dependency\_injection#Existing\_frameworks , dar eu am utilizat StructureMap, http://structuremap.sourceforge.net/Default.htm.
L-am folosit cind aveam de exportat niste date in format Excel si vroiam doar sa obtin datele cu StringTemplate,fara sa mai ating BD.

unde variabila fe din

public ActionResult ExportDate(string id)

        {

            DateExport fe = new DateExport();

            fe.id = id;

            export exp = new export(Server.MapPath("~/bin/Templates"));

            FileContentResult fcr = new FileContentResult(exp.Export(fe), "application/ms-word");

            fcr.FileDownloadName = fe.Number + ".doc";

            return fcr;

        }

DateExport fe = new DateExport();

atingea BD in momentul in care cineva ii cerea niste date.

Ca sa fac acest lucru, am extras metodele care ma interesau din DateExport, am facut o interfata din ele IDateExport, am inlocuit parametrul din functia export ce cerea un DateExport cu IDateExport si am folosit StructureMap :

In global.asax am definit cererea default :

ObjectFactory.Initialize(x =>

            {

                x.ForRequestedType<IDateExport>().TheDefault.Is.ConstructedBy(() => new DateExport(HttpContext.Current.User.Identity.Name));

            });

si apoi in metoda am inlocuit new cu ObjectFactory :

public ActionResult ExportDate(string id)

        {

            IDateExport fe = **ObjectFactory.GetInstance<IDateExport>()**;

            fe.id = id;

            export exp = new export(Server.MapPath("~/bin/Templates"));

            FileContentResult fcr = new FileContentResult(exp.Export(fe), "application/ms-word");

            fcr.FileDownloadName = fe.Number + ".doc";

            return fcr;

        }

Acum testul automat a devenit floare la ureche – sa zicem ca vreau ca metodele din IDataExport sa nu aduca nimic –si sa testez acest lucru . Creez o clasa DataExportNotFind, implementez IDataExport ca sa nu aduca nimic si scriu in test :

ObjectFactory.Initialize(x =>
{
    x.ForRequestedType<IDataExport>().TheDefaultIsConcreteType<DataExportNotFind>();
});

Transmiterea datelor intre BusinessLogic si Controllere/View

De obicei aveti de luat datele din DAL si pus in Repository ( BLL ?). La fel , cind afisati datele in GUI , aveti o clasa care corespunde "in mare" cu tabela respectiva din BD + inca citeva date.
AutoMapper, http://www.codeplex.com/AutoMapper , poate face acest lucru semi-automat, inclusiv pentru List<T>.
Sa zicem ca avem in baza de date avem Client cu an, luna, zi - si o clasa Linq/EF - iar in GUI vrem sa afisam ca DataClient care are un cimp data -caci ne este mai usor la .ToString(""). In rest variabilele sunt aceleasi.

De cite ori mai adaugam un cimp in DataClient , si il adaugam IDENTIC in Client, AutoMapper-ul se ocupa sa transfere automat.

Mapper.CreateMap<DataClient, Client>()

                    .ForMember(tx => tx.Data,

                    src => src.MapFrom(z => new DateTime(z.an, z.luna, z.zi)));

            Client c = Mapper.Map<DataClient, Client>(ClientRepository.GetClient(id));

Cum sa faceti footer la un grid - sau sa adaugati alte rinduri

In primul rind, trebuie sa spun ca folosesc HTml.Grid de la MvcContrib, http://www.codeplex.com/MVCContrib . Acesta este super la afisarea dateor si merge super ok.
Problema care am avut-o este sa afisez un total la o factura.
In primul rind trebuie sa extind Grid-ul si sa ii adaug o functie de footer:

Apoi FooterHtmlGrid e simplut:

public class HtmlGridFooter<T> : Grid<T>

        where T : class

{

    public HtmlGridFooter(IEnumerable<T> dataSource, TextWriter writer, ViewContext context) :

        base(dataSource, writer, context)

