

Microsoft



Microsoft®
WebMatrix

Ľuboslav Lacko

Vytvárajte moderné Web 2.0 aplikácie

Vytvárajte moderné Web 2.0 aplikácie

Internet už dávno nie je výsadou skupiny počítačovo gramotných ľudí. Začal sa proces masovej socializácie webu a používatelia v súčasnosti prevzali moc nad jeho obsahom. Nielen rozvoj technológií, či rýchle a dostupné pripojenie, ale aj potreba byť na pulze života a biznisu a potreba zdieľať zážitky a názory je hybnou silou, ktorá posúva Internet k novým métam. Zároveň sa však používatelia Internetu stali náročnejšími na formu prijímaného posolstva a hlavne majú túžbu po interaktivite a možnosti ku všetkému sa vyjadriť. Samozrejme nový interaktívny web má aj iné dimenzie, používatelia si uvedomujú, že napriek tomu, že väčšina masových služieb a sociálnych sietí je bezplatná, prevádzka webovej infraštruktúry niečo stojí a tak sú za relevantnú protihodnotu ochotní prijímať aj reklamné posolstvo. Tí podnikavejší objavajú aj možnosti ako reklamné posolstvo šíriť.

Web 2.0 je súhrnný názov pre nové internetové technológie, ktoré umožňujú konzumentom webového obsahu aby sa bez špeciálnych znalostí stali jeho tvorcami. Napríklad ak je vaším motívom pre vytváranie webových stránok hobby, nemusíte byť špecialistom na informačné technológie, stačí vám poznať základy a mať k dispozícii dobré nástroje.

Internet ponúkal návštevníkom spočiatku iba jednosmernú konzumáciu obsahu z webových stránok k používateľovi. Pre Web 2.0 je charakteristický viacsmerový komunikačný proces, teda zo stránky k používateľovi, od používateľa k webovej stránke, či medzi portálmi navzájom. Do toho vstupuje aj takzvaný „Internet vecí“, teda komunikácia medzi rôznymi zariadeniami (chladnička, systém riadenia inteligentného domu...) a poskytovateľmi služieb.

Interaktivita Webu 2.0 umožnila stieranie rozdielov medzi profesionálnymi a laickými tvorcami webového obsahu.

Typickými fenoménmi webu 2.0 sú blogy, sociálne siete a wiki, prostredníctvom ktorých sa každý používateľ stáva automaticky tvorcom obsahu. Stačí pridať trochu kreativity a zvládnuť jednoduché technológie pre vytváranie webových stránok a môžete postúpiť na najvyššiu métu webu – stať sa tvorcom stránok a aplikácií.

Postupným vývojom webových aplikácií sa neustále zvyšuje ich interaktivita a kvalita prezentačného rozhrania. Tento vývoj je evolučný, medzi používateľmi a tvorcami webových aplikácií vznikol určitý druh kladnej spätnej väzby, kedy nové možnosti, ktoré poskytujú nové technológie evokujú čoraz vyššie a náročnejšie požiadavky zo strany používateľov. Tento trend vyústil do požiadavky, aby sa možnosti používateľského rozhrania webových aplikácií, čo najviac priblížili možnostiam klasických aplikácií bežiacich na desktopoch.

Dynamické, interaktívne aplikácie, ktoré umožňujú prispôsobenie užívateľmi sú základným pilierom Webu 2.0.

Úspešnosť aplikácií a to nielen webových závisí v prvom rade od spokojnosti používateľov. Platí to hlavne pri aplikáciách s ktorými sa pracuje denne. Aplikácia musí mať dobrý dizajn, príjemné

a ergonomické ovládanie a rýchlu odozvu. Pre hodnotenie spokojnosti používateľov platí jednoduchý slovný vzorec

Funkcionalita + Vzhľad + Vzťah + Príbeh = Spokojnosť používateľov.

Na rozdiel od klasických príručiek, ktoré sú koncipované od základov po najpokročilejšie témy, v príručke pre tvorbu moderných Web 2.0 Mashup aplikácií sme zvolili trochu odlišný postup. Uvidíte, že niektoré veľmi sofistikované záležitosti, ako napríklad integrovanie vyhľadávania do vašej aplikácie, či spolupráca so sociálnymi sieťami sa dá vyriešiť jednoducho a intuitívne doslova na niekoľko riadkov kódu.

Avšak prílišnému optimizmu z predošlého odseku bráni slovné spojenie „do vašej aplikácie“. Aby ste mohli prepojiť vlastnú aplikáciu s Facebookom a tak na ňu prilákať ľudí, musíte vlastnú aplikáciu mať a musí byť pre používateľa užitočná. A aby ste takú aplikáciu vytvorili, potrebujete zvládnuť prácu s obrázkami, multimédiami, ba aj s údajmi. A to je dôvod, aby ste čítali aj tie kapitoly, ktoré sa zdajú byť zložitejšie.

Táto publikácia je primárne zameraná na ľudí, zamýšľajúcich vytvoriť si vlastné stránky, pričom predpokladáme, že primárna oblasť ich záujmov je iná.

Web Matrix ako ideálny nástroj pre začiatočníkov

Ak niektorým záujemcom o vývoj webových aplikácií tento nadpis niečo pripomína, máte pravdu. Vývojové prostredie od spoločnosti Microsoft s názvom WebMatrix tu už raz bolo, konkrétne v roku 2003. Bolo to odľahčené open-source vývojové prostredie pre vývoj hobby projektov na platforme ASP.NET. Postupne ho však vytlačili voľne šíriteľné „Express“ verzie nástroja VisualStudio a samostatný produkt VisualWebDeveloper Express.

WebMatrix obsahuje v jednom softvérovom balíku všetko, čo potrebujete na vytváranie webových aplikácií na platforme Windows, vrátane webového servera **IIS Express**, webového frameworku, **ASP.NET** a embedded databázy **SQL Server Compact** (embedded databáza). Všetko je zadarmo a open-source. Skúsenosti a prípadne aj osvedčené aplikácie vytvorené v tomto prostredí môžete využívať aj na profesionálnych komerčných platformách **VisualStudio** a **SQL Server**.

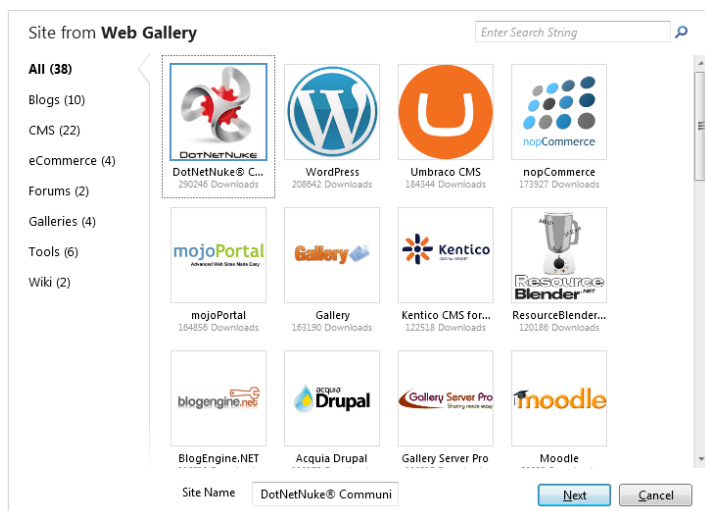
Produkt nájdete na adrese <http://www.microsoft.com/web/webmatrix/>.



Matrix – komplexná webová platforma

Idea nástroja WebMatrix vtedy, aj teraz po siedmich rokoch, je podobná – pritiahnúť hobby vývojárov na platformu .NET. WebMatrix je koncipovaný tak, že prvé projekty v ňom môžete vytvárať prakticky okamžite a postupne si „za pochodu“ osvojíte aj zložitejšie témy. Najvýznamnejšou novinkou je technológia **Razor** a súbor knižníc pre implementáciu širokej palety webových služieb a sociálnych sietí.

Vývojové prostredie Web Matrix má implementovanú galériu aplikácií pre populárne weby ako napríklad blogovú platformu WordPress.



Galéria aplikácií pre populárne weby

WebMatrix umožňuje tvorbu dynamických webových stránok, ktorých obsah sa mení na základe vstupov používateľov, prípadne aktuálnych údajoch v databáze. Kód je umiestnený v súboroch s príponami cshtml, alebo vbhtml, ktoré využívajú novú syntax nazývanú **Razor**. Riadky, či bloky Razor kódu rozoznate veľmi jednoducho, začínajú prefixom @, čiže zavináčom.

Web Matrix odlišuje Razor kód od ostatného kódu (HTML, JavaScript...) sivým podfarbením

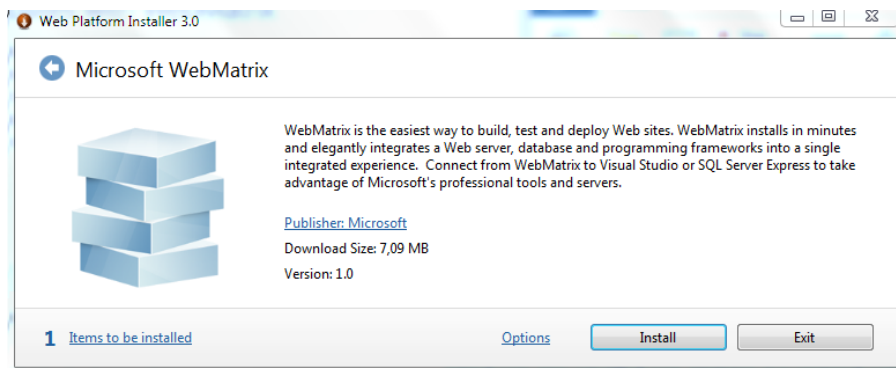
Projekty z nástroja WebMatrix je možné jednoducho migrovať do nástroja VisualStudio a tak sa v prípade osvedčeného webového projektu posunúť na vyššiu profesionálnejšiu métu.

Ak webová aplikácia pracuje s údajmi, súčasťou projektu sa stáva aj **SQL Server Compact**. Je to odľahčená embedded databáza, čo znamená, že jej engine sa pripája priamo k projektu s ktorým sa šíri. Z tohto dôvodu je (podobne ako kancelárska databáza Access) plne súborovo orientovaná.

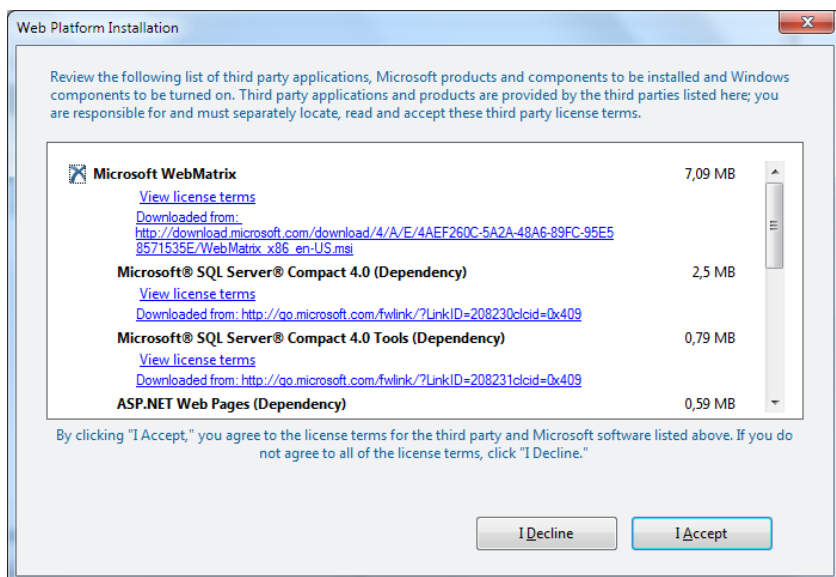
Inštalácia nástroja Web Matrix

Inštaláciu nástroja WebMatrix začinite na adrese <http://www.microsoft.com/web/webmatrix/>.

V úvodnom dialógu sa zobrazia podrobnosti o komponentoch, ktoré sa nainštalujú.

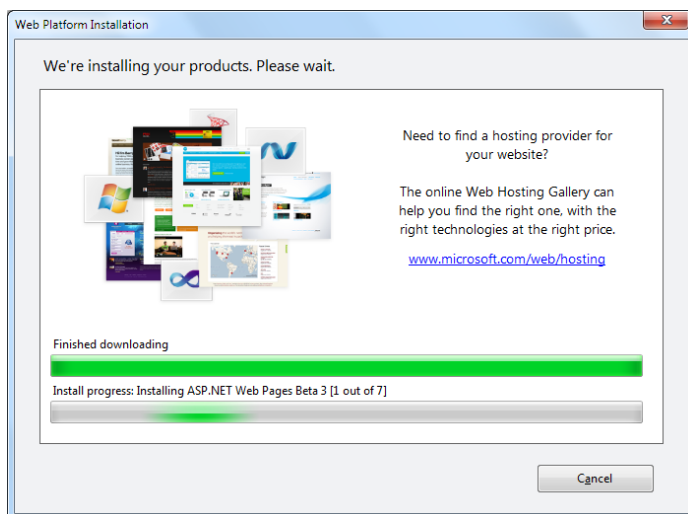


Inštalácia nástroja Web Matrix



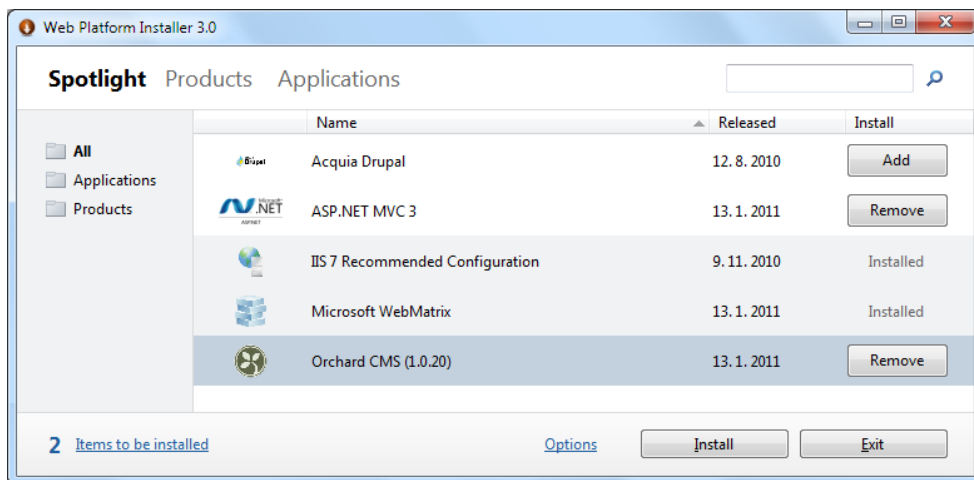
Podrobnosti o inštalovaných komponentoch

Samozrejme tvorcovia nástroja WebMatrix predpokladajú, že s ich projektom dosiahnete úspech, teda, že vytvoríte dobrú aplikáciu, ktorú budete chcieť umiestniť na web. Preto v priebehu inštalácie budete informovaní o možnostiach webhostingu. Vstupný bod je na adrese <http://www.microsoft.com/web/hosting/home>.