    {

    }

    public Func<IEnumerable<T>, string> footer

    {

        set

        {

            this.RenderUsing(new FooterHtmlGrid<T>() { actions = base.DataSource, footer = value });

        }

    }

}

public class FooterHtmlGrid<T> : HtmlTableGridRenderer<T>

        where T : class

{

    public IEnumerable<T> actions;

    public Func<IEnumerable<T>, string> footer;

    protected override void RenderGridEnd(bool isEmpty)

    {

        if (footer != null)

        {

            base.RenderText(footer(actions));

        }

        base.RenderGridEnd(isEmpty);

    }

}

Apoi am adaugat un extension method , de genul :

Si, in sfirsit, am putut scrie codul usor :

public static IGrid<T> HtmlGridFooter<T>(this HtmlHelper helper, IEnumerable<T> dataSource, Func<IEnumerable<T>, string> footer)

        where T : class

        {

            HtmlGridFooter<T> g = new HtmlGridFooter<T>(dataSource, helper.ViewContext.HttpContext.Response.Output, helper.ViewContext);

            g.footer = footer;

            return g;

        }

      <%=Html.HtmlGridFooter<Lines>(ViewData.Model.Lines,

        (x) =>

        {

            decimal total = 0;

            foreach (var s in x) { total += s.Total; };

            return"<tr><td colspan=1 align=right>Total</td><td colspan=1>"+ total.ToString("#.00")+"</td></tr>";

        }

  )

               .Columns(col =>

               {

col.For(line => line.LineNumber).Named("Number");

            col.For(line => line.Total).Named("Total");

                }

)

%>

      Designul site-ului

Daca nu aveti un designer bun , atunci puteti downloada exemplele de la http://www.asp.net/mvc/gallery/ . Cel mai bine este sa o faceti de la inceput.
Daca va da “The type or namespace name 'Helpers' could not be found (are you missing a using directive or an assembly reference?)” atunci e cazul sa dati “Show all files” si pe urma sa includeti fisierele lipsa . Recompilati. Rulati.Vedeti referintele pentru CSS.

      Cum sa aveti un sitemap

Daca aveti nevoie de SiteMap, va recomand http://mvcsitemap.codeplex.com/ . Va las cu explicatiile de pe site - sunt complete si usoare.

Optimizarea   aplicatiei

    Accesul la baza de date

    Pentru a optimiza accesul la baze de date, incarcati doar datele de care aveti nevoie( de ex., daca afisati doar nume si prenume, nu incarcati si data de nastere). Sfatul obisnuit : nu faceti select \*
    Puteti , de asemenea, incarca dictionarele( de ex., lista de judete) intr-o variabila globala (in application) la Application\_Start.

    Code protection from decompilers

Aveti posibilitatea sa alegeti Community Dotfuscator Edition- gratuit in VS2008- sau puteti alege SmartAssembly(din pacate, nu e gratis).
Sunt destul de incantat cu SmartAssembly, deoarece este mult mai prietenos, stie sa faca si merge la assembly şi aveţi posibilitatea să se integreze în procesul de MSBuild - si doar cind e Release, nu si pe debug - ca sa nu fiti penalizati la viteza:

 <UsingTask TaskName="SmartAssembly.MSBuild.Tasks.Build" AssemblyName="SmartAssembly.MSBuild.Tasks,&#xD;&#xA;Version=4.0.0.0, Culture=neutral, PublicKeyToken=cd3409ee69028647" />
  <Target Name="BeforeBuild" Condition=" '$(Configuration)' == 'Release' ">
    <CreateProperty Value="true">
      <Output TaskParameter="Value" PropertyName="RunSmartAssembly"/>
    </CreateProperty>
  </Target>
  <Target Name="AfterBuild" Condition=" '$(RunSmartAssembly)' != '' ">
    <SmartAssembly.MSBuild.Tasks.Build ProjectFile="readcur.{sa}proj" OverwriteAssembly="false" MarkAsReleased="false" />

  </Target>

ilmerge si web deployment projects

Pentru un deployment mai usor, puteti folosi web deployment projects, http://www.microsoft.com/downloads/details.aspx?FamilyId=0AA30AE8-C73B-4BDD-BB1B-FE697256C459&displaylang=en . Nu merge decit cu VS 2008 , nu express.Ceea ce face este sa compileze site-ul si, la nevoie, sa inlocuiasca portiuni din Web.Config( de ex.,partea cu connection string sa o schimbe - nu vreti pe serverul de productie/hosting sa fie cel de pe masina locala).

Detalii http://msdn.microsoft.com/en-us/library/aa479568.aspx si http://weblogs.asp.net/scottgu/archive/2005/11/06/429723.aspx

De asemenea, are inclus si ILMerge,http://research.microsoft.com/en-us/people/mbarnett/ILMerge.aspx,  - care face "merge" la mai multe assembly-uri.

  Ajax with jquery

JQuery, http://jquery.com/ , este o librarie/framework de JavaScript care face mai usoara utilizarea JavaScript/HTML DOM/evenimente (bineinteles, dupa curba de invatare…)

Pentru a nu face postback de fiecare data, puteti folosi ajax cu jquery ca sa fie mai usor. Sa zicem ca avem de facut ceva simplu, de genul stergere de rind . Mai intii scriem in controller functia Delete :

Acum scriem codul pe pagina HTML :

       **[AcceptVerbs(HttpVerbs.Post)]** //de pus neaparat, ca nu cumva un bot de indexare sa ne stearga din greseala cind executa un get / o cerere simpla
**[Authorize()]** // daca vrem sa identificam user-ul

public JsonResult Delete(string ID)

        {

            try

            {

                ClientRepository c = new ClientRepository();

                c.Delete(long.Parse(ID));

                return Json(new { ok=true,mesaj = " Sters !" }); //TODO : poate sa fie schimbat mesajul dupa limba user-ului

            }

            catch (MyCustomException ex)

            {

                return Json(new { ok = false, mesaj = "Eroare de aplicatie :" + ex.Message });

            }

            catch (Exception ex)

            {

                return Json(new { ok = false, mesaj = "Eroare de sistem :" + ex.Message });

            }

        }

 function DeleteClient(id) {

$.ajax({

            type: "POST",

            url: "/Controller/Delete/" + id,

            data: "",

            contentType: "application/json; charset=utf-8",

            dataType: "json",

            success: function(msg) {

                window.alert(msg.mesaj);

                if (msg.ok) {

                    $('#client\_' + id).remove(); // il inlaturam din HTML DOM

                }

            },

            error: function(xmlhttprequest, textstatus, errorthrown) {//ajunge aici doar daca e o problema de conexiune la site

                window.alert('Error :' + xmlhttprequest.responseText);

            }

        });
        }

Microsoft vine si el puternic din urma cu ASP.NET 4.0 AJAX , dar deocamdata este preview : http://aspnet.codeplex.com/Wiki/View.aspx?title=AJAX

Optimizarea HTML

    Expires header pentru imagini

Folositi un ashx - sau o ruta. Eu prefer ruta.
In pagina HTML :

<img src="/content/images/ya.gif.ashx"alt="yahoo"style="border-width: 0"width="64"height="16"/>

In ruta :

routes.MapRoute("content", "content/images/{id}.ashx",

                new { controller = "imagini", action = "content" });

In controller-ul "imagini" :

        [CacheFilter(Days = 60)]

        publicvoid content(string id)

        {

            string file = Server.MapPath(string.Format("/content/images/{0}",id));

            string filename = Path.GetFileName(file);

                Response.ContentType = "image";

            Response.AddHeader("content-disposition", "inline; filename=" + filename);

            Response.WriteFile(file);

         }

  Minimizarea html

Pentru aceasta folosesc pe actiune :

Compress list il luati de la http://www.58bits.com/blog/2009/05/02/GZip-And-Deflate-Compression-Filter-For-ASPNet-MVC.aspx