Počas inštalácie budete informovaní o možnostiach webhostingu

Po nainštalovaní nástroja WebMatrix a základných komponentov integrovaných do tohto prostredia (Webový server IIS, SQL Server Compact...) dostanete ponuku na inštaláciu prídavných komponentov a aplikácií, napríklad ASP.NET MVC 3, či CMS (ContentManagement System) systémov.



Možnosť inštalácie prídavných komponentov a aplikácií

Prvý projekt webovej aplikácie

Motto: „Nebojte sa“, upokojuje lekár pacienta, ktorému práve prišiel dva prsty odseknuté pri rúbaní dreva, „keď sa vám to zahojí, budete môcť hrať aj na klavíri...“. „No to som zvedavý“, čuduje sa pacient, „ja totiž na klavíri hrať neviem...“

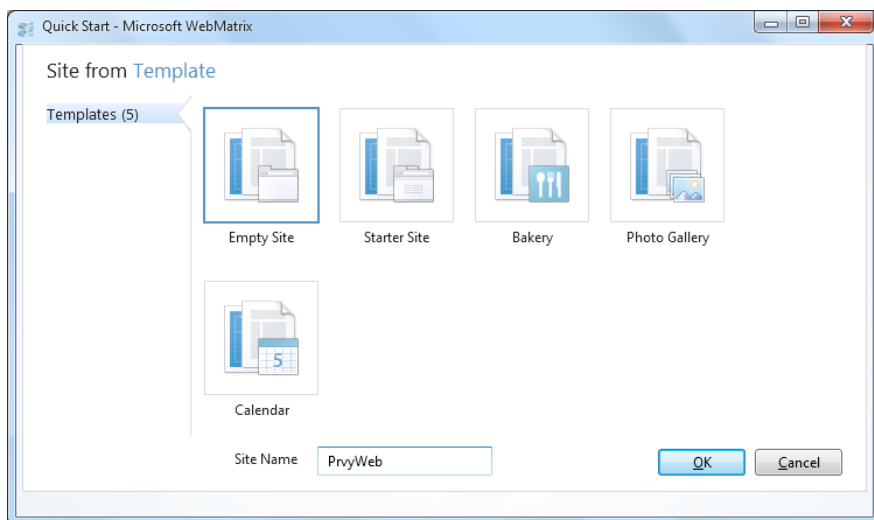
Ako bolo konštatované v úvodnej stati, vývojové prostredie WebMatrix je koncipované tak, že v ňom môžete začať *ihneď po inštalácii vytvárať webové aplikácie*. **A to platí v plnom rozsahu aj pre úplných začiatočníkov**. Samozrejme, hlavne pre začiatočníkov platí aj druhé tvrdenie, že si postupne prakticky osvoja aj pokročilejšie témy, takže vaša druhá aplikácia už bude oveľa sofistikovanejšia.



Sprievodca rýchlym vytvorením projektu

Začnite vytvorením projektu zo šablóny, teda aktivujte voľbu **Site From Template**. Šablóny sú vzory vopred vytvorených súborov a stránok pre rôzne typy webových stránok.

Pre prvý projekt vyberte šablónu **Empty Site** a pomenujte projekt, napríklad **PrvyWeb**.



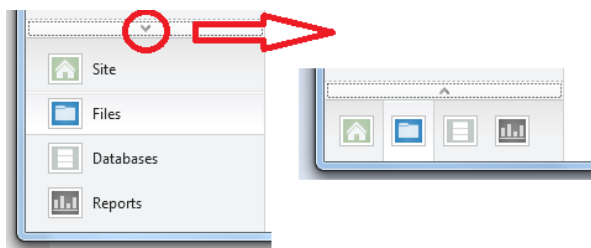
Vytvorenie projektu zo šablóny

Teraz nastal čas na zoznámenie sa s používateľským rozhraním vývojového prostredia WebMatrix. V hornej časti obrazovky nástroja WebMatrix je osvedčený panel nástrojov a pás kariet (v originálnej

terminológii Ribbon), ktorý dôverne poznajú používatelia kancelárskeho balíka Microsoft Office 2010 (2007). Panel obsahuje tlačidlá podľa vybraného pracovného priestoru.

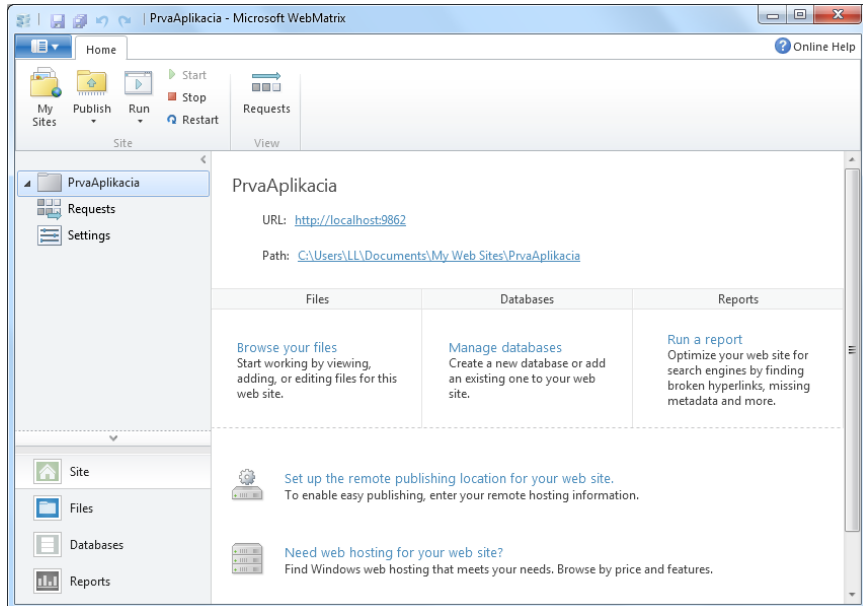
V ľavej dolnej časti je okno s tlačidlami na výber pracovného priestoru, teda režimu práce. Okno obsahuje tlačidlá:

- Site,
- Files,
- Databases,
- Reports.



Kliknutím na symbol šípky môžete okno s tlačidlami pre výber pracovného priestoru zminimalizovať

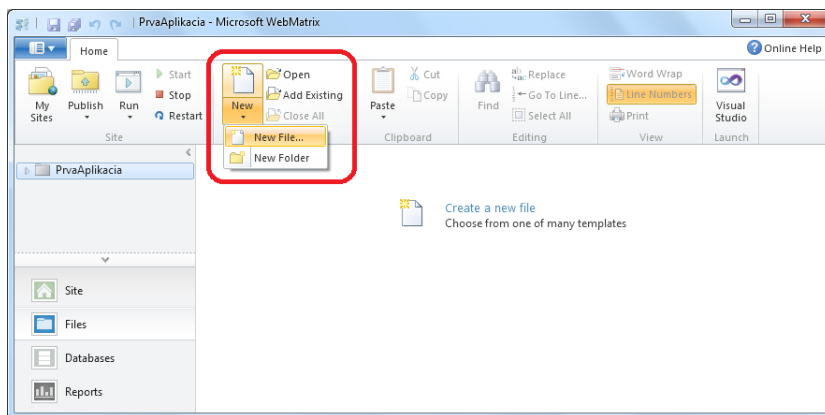
Vpravo je pracovná plocha, kde v závislosti od režimu môžete robiť návrh používateľského rozhrania, upravovať súbory, zobrazovať štatistiky a tak ďalej. V dolnej časti je panel oznámení.



Vytvorenie projektu zo šablóny

Vytvorenie webovej stránky

Prepnite WebMatrix do režimu **Files**. Projekt zatiaľ neobsahuje žiadne stránky. Na paneli nástrojov v skupine **Files** kliknite na tlačidlo **New** a v menu vyberte položku **New File...**



V režime Files môžete pridávať do svojho projektu súbory

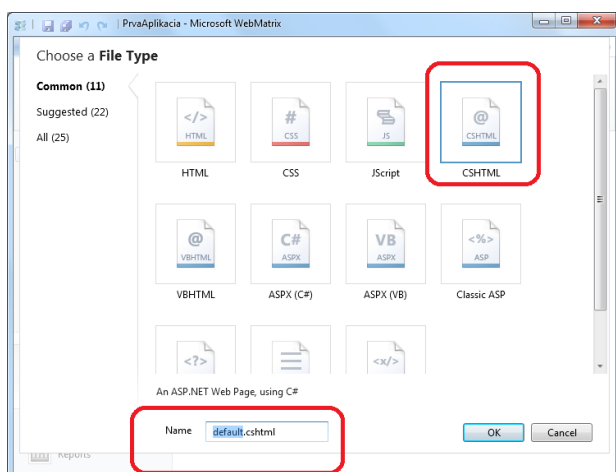
Jadrom prvej cvičnej webovej aplikácie bude stránka, na ktorej bude kód s Razor syntaxou. Stránky obsahujúce Razor kód majú špeciálnu príponu v závislosti od použitého programovacieho jazyka:

- **cshtml** – pre programovací jazyk C#,
- **vbhtml** – pre programovací jazyk VisualBasic.

Rozdiely v syntaxi oboch programovacích jazykov nie sú príliš rozdielne. V tejto publikácii sa zameriame na moderný a perspektívny programovací jazyk C#.

Poznámka: V novo vytvorenom projekte vo WebMatrixe sa objavil súbor robots.txt. Ako sa dá vydedukovať z názvu, jeho účelom je zjednodušiť fungovanie internetových vyhľadávačov. Tento súbor vo verzii Beta 3 nástroja WebMatrix nebol a v novej verzii sa nedá prehliadnuť, lebo je to vlastne jediný súbor v novovytvorenom projekte. Niektorí chcú dosiahnuť aby vyhľadávače na jeho stránkach lepšie fungovali, iní zasa chcú zabrániť aby im roboty „korzovali“ po ich privátnych stránkach. Viac podrobností nájdete na <http://tools.seobook.com/robots-txt/>.

Vyberte typ súboru CSHTML a pomenujte hodefault.cshtml.



Ponuka typov súborov

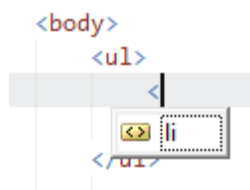
CHTML je špeciálny typ webovej stránky používaný vo WebMatrixe. Okrem štandardného HTML kódu klasickej webovej stránky, a JavaScript kódu môže tiež obsahovať špeciálny kód s Razor syntaxou.

WebMatrix vytvorí stránku a otvorí ju v prostredí editora. Stránka obsahuje základné tagy HTML stránky, zatiaľ bez Razor príkazov.

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="utf-8" />
<title></title>
</head>
<body>

</body>
</html>
```



Inteligentný pomocník Intellisense pomáha pri tvorbe HTML kódu

Na prvej stránke sa pokúsime niečo vypísať. Toto je jednoduchá statická úloha, takže by ste ju mohli realizovať výhradne pomocou HTML kódu. Aby sme urobili prvý krok do sveta ASP.NET, zrealizujeme túto stránku tak, že naplníme dve premenné textovými reťazcami a vypíšeme ich obsah:

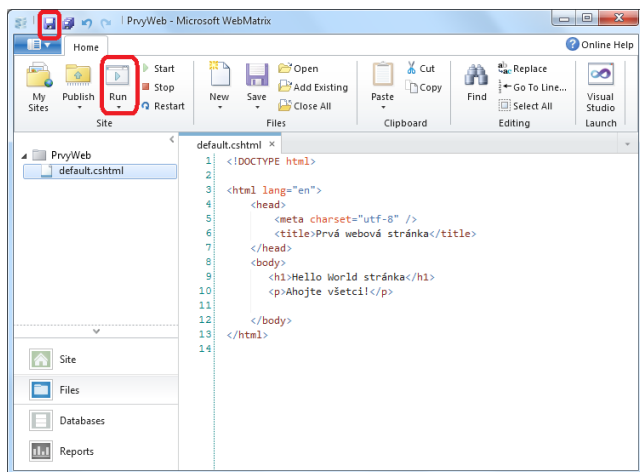
```
<html lang="en">
<head>
<meta charset="utf-8" />
<title>Prvá webová stránka</title>
</head>
<body>
    @{ var sNadpis = "Hello World stránka"; }
    @{ var sText = "Ahojte všetci!"; }

    <h1>@sNadpis</h1>
    <p>@sText</p>
</body>
</html>
```

Spustenie webovej aplikácie

Hoci webové aplikácie bežne používate, nemusíte sa starať o ich beh a vo väčšine prípadov ani netušíte kde je server na ktorom vaša aplikácia beží umiestnený. Teraz je všetko inak, webová aplikácia bude pre účely testovania a ladenia spustená priamo na vašom počítači.

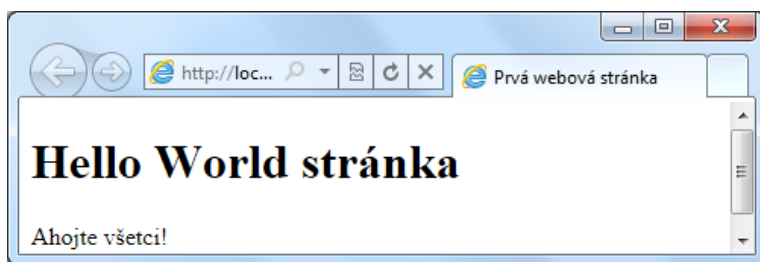
Pomocou ikony so symbolom diskety v hornom ráme nástroja WebMatrix uložte vykonané zmeny a na paneli nástrojov aktivujte tlačidlo **Run**.



Prvé spustenie webovej aplikácie z prostredia WebMatrix

Poznámka

*Pred spustením aplikácie sa uistite, že máte v ľavom paneli nástroja WebMatrix vybranú stránku, ktorú chcete testovať. WebMatrix spustí vybranú stránku a to aj v prípade ak práve upravujete inú stránku. Ak nie je žiadna stránka vybraná WebMatrix sa pokúsi spustiť implicitne predvolenú stránku **default.cshtml**. Ak nenájde ani stránku s týmto názvom, prehľadávač zobrazí chybu.*

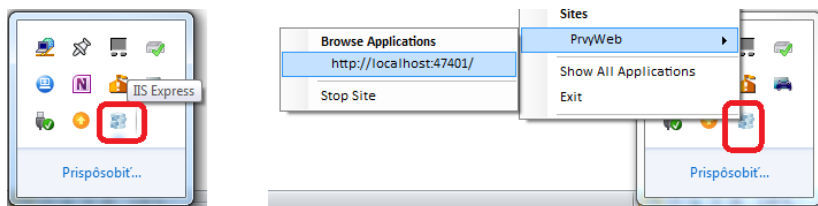


Zobrazenie stránky aplikácie v prehľadávači webového obsahu

Spustenie webovej aplikácie je značne odlišné od behu aplikácií ktoré na svojich počítačoch bežne spúšťate, či už pomocou menu, alebo ikon na pracovnej ploche. Webové aplikácie bežia reálne na serveri, preto pri ich spustení na lokálnom počítači je potrebné zabezpečiť funkcie takéhoto servera. Našťastie WebMatrix pri spustení aplikácie toto všetko zabezpečí automaticky. Aktivuje webový server (IIS Express) a jeho prostredníctvom zobrazí vašim kódom vygenerovaný obsah stránky v prehľadávači webového obsahu. Všimnite si v prehľadávači adresu URL

<http://localhost:47401/default.cshtml>.

Spustenie takéhoto servera, ktorý je, integrálnou súčasťou vývojového prostredia je indikované ikonou v pravom dolnom rohu obrazovky operačného systému v blízkosti ikony času.



Ikona webového servera IIS Express

Hoci ste sa zatiaľ nezoznámili s Razor syntaxou, skúste nasledujúci príklad, ktorý vypíše aktuálny dátum a čas:

```
<body>
<h1>Hello World stránka</h1>
<p>Dátum a čas: @DateTime.Now</p>
</body>
```

Aktuálny dátum a čas sa na stránke zobrazí v tvare: **Dátum a čas: 3. 1. 2011 20:27:50.**

Teraz možno klasici skeptici očakávajú, že po tomto klasickom príklade výpisu budeme pokračovať formátovaným výpisom obsahu premenných, výpisom v cykle a podobným klasickým finesami. Pozrite sa však pozornejšie na rok v dátume. V roku 2011 sa od webových stránok očakáva podstatne viac a medzi základné funkcie patrí čítanie, pripojenie sa na sociálne siete, zdieľanie obsahu... Takže absolvujete len úvod do syntaxe Razor a návod na inštaláciu doplnkov a ste pomerne dobre vyzbrojení pre tvorbu aplikácií využívajúcich sociálne siete.

Ľahký úvod do Razor syntaxe

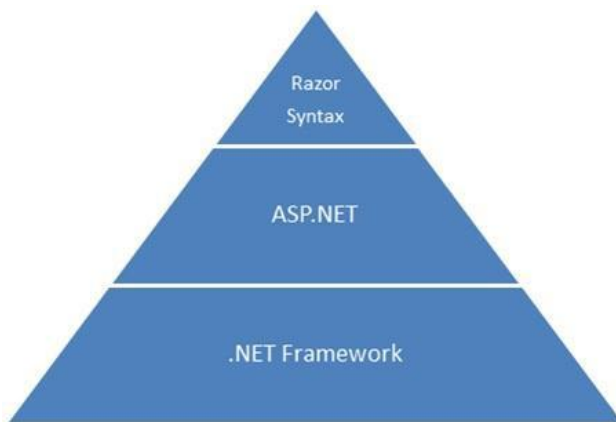
Razor je syntaktický predpis pre vkladanie kódu, ktorý sa vykonáva na serveri do webovej stránky. Kód bežiaci na serveri môže dynamicky vytvárať obsah interpretovaný vizuálne v klientovom prehľadávači webových stránok.

Ak sa spoliehate na intuíciu, môžete túto kapitolu preskočiť a pustiť sa do tvorby stránok podľa vzorových príkladov a sem sa vrátiť len v prípade nejasností.

Stránky obsahujúce Razor kód majú špeciálnu príponu v závislosti od použitého programovacieho jazyka:

- Cshtml – pre programovací jazyk C#,
- Vbhtml – pre programovací jazyk VisualBasic.

Ak server spracováva stránky s týmito príponami, spustí kód, ktorý je označený syntaxou Razor a potom pošle výsledok na stránku do prehľadávača.



Razor je z hľadiska architektúry situovaný nad vrstvami .NET Framework a ASP.NET

Nový typ stránok s Razor syntaxou s príponami (.cshtml resp. vbhtml) je možné na serveri kompilovať priamo, bez nutnosti prechodu cez ASP.NET runtime.

Razor vám umožní jednoduchý a intuitívny prechod medzi kódom a značkovacím jazykom, porovnajte jeho jednoduchosť s ASP.NET webovými formulármi alebo PHP.

```
<ul>
  <% for (int i = 0; i < 10; i++) { %>
    <li><% =i %></li>
  <% } %>
</ul>
```

```
<ul>
  <?php
    for ($i = 0; $i < 10; $i++) {
      echo("<li>$i</li>");
    }
  ?>
</ul>
```

```
<ul>
  @for (int i = 0; i < 10; i++) {
    <li>@i</li>
  }
</ul>
```

Porovnanie syntaxe ASP.NET WebForms (hore), PHP (v strede) a RAZOR (dole)

Razor výrazy sa začínajú znakom @, za ktorým môže nasledovať jednoriadkový výraz, alebo viacriadkový blok kódu @{ } v zložených zátvorkách.

```
@{ var nCislo = 49; }
@{ var sText = "Ahoj"; }
```

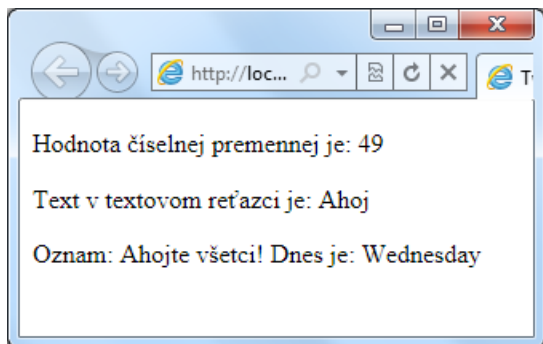
Ako uvidíte v príklade, Razor premenné začínajúce znakom @ môžu byť kdekoľvek v HTML kóde:

```
<p>Hodnota číselnej premennej je: @nCislo </p>
<p>Text v textovom reťazci je: @sText</p>
```

Blok kódu v zložených zátvorkách:

```
@{
    var sPozdrav = "Ahojte všetci!";
    var dtDen = DateTime.Now.DayOfWeek;
    var sOznam = sPozdrav + " Dnes je: " + dtDen;
}
<p>Oznam: @sOznam</p>
```

Vo vnútri bloku kódu je potrebné oddeľovať jednotlivé príkazy bodkočiarkou.



Výpis pomocou vyššie uvedených Razor výrazov

Vo vnútri Razor kódu môžete používať HTML tagy, napríklad:

```
@if(IsPost)
{
<p>Ahoj už sa poznáme teraz je: @DateTime.Now a na túto
    stránku ste prišli cez postback!</p>
}
else
{
<p>Ahoj, si tu prvý krát, teraz je: <br /></p> @DateTime.Now
}
```

Komentáre

K dobrým zvykom úspešných programátorov patrí komentovanie jednotlivých blokov kódu, aby ste mohli aj po niekoľkých mesiacoch nadviazať na svoju prácu. Komentovanie kódu je taktiež nutným predpokladom tímovej spolupráce viacerých ľudí nad jedným projektom. Razor syntax umožňuje používať jednoriadkové a viacriadkové komentáre.

Jednoriadkový komentár:

```
@* Jednoriadkový komentár. *@
```

Viacriadkový komentár:

```
@*
Viacriadkový komentár.
druhý riadok.
*@
```

Vo vnútri Razor blokov je možné využívať pre komentár syntax príslušného programovacieho jazyka:

```
@{
    // jednoriadkový komentár.
    var nPocet = 33;

    /* Viacriadkový komentár
       Podľa syntaxe programovacieho jazyka C#. */
}
```

Používanie premenných

Do premenných serverového kódu sa môžu ukladať rôzne dočasné údaje a takto je možné prekonať jednu z najväčších nevýhod HTML protokolu – bezstavovosť. Premenné môžu byť rôznych typov, číselné, textové, dátumové... napríklad:

```
@{ var nCislo = 49; }
@{ var sText = "Ahoj"; }
@{ var dtAktRok = DateTime.Now.Year; }
```

Razor syntax v programovacom jazyku C# rozlišuje veľké a malé písmená, takže nasledujúci kód vytvorí dve rôzne premenné:

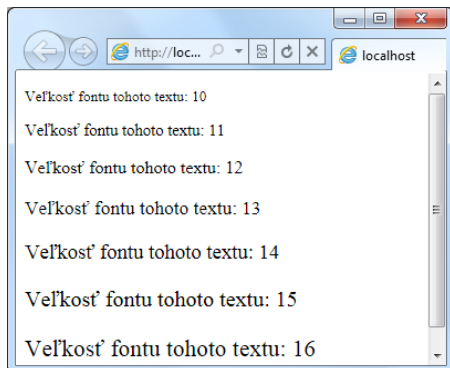
```
@{  
var NazovProduktu = "Mydlo";  
var nazovProduktu = "Mydlo";  
}
```

Razor syntax v programovacom jazyku VisualBasic veľké a malé písmená nerozlišuje.

Cyklus

Pre vykonávanie opakovaných činností slúži cyklus, pričom v každej iterácii môže byť činnosť modifikovaná napríklad aktuálnou hodnotou riadiacej premennej cyklu. V príklade sa hodnotou tejto premennej riadi veľkosť fonu.

```
@for(var i = 10; i < 17; i++)  
{  
<p style="font-size: @(i + "pt")">Veľkosť fonu tohoto textu: @i</p>  
}
```



Výpis pomocou cyklu

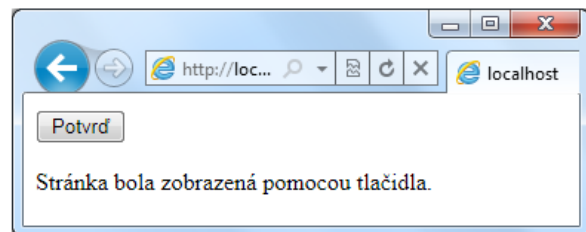
Vetvenie kódu na základe podmienky

Základnou programovou konštrukciou pre dynamiku aplikácie je podmienka, na základe jej vyhodnotenia sa program následne vetví. Najjednoduchšie sa podmienka vytvorí pomocou príkazu `if`. Napríklad:

```
@{  
var bDatum = false;  
  
if(bDatum)  
{  
@DateTime.Today;  
}  
else  
{  
<text>- - </text>  
}  
}
```

V ďalšom príklade sa vyhodnotí hodnota stavovej premennej `IsPost`, podľa ktorej sa zistí, či bola stránka otvorená prvýkrát, alebo po stlačení tlačidla:

```
@{
    var oznam = "";
    if(IsPost)
    {
        oznam = "Stránka bola zobrazená pomocou tlačidla.";
    }
    else
    {
        oznam = "Prvé zobrazenie tejto stránky.";
    }
}
<!DOCTYPE html>
<html><head></head>
<body>
<form method="POST" action="" >
<input type="Submit" name="Potvrđ" value="Potvrđ"/>
<p>@oznam</p>
</form>
</body>
</html>
```



Vetvenie programu na základe podmienky.

Formulár na zadávanie údajov

Cieľom tejto state je interakcia medzi používateľom a webovou aplikáciou, ktorá spočíva v spracovaní údajov zadaných používateľom prostredníctvom webových formulárov. Následne sa tieto údaje spravidla spracujú pomocou aplikačného kódu, prípadne uložia do databázy. Najskôr ukážeme príklad jednoduchého formulára:

```
@{
    if (IsPost) {
        string sMeno = Request["Meno"];
        int nVek = Request["Vek"].AsInt();
        string sHobby = Request["Hobby"];
    }
    <text>
        Zadané údaje: <br />
        Meno: @sMeno <br />
        Vek: @nVek <br />
        Hobby: @sHobby<br />
    </text>
}
<!DOCTYPE html>

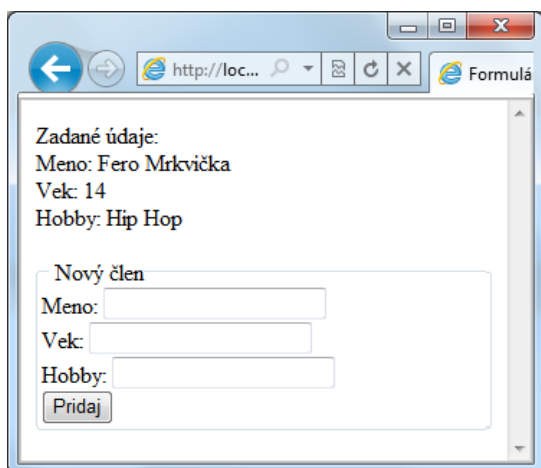
<html lang="en">
<head>
<meta charset="utf-8" />
<title>Formulár</title>
```

```

</head>
<body>
<form method="post" action="">
<fieldset>
<legend>Nový člen</legend>
<div>
<label for="Meno">Meno:</label>
<input type="text" name="Meno" value="" />
</div>
<div>
<label for="Vek">Vek:</label>
<input type="text" name="Vek" value="" />
</div>
<div>
<label for="Hobby">Hobby:</label>
<input type="text" name="Hobby" value="" />
</div>
<div>
<label></label>
<input type="submit" value="Pridaj" class="submit" />
</div>
</fieldset>
</form>

</body>
</html>

```



Zobrazenie formulára a zadaných údajov

Validovanie používateľom zadaných hodnôt

Na formuláre na webových stránkach sú kladené určité požiadavky, hlavne vzhľadom na komfort používateľov. Hlavne v aplikáciách kde „ide o biznis“ je dôležité, aby údaje zadané používateľom (údaje, rodné číslo, číslo objednávky....) boli správne. Od bežného klienta, napríklad návštevníka internetového obchodu samozrejme nemôžete chcieť, aby poznal všetky nuansy vašej aplikácie. Pre vývojára je „samozrejmé“ že sa bude používať desatinná bodka, matematicky „odchovaný“ klient neprogramátor tam celkom pochopiteľne zadá čiarku, problémy sú zo zadávaním formátu dátumu a času a podobne. Preto musíte používateľovi pomôcť nielen vhodným a jednoznačným návrhom formulára na zadávanie údajov, ale aj zadávané údaje kontrolovať a používateľa usmerňovať tak, aby výsledkom boli správne a konzistentné údaje. V príklade je ukázané ošetrenie zadávania pre údaj, ktorý používateľ musí zadať:

```

@{
    if (IsPost) {

        var chyba = false;
        string sMeno = Request["Meno"];
        if (sMeno.IsNullOrEmpty())
        {
            chyba = true;
            @:Musite zadat meno.<br />
        }

        int nVek = Request["Vek"].AsInt();
        string sHobby = Request["Hobby"];

        if (chyba == false)
        {
<text>
            Zadané údaje: <br />
            Meno: @sMeno <br />
            Vek: @nVek <br />
            Hobby: @sHobby<br />
</text>
        }
    }
}