Whitespacefilter il luati de la http://www.codeproject.com/KB/aspnet/WhitespaceFilter.aspx - cu mentiunea ca ar trebui rescrise proprietatile facind apel la variabila dinauntru.
De exemplu :

        [CompressFilter(Order = 1)]
   [WhitespaceFilter(Order = 2)]
   [HandleErrorAndLog]

Public Overrides ReadOnly Property Length() AsLong
Get
Return 0
End Get
End Property

O rescrieti asa:

Public Overrides ReadOnly Property Length() AsLong
Get
Return \_Sink.Length
End Get
End Property

Pentru C#(puteti translata codul din VB.NET in C# cu http://www.carlosag.net/Tools/CodeTranslator/ )

Exemplu de functionare il vedeti la http://www.infovalutar.ro/bnr - daca va uitati la sursa este gzip + compresata.

public override long Length

        {

            get { return \_sink.Length; }

        }

Minimizare incarcare  javascript

1. Puteti folosi Google ( si mai nou, Microsoft) va lasa sa gazduiti la ei jquery - cu avantajul de a "trage" jquery de pe cel mai apropiat server(vezi CDN, http://en.wikipedia.org/wiki/Content\_Delivery\_Network) . E clar ca nu o fac DOAR pentru ca iubesc dezvoltatorii -si ca au interesele lor - dar, daca vreti sa scapati de citiva KB in plus de pe site( inmultit cu citiv vizitatori aveti), atunci puteti folosi Google ca aici  http://code.google.com/apis/ajaxlibs/documentation/index.html#jquery sau Microsoft ca aici <http://www.asp.net/ajax/cdn/#2>
2. Daca vreti sa va minimizati propriul jscript, atunci puteti face creea un handler , prin care sa minimizati(si  gzip )  - ca la imagini

La mine handler-ul se activeaza asa :
In codul sursa am pus :

<script src="/scripts/index.js.ashx"type="text/javascript"></script>

Ca sa interceptez chemarea , am configurat o ruta

 routes.MapRoute("scripts", "scripts/{id}.ashx",

                new { controller = "minimy", action = "scripts" });

Pe ruta respectiva am pus:

Observatie : atentie la comentarii - un comentariu pus in sursa neminimizata face ca sa nu se mai "vada " nimic minimizat, ba, in plus, da erori . Exemplu :

        [CompressFilter(Order = 1)]
        [CacheFilter(Days = 60, Order = 3)]

public void scripts(string id)

        {

string filename="scripts/" + id;

                Response.AddHeader("content-disposition", "inline; filename=" + filename);

                Response.WriteFile( Server.MapPath(filename));

        }

ajunge minimizata la

        function v(id)
{

//id e id-ul clientului

window.alert(id);
}

       function v(id) {  //id e id-ul clientuluiwindow.alert(id);}

Si imediat va da eroare, pentru ca functia NU se mai termina( nu exista acolada de sfirsit)

1. multiple jscript in one : fie folosit un ScriptManager, ca in "Using Script Combining to improve AJAX performance", http://www.asp.net/learn/3.5-SP1/video-296.aspx  fie folositi "HTTP handler to combine multiple files, cache and deliver compressed output for faster page load ", ca la http://msmvps.com/blogs/omar/archive/2008/08/28/http-handler-to-combine-multiple-files-cache-and-deliver-compressed-output-for-faster-page-load.aspx. Puteti sa combinati cu punctul anterior.

Interfata grafica asemanatoare in diferite browsere

Ca sa puteti avea aceeasi interfata in toate browserele (acum , in 2009, IE6, IE7, FF3.0 si FF3.5 , cu Chrome care vine puternic din urma) v-as sugera sa porniti de la un CSS de reset - de exemplu de la http://developer.yahoo.com/yui/reset/ .
Pentru mai multe detalii cautati dupa "CSS frameworks" - o sa gasiti http://developer.yahoo.com/yui/3/cssbase/ , http://www.blueprintcss.org si altele.

   Search Engine Optimization

Daca vreti sa faceti si un pic de SEO, downloadati pentru IIS 7 un programel de SEO ce evidentiaza eventualele erori de la http://www.microsoft.com/downloads/details.aspx?FamilyID=D8C32F95-2D50-43A1-99B9-FA8E5BBFC34C&displaylang=en
Odata instalat, la selectarea unui site din IIS va va aparea :

Rulati si o sa va mirati cite reguli puteti incalca.

Puteti folosi acest tool si pentru site-uri de pe internet …

      Additional helpers

a.      Firefox + FireBug + Yahoo Slow

       Iti arata cum se incarca site-ul  - si potentialele probleme. In plus , face si un scor si statistici - cit se downloadeaza la prima intrare pe site si cit dupa aceea

b.      Firefox + FireBug + Google PageSpeed

            Google Page Speed iti arata in plus de Yahoo Slow inca citeva chestii care pot fi optimizate( cookie size , de ex.) si in plus header-ele cererilor

Din poza alatura vedeti ca site-ul are "Content-encoding" la gzip - ceea ce e bine...

c.      Firefox cu Web developer

        Cu ajutorul WebDeveloper putem modifica pagina si vedea eventuale inadvertente din pagina - de ex., putem vedea care imagini nu au height si width(conteaza la afisarea initiala).

d.      Fiddler

    Fiddler poate intercepta traficul din IE si iti poate arata o mare varietate de informatie despre cererile facute.

e. Html Validator - valideaza corectitudinea codului HTML din punct de vedere a W3C standard.

f. Browsercam, , <http://www.browsercam.com/Default2.aspx>

        Browsercam iti poate arata cum arata site-ul tau in diferite browser-e.

Ce urmeaza

Asp.NET MVC se dezvolta , in urma feedbackurilor venite . Un anunt gasiti aici: http://haacked.com/archive/2009/10/01/asp.net-mvc-preview-2-released.aspx

Va sfatuiesc sa urmariti dezvoltarea lui !

Linkuri utile care nu au incaput

Manager ASP.NET MVC , http://haacked.com/tags/aspnetmvc/default.aspx

Creator ASP.NET MVC , http://weblogs.asp.net/scottgu/

Alte tips and tricks , http://stephenwalther.com/blog/category/4.aspx?Show=All

ASP.NET MVC T4 No magic Strings :
                     Autor :  http://blogs.msdn.com/davidebb/archive/2009/06/26/the-mvc-t4-template-is-now-up-on-codeplex-and-it-does-change-your-code-a-bit.aspx
                    Exemplu : http://www.hanselman.com/blog/TheWeeklySourceCode43ASPNETMVCAndT4AndNerdDinner.aspx

SEO tips , http://www.google.co.uk/intl/en/landing/conversion/ebook.html

Asp.NET MVC View Engine, de exemplu http://codebetter.com/blogs/jeffrey.palermo/archive/2008/01/27/mvccontrib-now-offers-four-4-alternative-view-engines-for-asp-net-mvc.aspx