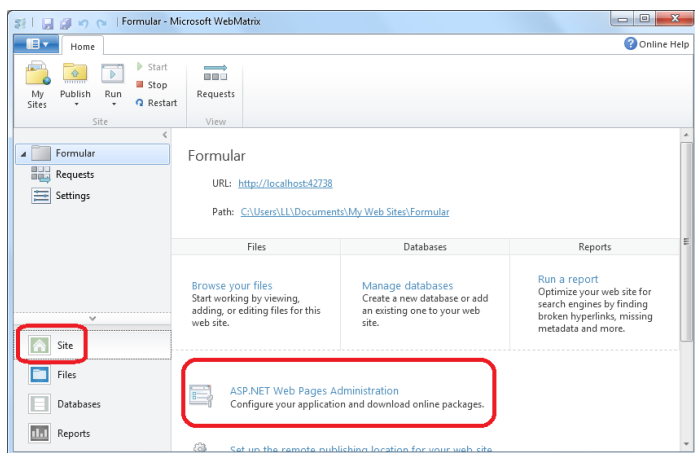
```

Správa aplikácie a inštalácia doplnkov

Kedže sme sľúbili, že sa budete s novými črtami zoznamovať postupne, nastal vhodný čas na nainštalovanie knižnice pomocných objektov, ktoré vám podstatne zjednodušia realizáciu funkcií, ktoré sa na webových stránkach často vyskytujú, napríklad pridanie tlačidla, ktorým môžu návštevníci vašej stránky označiť na sociálnej sieti (Facebook...), že sa im páči, prípadne zobrazenie údajov z databázy s možnosťou utriedenia a podobne.

Pre pripojenie doplnkov k vašej aplikácii je potrebné prejsť do režimu administrácie.

Môžete to urobiť dvomi spôsobmi: buď z pracovného priestoru Site, ako je znázornené na obrázku, alebo modifikáciou adresy URL aplikácie.



Aktivovanie administrátorskej stránky

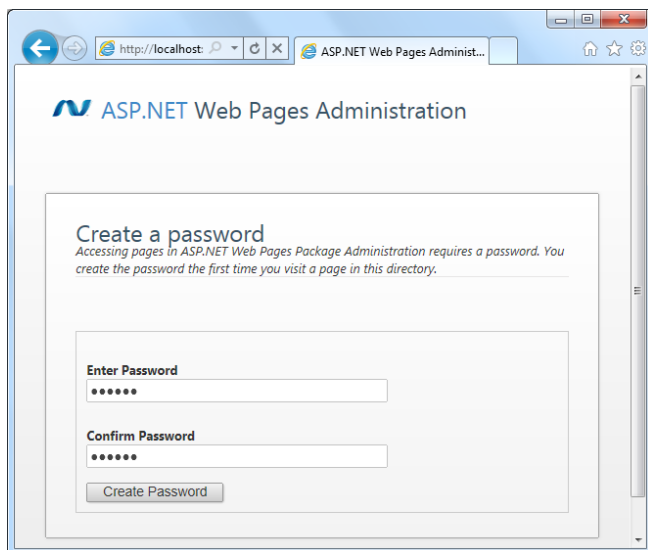
Spustíte z prostredia nástroja WebMatrix webovú aplikáciu a následne v prehľadávači webového obsahu nahradíte v adrese URL názov stránky za poslednou lomkou reťazcom „_Admin“. V našom príklade sme adresu

<http://localhost:47401/default.cshtml>

modifikovali na

http://localhost:47401/_Admin.

Pri prvom prístupe na administrátorskú stránku je potrebné vytvoriť heslo pre administrátora.



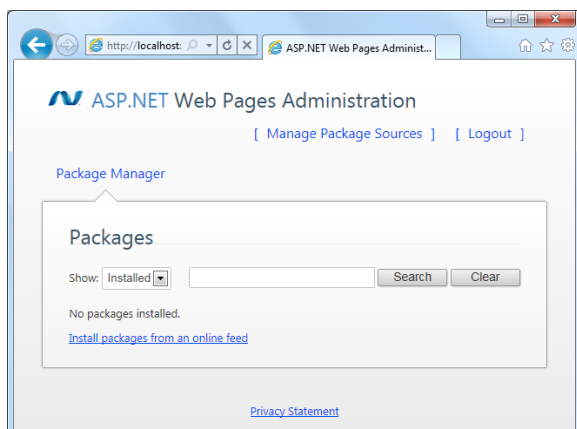
Administrátorská stránka ASP.NET stránok

Po prihlásení sa zobrazí upozornenie, že zašifrované heslo je uložené v súbore `_Password.config` v podpričinku vašej aplikácie `/App_Data/Admin/`. Je to z bezpečnostných dôvodov, nakoľko ASP.NET neumožňuje prístup k súborom, ktoré začínajú podčiarknikom (`_`). Aby administrácia začala fungovať, je potrebné z názvu súboru podčiarkník odstrániť. Súbor teda premenujte na **Password.config**.



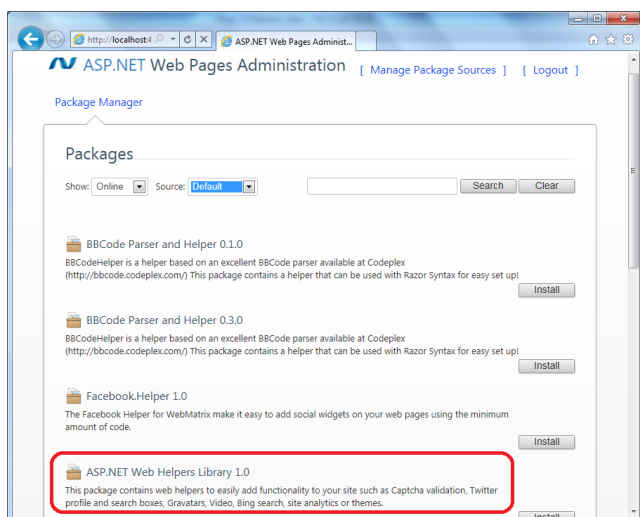
Bezpečné uloženie hesla

Po premenovaní sa spustí správca nainštalovaných doplnkov **PackageManager**.

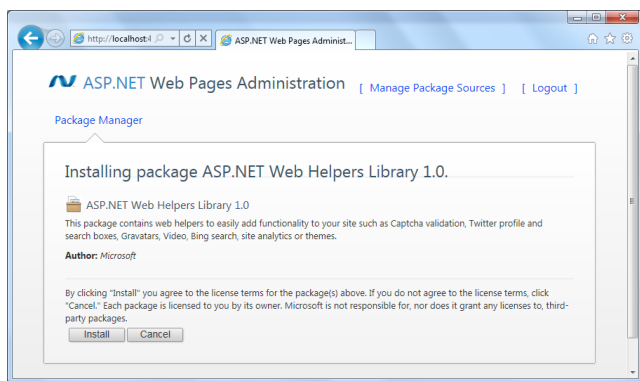


PackageManager

Prepnite na vyhľadavanie „Online“ balíčkov. Zobrazí sa vám zoznam dostupných doplnkov, ktoré môžete pridať do aplikácie. Nainštalujte balíček **ASP.NET Web HelpersLibrary**. Následne sa vám zobrazia podrobnejšie informácie o vybranom doplnku. Doplnok nainštalujete kliknutím na tlačidlo **Install**.



Inštalovanie ASP.NET Web HelpersLibrary



Detaily o inštalovanom balíku

Po nainštalovaní sa zobrazí zoznam nainštalovaných doplnkov. Pomocou tohto dialógu môžete doplnky aj odinštalovať.

Vzhľadom na to, že dnes sa už prakticky žiadna dobrá a moderná webová aplikácia nezaobíde bez konektivity na najrozšírenejšiu sociálnu sieť, zopakujte postup popísaný v predchádzajúcej stati aj pre **Facebook.Helper**.

Prepojenie vašej webovej aplikácie na sociálne siete

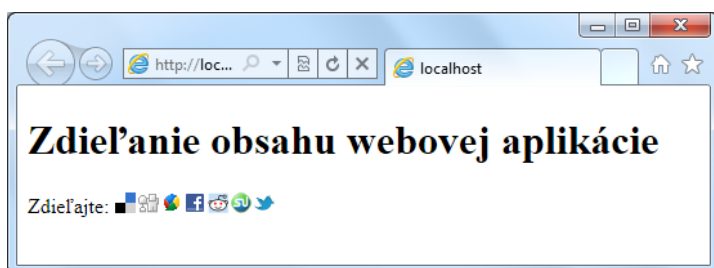
Jedna z prvoradých vecí, ktoré by ste mali urobiť, aby sa vaše stránky stali viac populárne, je ich integrácia s populárnymi sociálnymi sieťami na webe. Sociálne siete slúžia na prepájanie ľudí, ktorí majú niečo spoločné. Každý používateľ si vytvorí vlastný profil, v ktorom napíše o sebe základné informácie. Na základe týchto informácií sa nadväzujú vzťahy medzi používateľmi, ktorí sa spájajú do skupín. Vzájomnými prepojeniami používateľov a skupín vzniká sieť vzťahov.

Zdieľanie obsahu vašej stránky na sociálnych sieťach

Na zaujímavé veci, ktoré sa nachádzajú na vašich stránkach budete určite chcieť upozorniť svojich priateľov na sociálnych sieťach. WebMatrix to umožňuje pomocou jediného riadku kódu.

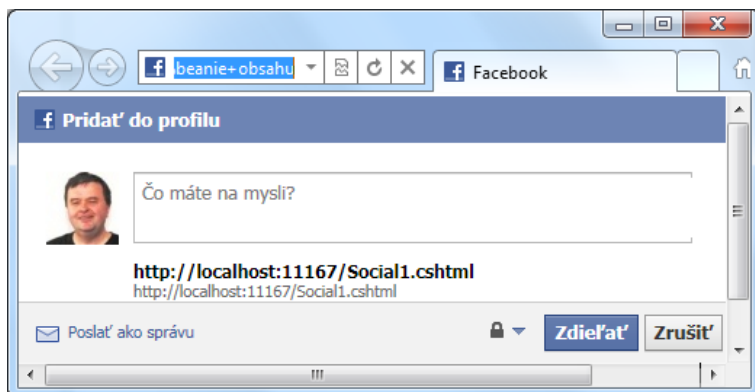
Pre túto funkciu je potrebné nainštalovať doplnok ASP.NET Web HelpersLibrary:

```
<body>  
<h1>Zdieľanie obsahu webovej aplikácie</h1>  
    Zdieľajte: @LinkShare.GetHtml("Zdieľanie obsahu")  
</body>
```

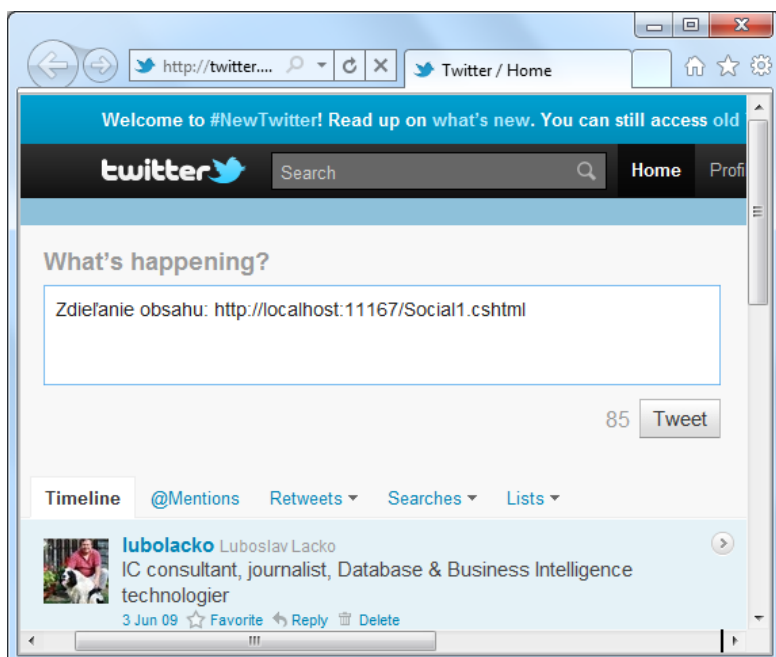


Zdieľanie obsahu so sociálnou sieťou

Ak používateľ klikne na ikonu príslušnej sociálnej siete, zobrazí sa okno pre pridanie príspevku. V našom prípade sme klikli na ikonu sociálnej siete Facebook, takže sa zobrazila stránka Facebooku so zadávacím oknom „Čo máte na mysli?“.



Zdieľanie obsahu na sociálnej sieti Facebook



Zdieľanie obsahu na sociálnej sieti Twitter

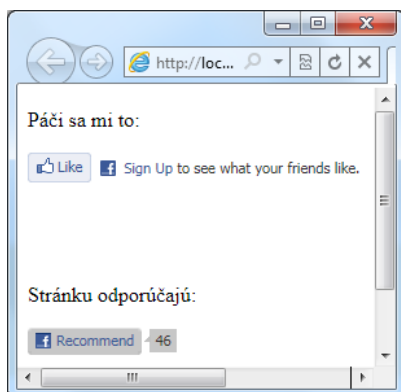
Facebook – páči sa mi to

Jednou z najčastejšie využívaných funkcií populárnej siete Facebook je tlačidlo „Páči sa mi to“, pomocou ktorej ľudia označia, že sa im niečo na vašej stránke páči. Facebook môžete využívať, aby sa ste sa spojili so svojou komunitou, prípadne zákazníkmi a niečo (názor, hobby, produkt...) komunikovali a propagovali.

Pre túto funkciu je potrebné nainštalovať doplnok Facebook:

```
<body>
<p>Páči sa mi to:</p>  @Facebook.LikeButton()

<p>Stránku odporúčajú:</p>
  @Facebook.LikeButton(
    href: "http://www.asp.net/webmatrix",
    action: "recommend",
    width: 250,
    buttonLayout: "button_count",
    showFaces: true,
    colorScheme: "dark")
</body>
```



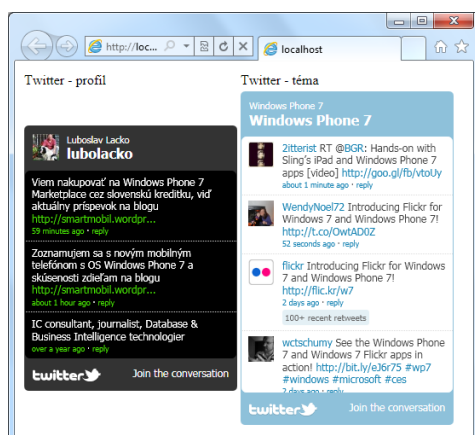
Hodnotenie obsahu cez Facebook

Twitter

Twitter je poskytovateľ sociálnej siete, ktorá umožňuje užívateľom posilať a čítať príspevky zaslané inými užívateľmi, nazývané tweety. Tweety sú textové príspevky dlhé maximálne 140 znakov, ktoré sa zobrazujú na užívateľovej profilovej stránke a na stránkach jeho odberateľov (followers). Užívatelia zasielajú alebo dostávajú tweety cez stránku twittera, pomocou krátkych textových správ (SMS) alebo externých aplikácií. Príspevky z mikrobloggerov napríklad k danej téme, môžete pridávať do svojej webovej aplikácie.

Pre túto funkciu je potrebné nainštalovať doplnok ASP.NET Web HelpersLibrary:

```
<body>
<table>
<tr>
<td>Twitter - profil</td>
<td>Twitter - téma</td>
</tr>
<tr>
<td>@Twitter.Profile("lubolacko")</td>
<td>@Twitter.Search("Windows Phone 7")</td>
</tr>
</table>
</body>
```



Zobrazenie informácií zo sociálnej siete Twitter

Môžete používateľovi umožniť vyhľadávanie najaktuálnejších tweetov k danej téme:

```
<h1>Twitter</h1>
<form action="" method="POST">
<div>
    Názov mikrobloggeru:
    <input type="text" name="TwitterUser" value=""/>
    <input type="submit" value="Submit" />
</div>
<div>
    @if (Request["TwitterUser"].IsEmpty()) {
        @Twitter.Search("microsoft")
    }
    else {
        @Twitter.Profile(Request["TwitterUser"])
    }
</div>
</form>
```



Zobrazenie mikrobloggerov Twittera. Implicitne sa zobrazí mikroblogger „Microsoft“. Vpravo je mikroblogger CNN.

Gravatar

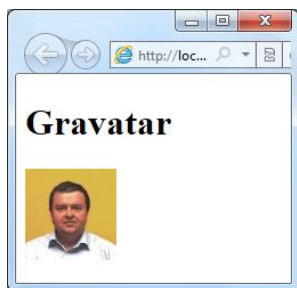
Pod týmto akronymom sa skrýva označenie GloballyRecognizedAvatar www.gravatar.com čiže po našom **globálne rozpoznávaný avatar**. Momentálne túto službu vlastní spoločnosť Automattic, ktorá prevádzkuje aj známy blogový portál Wordpress.com.

Ludia vnímajúci tváre a obrázky oveľa lepšie ako mená a ľahšie si asociujú konverzácie. Gravatar pridá stránkam individuálny nádech.

Môžete si tu vytvoriť identitu s obrázkom, ktorú môžete používať nielen na WordPressblogoch, ale aj mnohých iných webových aplikáciách a službách. Gravatar môžete jednoducho vložiť do svojej webovej stránky.

Pre túto funkciu je potrebné nainštalovať doplnok ASP.NET Web HelpersLibrary:

```
<body>
<h1>Gravatar</h1>
    @Gravatar.GetHtml("lubolacko@hotmail.sk", 80)
</body>
```



Zobrazenie Gravatara

Pridaním Gravatarov napríklad do komentárov sa stránky stanú osobitejšie a živšie, čo následne podnieti ľudí aby sa zapojili do aktivít na vašej stránke. Pre zobrazenie Gravatarov nie je potrebné vyplňať žiadnu registráciu a dokonca ani sami nemusíte mať Gravatar.

Integrovanie vyhľadávania do webovej aplikácie

Vaše stránky môžu používateľovi umožniť vyhľadávanie na Internete, bez toho aby ich musel opustiť a použiť niektorý z vyhľadávačov. Samozrejme z hľadiska vášho webu je užitočné ani nie tak všeobecné vyhľadávanie na Internete, ako skôr vyhľadávanie v kontexte vášho webu. Veľakrát sa stáva, že webová aplikácia, alebo portál používateľom požadované informácie obsahuje, no napríklad kvôli neprehľadnej navigačnej štruktúre sa k nim používateľ nedostane.

Úvaha: Možno ste si uvedomili, že vo väčšine prípadov oveľa ľahšie nájdete požadovanú informáciu na Internete, ako vo svojom vlastnom počítači a na externých USB diskoch, aj keď tušíte, že ju tam niekde máte. Našťastie systém Windows 7 podporuje organizovanie obsahu do knižníc a vynikajúce možnosti vyhľadávania v knižniciach.

Preto je dôležitá funkcia typu „Hľadať na tomto webe“, ktorá umožňuje užívateľom vyhľadávať a nájsť požadovaný obsah vašich stránok.

Pre fungovanie vyhľadávania je potrebné nainštalovať doplnok ASP.NET Web HelpersLibrary.

Rovnako používateľom uľahčíte prácu pridaním možnosti jednoducho vyhľadávať súvisiace stránky. Napríklad ak robíte stránky členov záujmového krúžku, môže byť zaujímavé prepojenie na ich školské, alebo osobné stránky, alebo blogy.

V prvom najjednoduchšom príklade pridáme na stránku pole pre zadanie vyhľadávaného výrazu. Po jeho aktivovaní sa zobrazia výsledky v samostatnom okne vyhľadávača Bing.

Tip: Tento scenár môže byť užitočný napríklad na stránkach venovaných nejakému hobby, kde sa vyskytujú aj odborné výrazy a začiatočník, ktorý navštíví vaše stránky si bude môcť vyhľadať, čo znamenajú výrazy, ktorým nerozumie.

Všimnite si, že metóde Bing.SearchBox, môžete zadať ako parameter webovú stránku, blog, alebo portál, na ktorom sa má vyhľadávať. Pod oknom pre zadanie vyhľadávacieho výrazu sa zobrazí prepínač, pomocou ktorého používateľ určí, či sa bude vyhľadávať na preddefinovanej stránke, alebo na celom webe.

Vyhľadavanie1.cshtml

```
<body>
<h1>Vyhľadavanie</h1>
  Nájdí na tematickom blogu: <br/>
  @Bing.SearchBox("www.aspnet.sk")
</body>
```

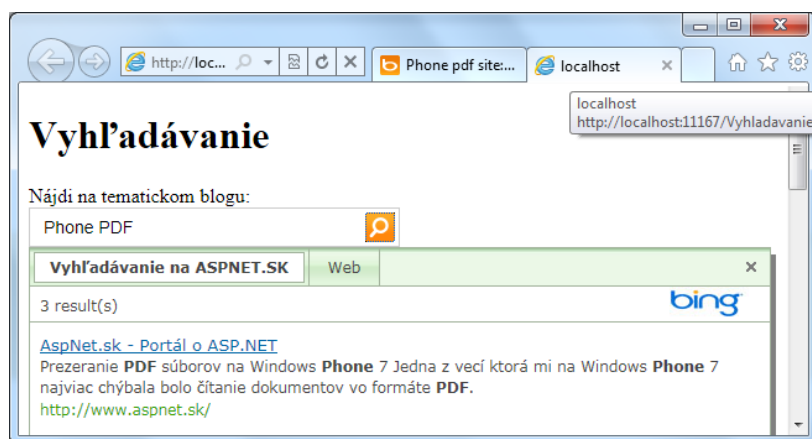


Integrovanie vyhľadávania do webovej stránky, výsledky sa zobrazia v samostatnom okne

Môžete použiť tesnejšiu integráciu výsledkov vyhľadávania do svojej stránky.

Vyhľadavanie2.cshtml

```
<body>
<h1>Vyhľadavanie</h1>
  Nájdi na tematickom blogu: <br/>
  @{ Bing.SiteUrl = "www.aspnet.sk";
    Bing.SiteTitle = "ASP.NET Custom Search";
  }
  @Bing.AdvancedSearchBox(boxWidth: "250px", resultWidth: 600, resultHeight: 900,
    themeColor: "Green", locale: "sk-SK")
</body>
```



Integrovanie vyhľadávania do webovej stránky

Posielanie elektronickej pošty z webovej aplikácie

Pri čítaní nadpisu tejto kapitoly vás asi ihneď napadne spam. No funkcia posielania elektronickej pošty z webovej aplikácie má oveľa širšie a užitočnejšie využitie. Môžete naprogramovať aplikáciu tak, aby vám e-mailom oznámila napríklad zaregistrovanie nového člena komunity, alebo môžete dostať oznam o problémoch a poruchách. Podobne aj návštevníci vašich stránok sa môžu zaregistrovať na odber e-mailov o novinkách, či pripravovaných podujatiach.

Pre fungovanie posielania správ je potrebné nainštalovať doplnok ASP.NET Web HelpersLibrary.

Ako príklad ukážeme typický formulár, pomocou ktorého môžu návštevníci stránky upozorniť na problém. Správa sa z formulára aplikácie automaticky odošle správcovi stránky.

Mail.cshtml

```
<html>
<head><title>Mail</title></head>
<body>
<h2>Upozornenie na problém</h2>
<form method="post" action="Posli.cshtml">
<div> Meno:
<input type="text" name="meno" />
</div>

<div> Popis problému: <br />
<textarea name="popis" cols="45" rows="4"></textarea>
</div>

<div><input type="submit" value="Odošli" />
</div>
</form>
</body>
```

Posli.cshtml

```
@{
    var meno = Request["meno"];
    var popis = Request["popis"];
    try {
        // Inicializácia WebMailu
        WebMail.SmtpServer = "xxxx.pcrevue.sk";
        WebMail.SmtpPort = 25;
        WebMail.EnableSsl = true;
        WebMail.UserName = "llacko";
        WebMail.From = "llacko@pcrevue.sk";
        WebMail.Password = "xxxx";