Tool-uri mentionate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Name | Download | Observatii | Free |
| 1 | log4net | http://logging.apache.org/log4net/index.html | Pentru logare | Da |
| 2 | RhinoMock | http://www.ayende.com/projects/rhino-mocks.aspx. | Pentru testare mock | Da |
| 3 | Selenium | http://seleniumhq.org/ | Testare pagini web | Da |
| 4 | RouteDebug | http://haacked.com/archive/2008/03/13/url-routing-debugger.aspx | Testare rute cu ASP.NET MVC | Da |
| 5 | NUnit | http://www.nunit.org/index.php | Testare | Da |
| 6 | SmartAssembly | http://www.smartassembly.com/ | Obfuscarea codului | Nu |
| 7 | Web Deployment project | http://www.microsoft.com/downloads/details.aspx?FamilyId=0AA30AE8-C73B-4BDD-BB1B-FE697256C459&displaylang=en | Pentru deployment pe server | DA(totusi, VS2008 se plateste)Nu merge pe VS Express |
| 8 | Jquery | http://jquery.com/ | Pentru Ajax si usurinta scrierii pe HTML client | Da |
| 9 | MvcContrib | http://www.codeplex.com/MVCContrib | Pentru usurinta grid-ului si a extensiilor | Da |
| 10 | StructureMap | http://structuremap.sourceforge.net/Default.htm | Testare | Da |
| 11 | AutoMapper | http://www.codeplex.com/AutoMapper | Transfer de date intre obiecte | Da |
| 11 | StringTemplate | http://www.stringtemplate.org/ | Export date in formatul specificat.Vedeti si http://www.codeplex.com/exporter | Da |
| 12 | Yahoo Slow | http://developer.yahoo.com/yslow/ | Evaluare siteului dpdv incarcare in browser.Are nevoie de Firefox + Firebug | Da |
| 13 | Google PageSpeed | http://code.google.com/speed/page-speed/ | Evaluare siteului dpdv incarcare in browser.Are nevoie de Firefox + Firebug | Da |
| 14 | Web developer | https://addons.mozilla.org/en-US/firefox/addon/60 | Vizualizare caracteristici DOM pagina(imagini, CSS, cookie, etc) Are nevoie de Firefox | Da |
| 15 | Html Validator | https://addons.mozilla.org/en-US/firefox/addon/249 | Valideaza cu HTML Tidy .Are nevoie de Firefox | Da |
| 16 | IIS Search Engine Optimization (SEO) Toolkit 1.0 Beta | http://www.microsoft.com/downloads/details.aspx?FamilyID=D8C32F95-2D50-43A1-99B9-FA8E5BBFC34C&displaylang=en | Evidentiaza citeva din problemele de SEODoar pe IIS 7 | Da |
| 17 | Yahoo CSS Reset | http://developer.yahoo.com/yui/3/cssreset/ | Acelasi look and feel in diferite browsere | Da |

# Postfata (Aurelian Popa)

Scott Guthrie și Mark Anders au codat în Java (da, Java) prototipul ASP.NET (denumit la vremea aceea XSP) în 2 luni in timpul vacanței de iarnă a lui 1997-1998.

După îndelungi dezbateri, evaluări și repoziționări ale prototipului, s-a luat decizia de a crea o nouă platformă peste CLR, platformă web care de data asta avea să fie implementată în C# și redenumită ASP+, apoi ASPX și în final ASP.NET.

XSP, cum era denumit inițial a fost prezentat publicului larg pentru prima dată în 2 mai 2000 la conferința ASP Connection din Phoenix Arizona împreună cu tot .Net Framework-ul v1.0.

ASP.NET a fost anunțat oficial în 11 iulie 2001 la conferința PDC din Orlando, împreună cu VB.NET noua versiune ce moștenea VB6-ele. VB6-iștii dezvoltau așa-numitele WinForms. Practic scenariu clasic era că dezvoltatorul făcea drag-and-drop ale controalelor din Toolbar-ul VS-ului și apoi se codau evenimentele acelor controale WinForms.

Și am ajuns la marea cauză care a făcut din ASP.NET Webforms ceea ce este el azi: exista o bază de programatori consistentă pe care Microsoft intenționa să-i migreze ușor către noul lor copilaș: .Net-ul.

În același timp, dezvoltatorii ASP-ului simplu își dezvoltau aplicațiile web într-un limbaj similar VB6, VBScript-ul. E adevărat că ASP-ul suporta și JScript, dar soarta a făcut ca 99% din aplicațiile ASP să fie scrise în VBScript (mult mai prietenos începătorilor dar extrem de non-OOP, din nefericire).