        // Send email
        WebMail.Send(to: "lubolacko@hotmail.sk",
            subject: "Upozornenie z aplikácie GALERIA - " + meno,
            body: popis
        );
    }
    catch (Exception ex ) {<text>Problém pri odosielaní</text>}
}
<html>
<head><title>Mail</title></head>
<body>
<p>Ďakujeme za upozornenie, <b>@meno</b>.</p>
</body>
</html>
```

The image shows a web browser window with a single tab titled 'Mail'. The address bar shows 'http://loc...'. The main content area displays a form titled 'Upozornenie na problém'. The form contains two input fields: 'Meno:' with the text 'Lubo' and 'Popis problému:' with the text 'Problém so súborami PNG'. Below the fields is a blue button labeled 'Odošli'.

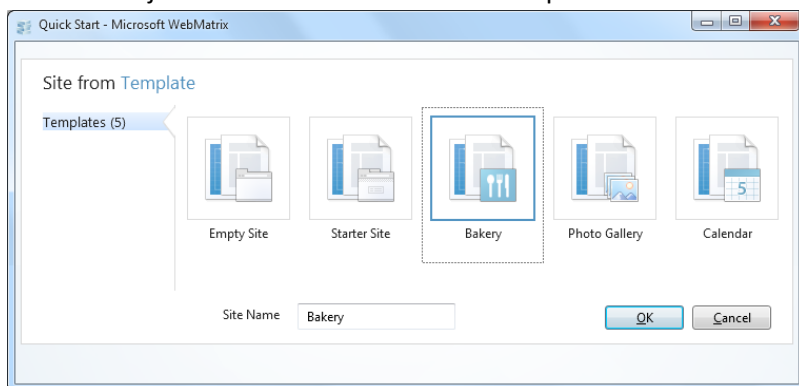
Formulár pre vytvorenie správy

Webová aplikácia zo šablóny

Prvý príklad pre zoznámenie sa so základnými funkciami vývojového prostredia WebMatrix v úvode bol veľmi jednoduchý, no tomu zodpovedá aj jeho funkčnosť. V reálnych webových aplikáciách sa požaduje oveľa viac obsahu, interaktivity a pochopiteľne aj dizajn na zodpovedajúcej úrovni. Zdalo by sa, že tak ako požadovaná funkcionálnosť reálnej webovej aplikácie je mnohokrát vyššia než pri spomínanom príklade, bude potrebné aj mnohokrát viac úsilia na jej vytvorenie. Šablóny v najmodernejších nástrojoch umožňujú vytvárať komplexné interaktívne webové aplikácie aj začiatočníkom, pričom pomerne bohatú funkcionálnosť dosiahnete bez akéhokoľvek programovania. Pre tento účel máte k dispozícii šablóny. Sú to plne funkčné vzorové aplikácie na riešenie veľmi často sa vyskytujúcich scenárov webových aplikácií. Niektoré šablóny

- **Starter Site**
- **Bakery**
- **PhotoGallery**
- **Calendar**

sa nainštalujú lokálne a mnohé ďalšie sú k dispozícii na webe.



Vytvorenie webovej aplikácie podľa šablóny Bakery

Použitie šablón lokalizovaných do slovenčiny

Aby sme vám uľahčili tvorbu stránok, všetky štyri šablóny sme lokalizovali do slovenčiny.

Môžete si ich prevziať na adrese na <http://msdn.microsoft.com/sk-sk/ee404903#rozne> v sekcii Rôzne.

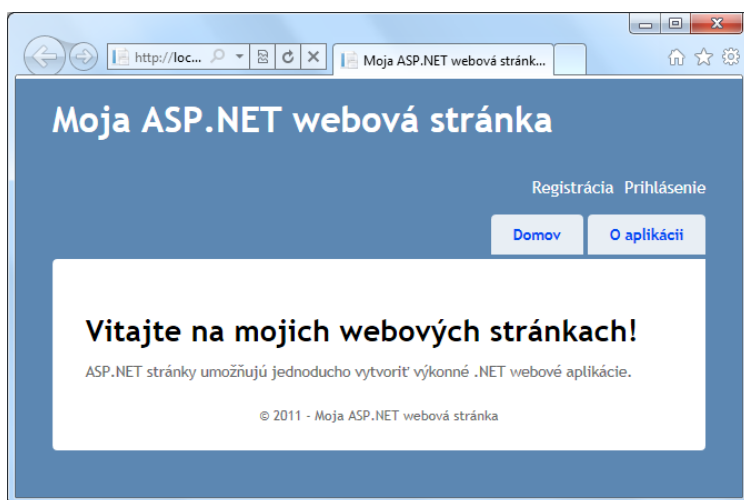
Nakopírujte lokalizovanú šablónu do vhodného priečinka a vo vývojovom prostredí nástroja Web Matrix ich otvoríte v úvodnej obrazovke aktivovaním voľby **Site FromFolder**.



Otvorenie lokalizovaných šablón z priečinka

Šablóna StarterSite

Táto šablóna slúži na vytvorenie stránok ku ktorým sa budú používatelia prihlasovať. Šablóna rieši nielen prihlasovanie, ale aj registráciu používateľov. Viac podrobností nájdete v stati **Len pre zaregistrovaných členov**.



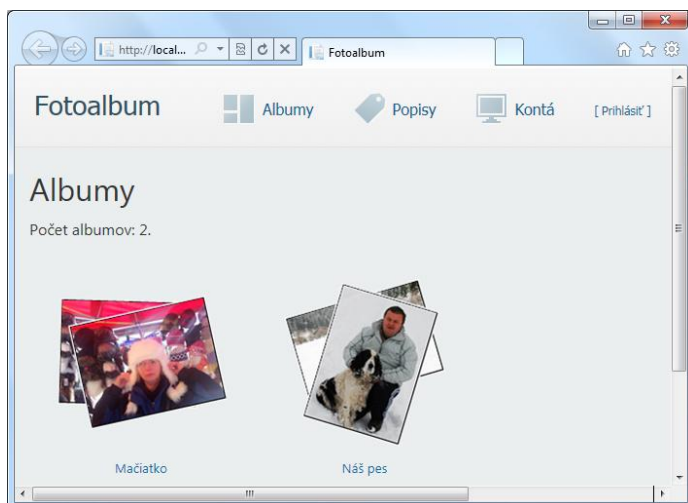
Šablóna aplikácie Starter Site

Šablóna Gallery (Fotoalbum)

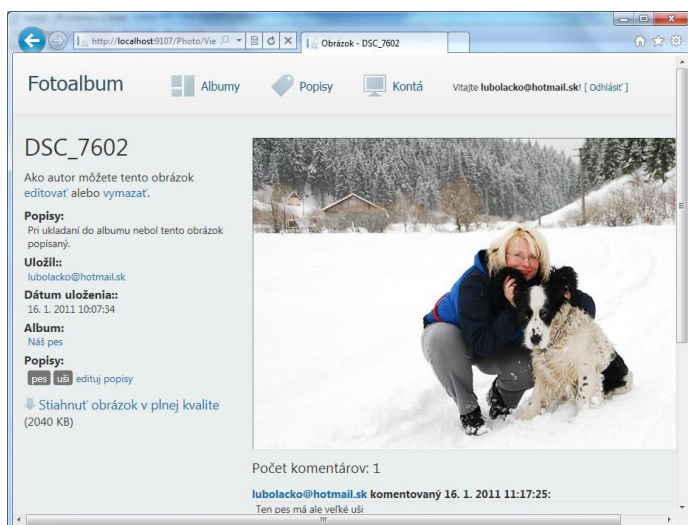
Každý sa rád podelí o zážitky s rodinou a priateľmi prostredníctvom fotiek. Éra klasických „papierových“ fotoalbumov je už nenávratne preč, a ak doma nejaké máte, tak snád' len na to aby ste návšteväť taktne naznačili, že je čas odísť. Aj posielanie veľkého množstva fotiek e-mailom je problematické a tak ideálnym riešením je webový album.

Možno budete namietat', že vo vývojovom prostredí bez možnosti grafického návrhu je vytváranie aplikácií typu webový fotoalbum zložité, nakoľko pri takejto aplikácii obzvlášť závisí na dizajne a v neposlednom rade aj na estetickom hľadisku používateľského rozhrania aplikácie. Ako to dosiahnuť vo vývojovom prostredí bez možnosti grafického návrhu?

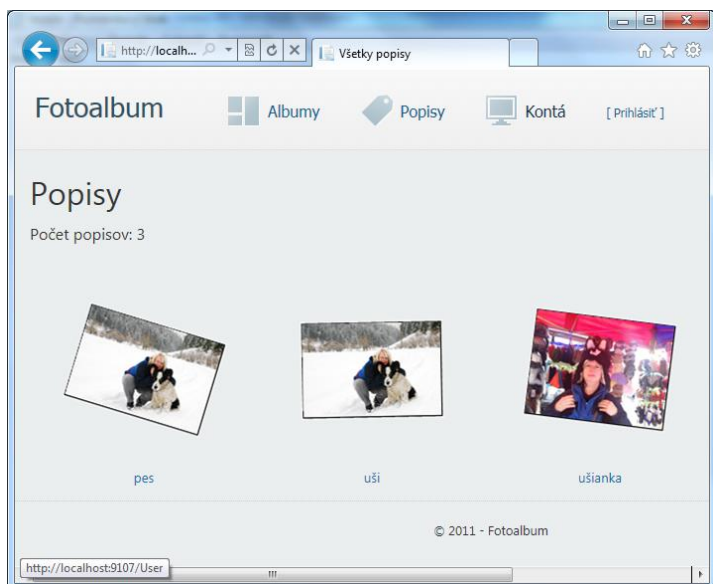
Odpoveď na túto otázku môžete nájsť po vyskúšaní aplikácie vytvorenej zo šablóny Gallery a zistíte, že vytváranie webových fotoalbumov vrátane správy používateľov už nemôže byť jednoduchšie. Návštevníci si môžu snímky prezerať a po zaregistrovaní aj komentovať, prípadne popisovať.



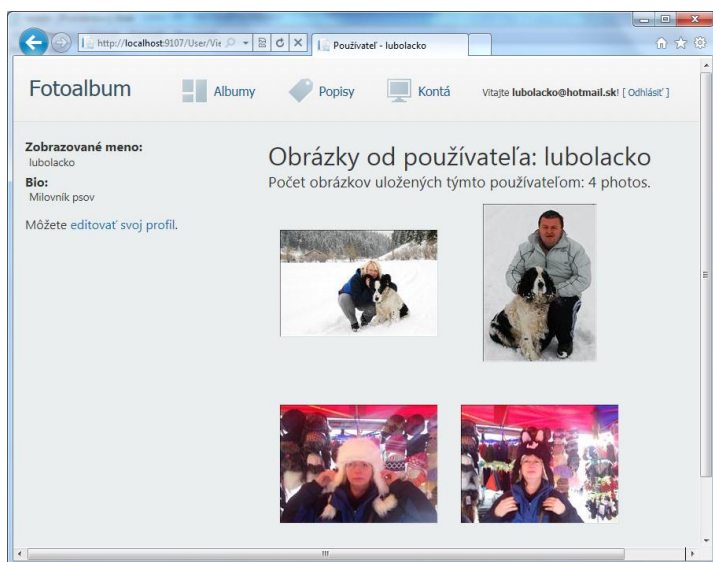
Pomocou šablóny Gallery si môžete vytvoriť webový album



Webový album umožňuje obrázky popisovať aj komentovať



Podľa popisov každý dokáže jednoducho nájsť obrázok ku ktorému sa popis viaže



Fotoalbum umožňuje kategorizovať aj obrázky od jednotlivých používateľov

Určite vás bude zaujímať, kam sa ukladajú obrázky. **Informácie o albumoch a obrázky sa ukladajú do databázy, ktorá je súčasťou aplikácie a nachádza sa v priečinku App_data. Obrázky sú fyzicky uložené v tabuľke Photos v atribúte FileContents. Tento atribút je typu image a je v ňom uložený binárny obsah súboru obrázka.**

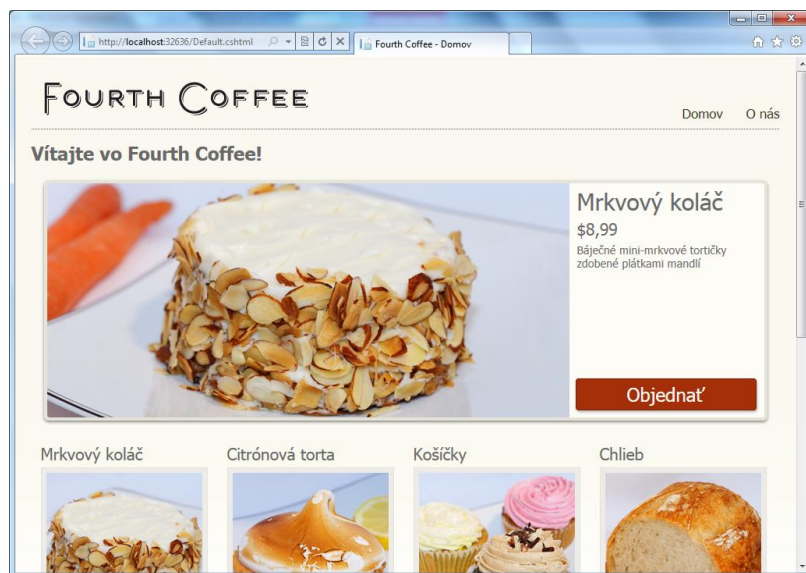
ContentType	Description	FileContents	FileExtension	FileSize	FileTitle	GalleryId
jpeg		<Binary Data>	.jpg	718177	WP_000038	1
jpeg		<Binary Data>	.jpg	825961	WP_000043	1
jpeg		<Binary Data>	.jpg	825827	WP_000046	1

Informácie o albumoch a obrázky sa ukladajú do databázy.

Šablóna Bakery – rozbehnite na Internete svoj biznis

Azda nikoho netreba presviedčať o tom, že Internet by sa zaručene nerozvinul do tej podoby ako ho poznáme dnes, bez komerčných aplikácií. Azda najčastejšou formou je aplikácia typu „internetový obchod“, ktorá umožňuje predávajúcemu prostredníctvom webových katalógov ponúkať svoj tovar a kupujúcim zasa nakupovať a samozrejme nejakou formou za tovar platiť. Pre tento účel bola vytvorená aj pomerne jednoduchá aplikácia Bakery (pekáreň).

Okrem aplikácií typu e-commerce je táto šablóna rovnako vhodná aj pre výmenné aktivity medzi zberateľmi (známky, zápalkové škatuľky).



Na hlavnej stránke aplikácie je aktuálna ponuka

Aplikácia funguje jednoducho. Používateľ v objednávacom formulári uvedie doručovaciu adresu, kam mu treba tovar doručiť a e-mailovú adresu pre zaslanie účtu. Platba sa dá vyriešiť napríklad na dobierku.

Formulár na objednávku tovaru

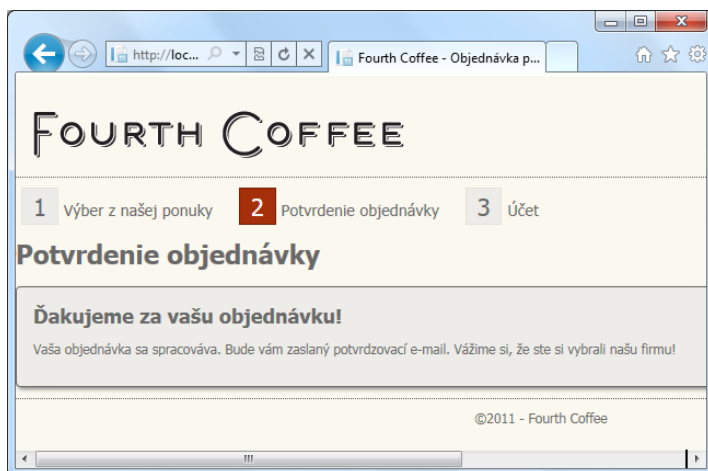
Pre odoslanie e-mailu je potrebné nastaviť v súbore Order.cshtml parametre e-mailového servera, v tomto prípade je podporovaný SMTP server a Hotmail.

Implicitne sú v kóde údaje nevyplnené:

```
//SMTP Configuration for Hotmail
WebMail.SmtpServer = "smtp.live.com";
WebMail.SmtpPort = 25;
WebMail.EnableSsl = true;

//Enter your Hotmail credentials for UserName/Password and a "From" address for the e-mail
WebMail.UserName = "";
WebMail.Password = "";
WebMail.From = "";
```

Pre odoslanie potvrdenia objednávky e-mailom je potrebné nastaviť parametre e-mailového servera



Potvrdenie objednávky e-mailom

Zákazníkovi príde potvrdenie objednávky e-mailom v tvare:

Ďakujem vám za vašu objednávku 1 kusov Košíčky!

Vaša adresa:

František Mrkvička

Mramorová 50

Lipt. Mikuláš

031 01

Celková suma je 5,99 EUR.

Ďakujeme! Ak budeme mať otázky týkajúce sa Vašej objednávky, budeme Vás kontaktovať.

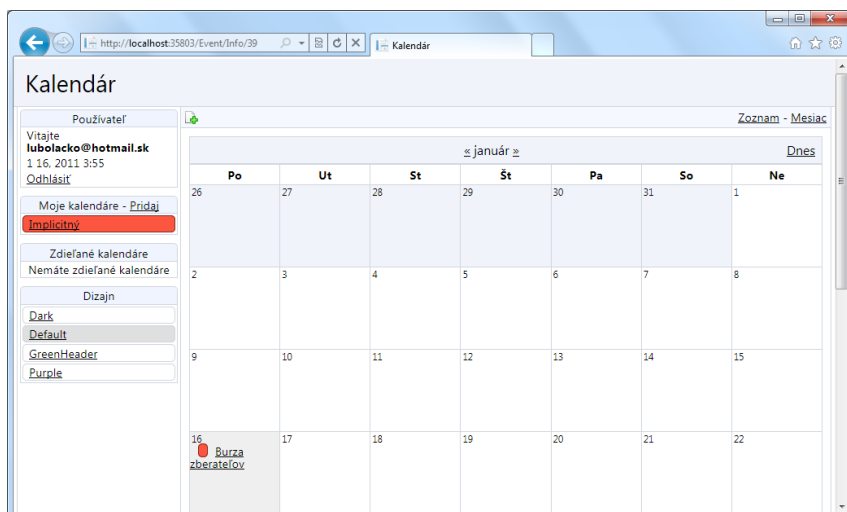
Zdôrazňujeme elegantné riešenie ukladania objednávok. Ak si všimnete databázu, aplikácia využíva len jednu databázovú tabuľku pre ukladanie údajov o produktoch. Údaje o objednávkach sa automaticky ukladajú v zozname odoslanej pošty, takže o túto vec sa ako prevádzkovateľ aplikácie vôbec nemusíte starať, stačí vytlačiť odoslané e-maily a máte zoznam objednávok, ktoré je potrebné vybaviť vrátane adries pre doručenie tovaru.

Table - (bakery.sdf).Products ✕					
	Id	Name	Description	Price	ImageName
▶	1	Mrkvový koláč	Báječné mini-mrkvové tortičky zdobené plátkami mandlí	8,99	carrot_cake.jpg
	2	Citrónová torta	Lahodný koláč s čerstvým citrónom	9,99	lemon_tart.jpg
	3	Košíčky	Delikátne vanilkové a čokoládové košíčky	5,99	cupcakes.jpg
	4	Chlieb	Čerstvý chlieb vo francúzskom štýle	1,49	bread.jpg
	6	Hrušková torta	Hrušková torta podávaná s plátkami mandlí a trochou škor...	5,99	pear_tart.jpg
	7	Čokoládový kol...	Fantastický mrazený čokoládový koláč	8,99	chocolate_cake...
*					

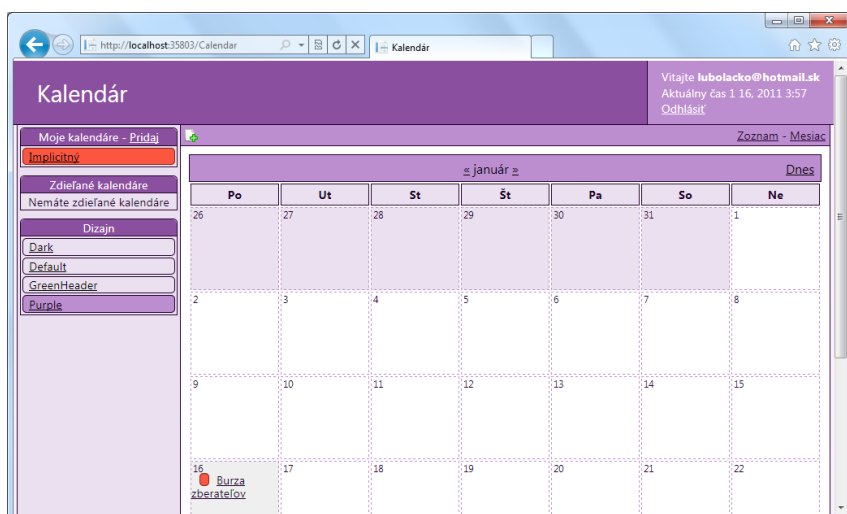
Databázová tabuľka obsahuje len katalóg produktov a odkazy na názvy obrázkov, ktoré sú fyzicky uložené v priečinku Images

Šablóna Kalendár – plánovanie podujatí pre členov tímu

V nadpise state sa zároveň skrýva aj odpoveď na otázku o význame kalendárovej šablóny. Pri webových aplikáciách pre rôzne hobby je pravdepodobne najdôležitejšia stránka avizujúca udalosti, ktorými príslušný klub žije a ktoré sú vo väčšine prípadov vlastne dôvodom existencie toho ktorého klubu, alebo záujmovej skupiny. V športových kluboch sú to zápasy, v záujmových združeniach podporujúcich rôzne záľuby sú to rôzne akcie, spoločenské udalosti, prípadne aj avízo na udalosti, ktoré sú pre členov klubu povinné, napríklad rôzne skúšky a uhrádzanie poplatkov v rybárskych zväzoch a podobne. Ak ste prihlásení k aplikácii ako vlastník kalendára, môžete do kalendára pridávať nové udalosti. Dátum začiatku a konca udalosti je možné zadávať priamo do editačného okna, alebo pomocou pohodlných kalendárových prvkov.



Zobrazenie kalendára



Používateľ si môže prispôbiť dizajn výberom témy

Pridaj udalosť	
Názov udalosti	Burza zberateľov
Popis udalosti	Výmenná burza zberateľských predmetov
Lokalita	Klubovňa Slávia
Celodenná udalosť	<input type="checkbox"/>
Začiatok	Datum: 16. 1. 2011 Čas: 13:00
Koniec	Čas: 16:00
Názov kalendára	Implicitný
Vytvor	

Vytvorenie novej udalosti

The screenshot shows a web browser window with the URL `http://localhost:35803/Calendar/Share/View.aspx?calendarid=...`. The page title is "Kalendár". In the top right corner, it says "Vitajte lubolacko@hotmail.sk", "Aktuálny čas 16. 1. 2011 4:00", and "Odhlásiť".

On the left sidebar, under "Moje kalendáre - Pridaj", the "Implicitný" calendar is selected. Below it, under "Zdieľané kalendáre", it says "Nemáte zdieľané kalendáre". Under "Dizajn", the "Purple" theme is selected.

The main content area is titled "Nastav zdieľanie pre: Implicitný". It contains a table for sharing settings:

E-mail	Úroveň oprávnenia	
Pridanie používateľa: <input type="text" value="lubolacko@pcrevue.sk"/>	<input type="text" value="Edit"/>	<input type="button" value="Pridaj"/>

Below the sharing settings, there is a monthly calendar view for "január". The days of the week are abbreviated as Po, Ut, St, Št, Pa, So, Ne. The dates 26, 27, 28, 29, 30, 31, and 1 are visible in their respective columns.

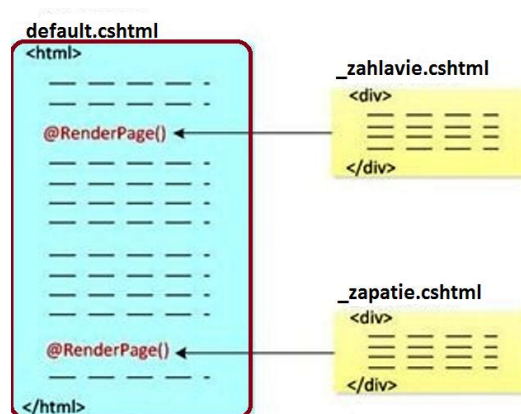
Kalendár je možné zdieľať s ostatnými členmi tímu, alebo záujmovej skupiny

Webové stránky ako stavebnica

Určite ste si pri mnohých webových aplikáciách všimli, že niektoré časti stránky majú viac-menej stály obsah, ktorý nejakým spôsobom dizajnovo aplikáciu charakterizuje a zároveň odlišuje od iných, takže pri surfovaní používateľov sa mení len určitá časť plochy stránky. Pre tvorcov webových aplikácií to znamená možnosť uľahčiť si prácu a časti stránky, ktoré sa zobrazujú na viacerých stránkach vytvárať len raz. Takáto dizajnová unifikácia uľahčuje návštevníkom stránky, aby sa na stránkach rýchlo zorientovali.

Razor preto poskytuje niekoľko príkazov, ktoré umožňujú opakované použitie častí obsahu na viacerých stránkach. Napríklad záhlavie a dolnú časť stránky s názvom firmy stačí navrhnuť len raz a vkladať ju z pripravenej šablóny do všetkých stránok. Podobne blok s informáciou, že je používateľ prihlásený.

Na rozdiel od nakopírovania častí kódu do všetkých stránok v prípade nutnosti úprav stačí urobiť zmeny len na jednom mieste v šablóne a tie sa okamžite prejavia na všetkých stránkach, ktoré šablónu využívajú.



Princíp stavebnicovej koncepcie webových stránok

@RenderPage sa hodí na renderovanie častí stránky, ktoré sa príliš nemenia, napríklad záhlavia, dolnej informačnej časti a podobne.

Vytvorte dve šablóny (súbory cshtml) pre záhlavie a dolnú časť:

`_zahlavie.cshtml`

```
<div class="header">
  Logo a obchodné informácie mojej firmy.
</div>
```

`_zapatie.cshtml`

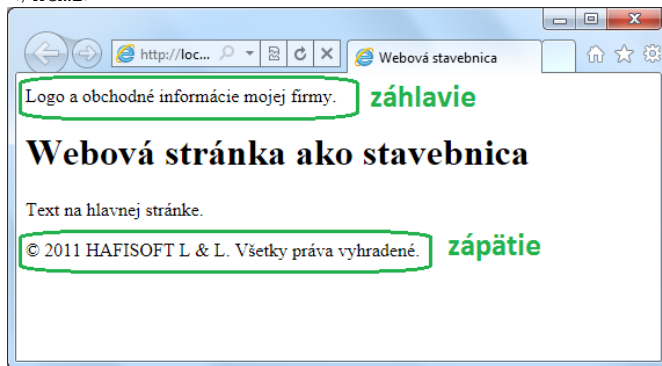
```
<div class="footer">
  © 2011HAFISOFTL & L. Všetky práva vyhradené.
</div>
```

Možno ste si všimli, že šablóny v príklade začínajú podčiarknikom (_). ASP.NET engine takéto stránky nedokáže poslať do klientovho prehľadávača, takže sa používatelia na tieto stránky či už náhodne, alebo úmyselne nedokážu priamo dostať.

Znovu použiteľné bloky sa budú vkladať do hlavnej stránky pomocou príkazu **RenderPage**:

default.cshtml

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>Webová stavebnica</title>
</head>
<body>
    @RenderPage("_zahlavie.cshtml")
<h1>Webová stránka ako stavebnica</h1>
<p>Text na hlavnej stránke.</p>
    @RenderPage("_Zapatie.cshtml")
</body>
</html>
```



Zloženie webovej stránky z pripravených opakovateľne použiteľných častí

Zdieľané časti stránok môžete umiestniť do vhodne pomenovaného priečinka, v našom prípade Shared. Potom sa k nim pristupuje takto:

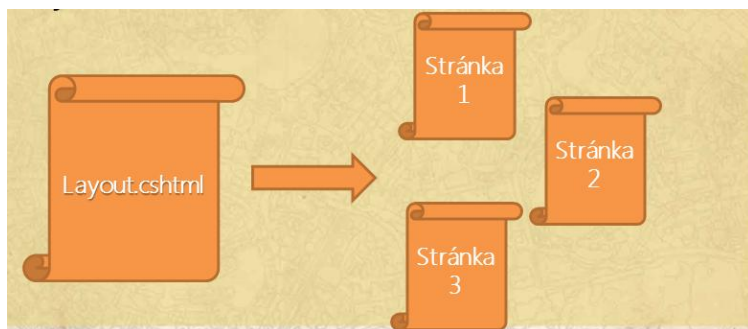
```
@RenderPage("/Shared/_zahlavie.cshtml")
```

Ak vás zaujíma, ako sa obsah šablón začlenil do HTML stránky vyrendrovanej pre klienta:

```
http://localhost:51618/default.cshhtml - Original Source
File Edit Format
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <title>Webová stavebnica</title>
6   </head>
7   <body>
8     <div class="header">
9       Logo a obchodné informácie mojej firmy.
10    </div>
11
12    <h1>Webová stránka ako stavebnica</h1>
13    <p>Text na hlavnej stránke.</p>
14    <div class="footer">
15      © 2011 HAFISOFT L & L. Všetky práva vyhradené.
16    </div>
17
18  </body>
19 </html>
20
21
22
```

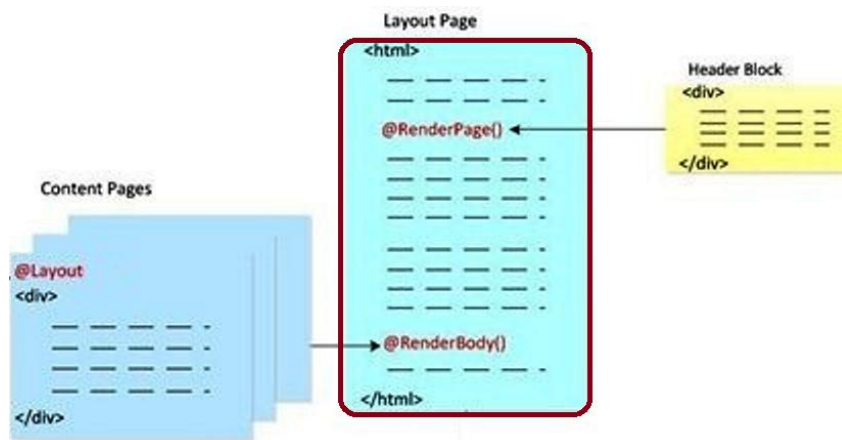
Kód HTML vygenerovaný pre klienta

V predchádzajúcom príklade sme vkladali šablóny do jednotlivých stránok. V mnohých prípadoch bude výhodnejší opačný postup. Dizajn všetkých stránok bude určovať šablóna a meniť sa bude obsah vkladajú do unifikovanej šablóny.



Princíp zostavovania stránky

Inak povedané, v predchádzajúcom príklade sa vkladal rovnaký obsah do viacerých stránok. V tomto príklade sa bude najskôr vytvárať rozvrhnutie stránky, ktoré definuje jej štruktúru, ale neobsahuje žiadny konkrétny obsah. Ten je v kontextových stránkach, ktoré sa prepoja na rozloženie stránky.



Stránka sa zostavuje z kontextových stránok a šablón

Pre naznačený účel sú k dispozícii dve funkcie:

RenderBody zobrazuje stránku s aktuálnym obsahom. Jej názov musí byť priradený do „Layout“ premennej.

@RenderSection(“názov”) renderovanie obsahu pomenovanej sekcie

Dizajn všetkých stránok určuje šablóna. Aby sa na ňu používateľ priamo nedostal, jej názov začína podčiarkovníkom.

_Dizajn.cshtml

```
<html>
<head>
<title>Viacnásobný obsah</title>
<link href="@Href("/Styles/Stranka.css")" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="header">
    @RenderSection("header")
</div>
<div id="list">
    @RenderSection("list")
</div>
<div id="main">
    @RenderBody()
</div>
<div id="footer">
    © 2011 HAFISOFT L & L. Všetky práva vyhradené.
</div>
</body>
</html>
```

Aby bola stavebnica naozaj univerzálna, záhlavie a pätička sa do kontextovej šablóny vložia z osobitných šablón.

Štýl zobrazovaných stránok je definovaný pomocou css súboru v priečinku Styles:

Stranka.css

```

h1 {
    border-bottom: 3px solid darkgrey;
    font: 2.75em/1.75em Georgia, serif;
    color: green;
}

ul {list-style-type: none;}

body {
    margin: 0; padding: 1em;
    background-color: white;
    font: 75%/1.75em "Trebuchet MS", Verdana, sans-serif;
    color: brown;
}

#list {
    margin: 1em 0 7em -3em; padding: 1em 0 0 0;
    background-color: #ffffff; color: #996600;
    width: 25%; float: left;
}

#header, #footer {margin: 0; padding: 0; color: #996600;}

```

Vkladaný obsah je v samostatných súboroch:

Obsah1.cshtml

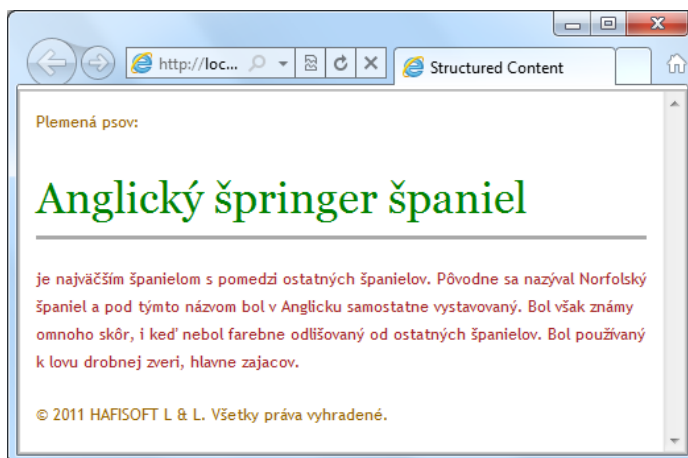
```

@{
    Layout = "_Dizajn.cshtml";
}

@section header {
<div id="header">
<h1> Anglický špringer španiel</h1>
</div>
}

<p>je najväčším španielom s pomedzi ostatných španielov.
    Pôvodne sa nazýval Norfolkský španiel a pod týmto názvom
    bol v Anglicku samostatne vystavovaný. Bol však známy
    omnoho skôr, i keď nebol farebne odlišovaný od ostatných
    španielov. Bol používaný k lovu drobnej zveri, hlavne zajacov.</p>

```

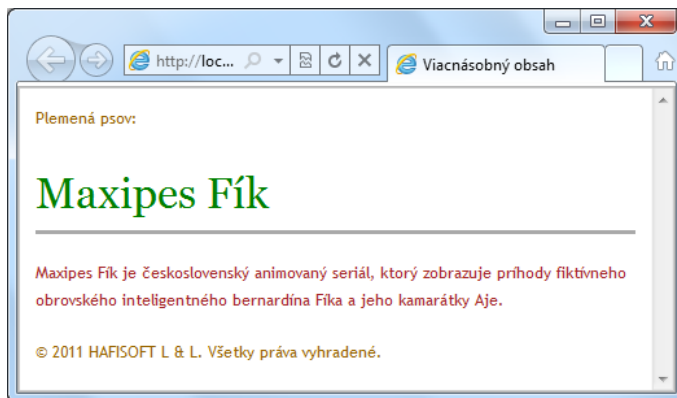


Zobrazenie

Obsah2.cshtml

```
@{
    Layout = "_Dizajn.cshtml";
}

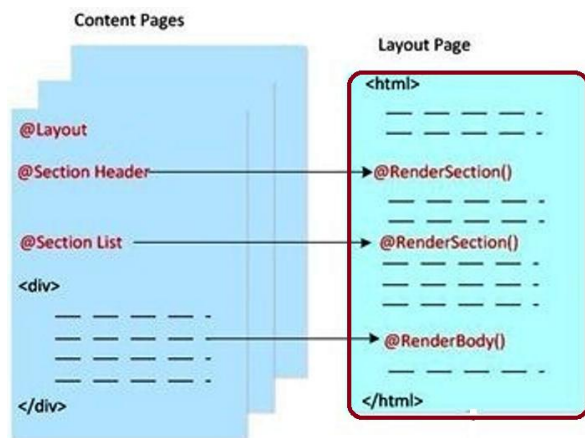
@section header {
<div id="header">
<h1>Maxipes Fík </h1>
</div>
}
<p>Maxipes Fík je československý animovaný seriál, ktorý zobrazuje
    príhody fiktívneho obrovského inteligentného bernardína Fíka
    a jeho kamarátky Aje.</p>
```



Zobrazenie

Stránky s viacerými sekciami

Webové stránky môžu mať viac sekcií. Ich poloha je definovaná v šablóne rozloženia. Nasledujúci príklad využíva dve sekcie, pričom v jednej je zoznam a v druhom je popis.