Deh nu le poți avea pe amândouă, ca limbaj, se pare. Desigur că în acea paradigmă a ASP-ului, atât codul client side (HTML) cât și codul server side (<%...%>) era stocat în același fișier sursă

Practic ASP.NET-ul alerga astfel după doi iepuri deodată:

* eliminarea spagetti-code-ului specific ASP-ului
* păstrarea și fidelizarea dezvoltatorilor VB6/VBScript obișnuiți cu WinForm-urile statefull

ASP.NET WebForms a rezolvat prima problemă cu așa-numita facilitate numită code behind, dar rămânea problema similarității cu Winforms. Astfel că s-au inventat așa zisele ASP.NET Server Controls care, colac-peste-pupăză se dorea a funcționa peste un protocol stateless: HTTP-ul.

Și așa s-a născut cel mai mare monstru posibil din aplicațiile web ASP.NET: ViewState-ul! A mai rezultat un alt monstruleț mai mic numit: suita ASP.NET Server(side) Controls care, pe de o parte semănau cu WinForms Controls, dar (și acesta este un DAR mare) , pe de altă parte, generau HTML on-the-fly fără ca dezvoltatorul să poată controla prea mult ce fel de HTML generau acele controale.

În plus mai era o problemă: testarea, sau mai bine spus aproape imposibilitatea de a dezvolta cu focus pe testare (TDD sau test driven development) aplicațiile web.

Datorită acestei paradigme, cuplată cu nivelul scăzut al abilităților OOP al marei mase de VB-iști și a lipsei infrastructurii de testare ,rezultau pagini web greoaie, mari, dificile cu ViewState-uri imense, non-SEO-friendly care au împânzit web-ul în lung și-n lat.

Astfel că a fost nevoie de o regândire a abordării pentru partea web din .Net Framework.

Astfel a apărut ASP.NET MVC. Tot Cott a scris prototipul și apoi Phil i s-a alăturat deoarece voia MVC-ul pentru .subtext-ul lui.

Se dorea un motor de aplicații web care rezolva toate probleme amintite de mai sus:

* fără viewState (compromisul e că s-au reintrodus <%...%>-urile, dar cu probabilitate mică de a fi folosite ca spaghetti-code
* cu control 100% asupra HTML-ului emis de scriptul web server-side, control realizat prin View-uri
* SEO-friendly (datorită Router-ului) astfel că /ShowProduct.aspx?id=1 devine /Products/Ceai-Oolong-Suzhou
* Separarea preocupărilor (modelare, operații și design,- sau prezentare adică, M-ul, C-ul și V-ul)
* V-ul din ASP.NET MVC sa fie extensibil. In acest moment V-ul = WebForms, dar deja există câteva View-uri compatibile cu ASP.NET MVC:
	+ MVCContrib (Brail, NVelocity, NHaml și XSLT),
	+ String View Engine (o portare la .Net a popularului Java Templeting Engine : String Template)
	+ Spark (compatibil și cu Castle Project MonoRail)

Această nouă abordare a ASP.NET-ului va fi benefică pentru industria de web development? Numai timpul ne va lămuri.

Va avea succes? Numai comunitatea dezvoltatorilor va influența răspunsul la această întrebare.

Acum, probabil că te întrebi „Bine, bine dar eu ce trebuie să fac acum?” … Ei bine, singurul meu sfat este: încearcă să înțelegi tehnologia ASP.NET MVC, să te joci cu ea și bineînțeles să participi activ în comunitatea sau „tribul” tău cu dezvoltatori web din care faci parte.

## Nota Andrei Ignat:

Cu aceasta va urez succes in invatarea ASP.NET MVC. De ce e bine sa il stii ? Pentru ca , „daca tot ce ai e un ciocan, toate celelalte vi se vor parea cuie” - este un tool bun de avut la briu daca dezvoltati pe Internet – alaturi de altele( de exemplu, Silverlight).

Daca aveti probleme legate de cod, intrebari, observatii – va rog sa imi scrieti pe adresa ignatandrei@yahoo.com

Va multumesc ca ati avut rabdarea de a citi aceasta carte!