Princíp

_Dizajn1.cshtml

```
<html>
<head>
<title>Viacnasobný obsah</title>
<link href="@Href("/Styles/Stranka.css")" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="header">
    @RenderSection("header")
</div>
<div id="list">
    @RenderSection("list")
</div>
<div id="main">
    @RenderBody()
</div>
<div id="footer">
    © 2011 HAFISOFT L & L. Všetky práva vyhradené.
</div>
</body>
</html>
```

Obsah1.cshtml

```
@{
    Layout = "/Shared/_Dizajn1.cshtml";
}

@section header {
<div id="header">
    Zoznam:
</div>
}

@section list {
<ul>
<li>be</li>
<li>become</li>
<li>begin</li>
<li>break</li>
<li>bring</li>
<li>build</li>
<li>burn</li>
</ul>
}

<h1>Nepravidelné slovesá</h1>
<p>Ich minulý čas a minulé prítomnosť sa môžu zhodovať aj odlišovať.
    Pri niektorých slovesách je nepravidelné iba minulé prítomnosť.
    Niekedy možno použiť pravidelný aj nepravidelný tvar.</p>
```



Princíp

Voliteľné zobrazovanie sekcií

Zobrazenie jednotlivých sekcií môže byť voliteľné, čo umožní definovať jednotné rozloženie pre viac stránok, pričom niektoré z nich nemusia niektorú z častí zobrazovať.

_Dizajn.cshtml

```
<html>
<head>
<title>Viacnasobný obsah</title>
<link href="@Href("/Styles/Stranka.css")" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="header">
    @RenderSection("header")
</div>
    @if (PageData["ShowList"] == true) {
<div id="list">
    @RenderPage("/Shared/_Zoznam.cshtml")
</div>
    }
</div>
<div id="main">
    @RenderBody()
</div>
<div id="footer">
    (c) 2011 HAFISOFT L & L. Všetky práva vyhradené.
</div>
</body>
</html>
```

_Zoznam.cshtml

```
<ul>
<li>be</li>
<li>become</li>
<li>begin</li>
<li>break</li>
<li>bring</li>
<li>build</li>
<li>burn</li>
</ul>
```

Obsah3.cshtml

```
@{
    Layout = "/Shared/_Dizajn2.cshtml";

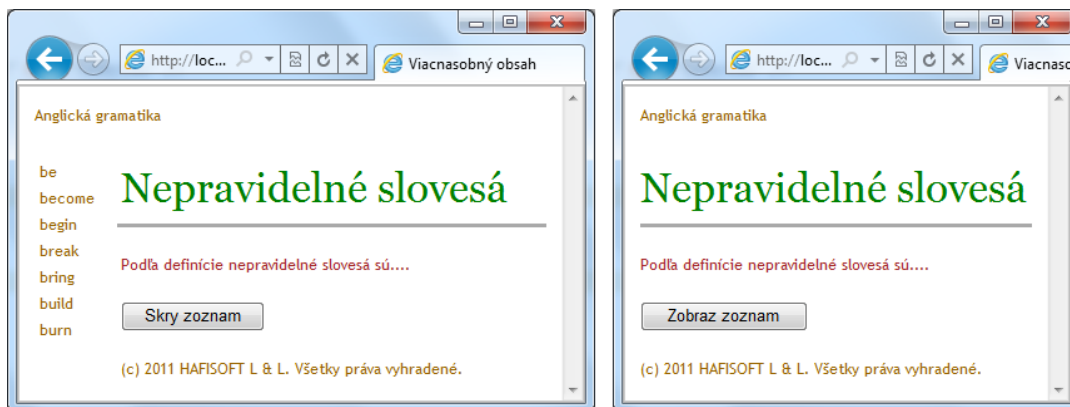
    PageData["Title"] = "Nepravidelné slovesá";
    PageData["ShowList"] = true;

    if (IsPost) {
        if (Request["list"] == "off") {
            PageData["ShowList"] = false;
        }
    }
}

@section header {
<div id="header">
    Anglická gramatika
</div>
}

<h1>@PageData["Title"]</h1>
<p>Podľa definície nepravidelné slovesá sú....</p>

@if (PageData["ShowList"] == true)
{
    <form method="post" action="">
        <input type="hidden" name="list" value="off" />
        <input type="submit" value="Skry zoznam" />
    </form>
}
else {
    <form method="post" action="">
        <input type="hidden" name="list" value="on" />
        <input type="submit" value="Zobraz zoznam" />
    </form>
}
```



Zobrazenie a skrytie sekcie

Práca s údajmi v súboroch a objektoch

Programátori by si v nedávnej minulosti asi nedokázali ani predstaviť fungovanie menších aplikácií bez toho, aby ukladali údaje do súborov, prípadne tieto údaje zo súborov čítali. V súčasnosti pre veľké objemy údajov a ich bezpečné a spoľahlivé uloženie jednoznačne preferujeme databázy, no aj tak existuje veľké množstvo webových aplikácií, ktoré si niekam potrebujú uložiť určité menšie množstvo údajov, takže klasický databázový server by mohol pôsobiť ako kanón na vrabce. Niekedy sa pre malý objem údajov neoplatí využiť ani embedded databázu. V takom prípade môžete údaje ukladať do premenných, polí, súborov, či XML dokumentov.

V niektorých prípadoch potrebujete ukladať údaje premenných typu **pole**, pričom tieto údaje sú nejakým spôsobom z hľadiska aplikačnej logiky spolu zviazané. Na rozdiel od premennej môže pole obsahovať viac hodnôt. Každý prvok poľa má okrem hodnoty aj svoj index, pomocou ktorého je prvok poľa jednoznačne identifikovaný.

```
@{
<h3>Znamenia zverokruhu</h3>
    string[] znamenania = {"Vodnár", "Ryby", "Baran", "Býk",
                           "Blíženci", "Rak", "Lev", "Panna",
                           "Váhy", "Škorpión", "Strelec", "Kozorožec"};
    foreach (var znamenanie in znamenania)
    {
<p>@znamenanie</p>
    }
}
```

Dictionary – dvojice k názvom priradených hodnôt

```
@{
    var dKatalog = new Dictionary<int, string>();
    dKatalog.Add(1, "Myš GF33");
    dKatalog.Add(2, "Klávesnica KB4");
    dKatalog.Add(3, "Web kamera Logitech");
    dKatalog.Add(4, "USB hub 6x");
}
<p>Položka č 3: @dKatalog[3]</p>
@ (dKatalog[4] = "USB hub 8x")
<p>Upravená položka č. 4 je: @dKatalog[4]</p>
```

Súbor údaje.txt obsahuje niekoľko riadkov údajov, pričom jednotlivé atribúty sú oddelené čiarkou:

```
Miroslav,Zemko,mirozem@mojafirma.sk
Zuzana,Fegyeres,zuzafe@mojafirma.sk
Fedor,Macko,fedomacko@gmail.com
Adam,Berka,aberka@mojafirma.sk
Fedor,Futik,fufu@post.sk
```

Údaje v tomto tvare získate napríklad exportom z aplikácie Microsoft Excel do formátu csv (comma-separatedfile).

Upozornenie: Údaje na serveri v žiadnom prípade neumiestňujte do absolútnych priečinkov napr. C:\Folder\File.txt , ale do priečinka viažuceho sa k aplikácii, napríklad App_data. Prepojenie na súbor potom bude v tvare /App_Data/data.txt.

V nasledujúcich príkladoch ukážeme niekoľko typických úkonov pre prácu s údajmi v súboroch.

Zobrazenie údajov zo súboru

```
@{
    var result = "";
    Array userData = null;
    char[] delimiterChar = {' ',' '};

    var dataFile = Server.MapPath("/App_Data/udaje.txt");

    if (File.Exists(dataFile)) {
        userData = File.ReadAllLines(dataFile);
        if (userData == null) {result = "Súbor neobsahuje údaje.";}
    }
    else {result = "Súbor neexistuje."; }
}

<html xmlns="http://www.w3.org/1999/xhtml" lang="sk" xml:lang="sk">
<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1250">
<title>Čítanie údajov zo súboru</title>
</head>
<body>
<div>
<h1>Čítanie údajov zo súboru</h1>
        @result
        @if (result == "") {
<ol>
            @foreach (string dataLine in userData) {
<li>
                Kontakt
<ul>
                    @foreach (string dataItem in dataLine.Split(delimiterChar)) {
<li>@dataItem</li >
                }
</ul>
</li>
            }
</ol>
        }
</div>
</body>
</html>
```

Čítanie údajov zo súboru

1. Kontakt
 - o Miroslav
 - o Zemko
 - o mirozem@mojafirma.sk
2. Kontakt
 - o Zuzana
 - o Fegyeres
 - o zuzafe@mojafirma.sk
3. Kontakt
 - o Fedor
 - o Macko
 - o fedomacko@gmail.com

Zobrazenie údajov zo súboru

Vkladanie údajov do textového súboru

```
@{
    var result = "";
    if (IsPost)
    {
        var firstName = Request["Meno"];
        var lastName = Request["Priezvisko"];
        var email = Request["Email"];

        var userData = firstName + "," + lastName +
            "," + email + Environment.NewLine;

        var dataFile = Server.MapPath("/App_Data/data.txt");
        File.WriteAllText(dataFile, userData);
        result = "Údaje sú uložené.";
    }
}

<html>
<head>
<title>Vkladanie údajov do textového súboru</title>
</head>
<body>
<form id="form1" method="post">
<div>
<table>
<tr>
<td>Meno:</td>
<td><input id="Meno" name="Meno" type="text" /></td>
</tr>
<tr>
<td>Priezvisko:</td>
<td><input id="Priezvisko" name="Priezvisko" type="text" /></td>
</tr>
<tr>
<td>Email:</td>
<td><input id="Email" name="Email" type="text" /></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Ulož"/></td>
</tr>
</table>
```

```

</div>
<div>
    @if(result != ""){<p>Výsledok: @result</p>}
</div>
</form>
</body>
</html>

```

Meno:

Priezvisko:

Email:

Vkladanie údajov do textového súboru

Upload súborov na server

ASP.NET umožňuje spracovať upload textového aj binárneho súboru z klientskeho počítača na server. Najskôr ukážeme jednoduchý príklad pre upload jedného súboru.

Pre fungovanie posielania súborov na server je potrebné nainštalovať doplnok ASP.NET Web HelpersLibrary.

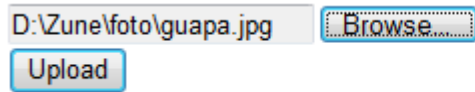
```

@{
    var fileName = "";
    if (IsPost) {
        var fileSavePath = "";
        var uploadedFile = Request.Files[0];
        fileName = Path.GetFileName(uploadedFile.FileName);
        fileSavePath = Server.MapPath("/App_Data/" +
            fileName);
        uploadedFile.SaveAs(fileSavePath);
    }
}

<html>
<head>
<title>Upload súborov na server</title>
</head>
<body>
<h1>Upload jedného súboru na server</h1>
    @FileUpload.GetHtml(
        initialNumberOfFiles:1,
        allowMoreFilesToBeAdded:false,
        includeFormTag:true,
        uploadText:"Upload")
    @if (IsPost) {
<span>Súbor bol prenesený!</span><br/>
    }
</body>
</html>

```

Upload jedného súboru na server



Upload jedného súboru na server

Niekedy potrebuje používateľ preniesť viac súborov, napríklad kolekciu fotografií a podobne:

```
@{
    var message = "";
    if (IsPost) {
        var fileName = "";
        var fileSavePath = "";
        int numFiles = Request.Files.Count;
        int uploadedCount = 0;
        for(int i =0; i < numFiles; i++) {
            var uploadedFile = Request.Files[i];
            if (uploadedFile.ContentLength > 0) {
                fileName = Path.GetFileName(uploadedFile.FileName);
                fileSavePath = Server.MapPath("/App_Data/" +
                    fileName);
                uploadedFile.SaveAs(fileSavePath);
                uploadedCount++;
            }
        }
        message = "Upload dokončený. Počet prenesených súborov: " +
            uploadedCount.ToString();
    }
}

<html>
<head><title>Upload viacerých súborov na server</title></head>
<body>
<form id="myForm" method="post"
    enctype="multipart/form-data"
    action="">

<div>
<h1>Upload viacerých súborov na server</h1>
    @if (!IsPost) {
        @FileUpload.GetHtml(
            initialNumberOfFiles:2,
            allowMoreFilesToBeAdded:true,
            includeFormTag:true,
            addText:"Pridajte ďalší súbor",
            uploadText:"Upload")
    }
<span>@message</span>
</div>
</form>
</body>
</html>
```

Upload viacerých súborov na server

D:\Zune\foto\cloud1.jpg	<input type="button" value="Browse..."/>
D:\Zune\foto\guapa.jpg	<input type="button" value="Browse..."/>
D:\Zune\foto\Babylon_Tow	<input type="button" value="Browse..."/>
Pridajte ďalší súbor	<input type="button" value="Upload"/>

Upload viacerých súborov na server

Obrázky a multimédiá

Zobraziť statický obrázok na HTML stránke je triviálna záležitosť, no pri moderných dynamických webových aplikáciách však so statickými obrázkami nevystačíte, vo väčšine prípadov budete chcieť zobraziť obrázok v závislosti od konkrétnej situácie.

Dynamické zobrazovanie vybraných obrázkov

```
@{ var imagePath= "";

    if( Request["vyberObrazka"] != null){
        imagePath = @"Images\" + Request["vyberObrazka"];
    }
}

<html>
<head><title>Obrázky</title></head>
<body>
<h1>Dynamické zobrazovanie obrázkov</h1>
<form method="post" action="">
<div>
    Zobraz psíka:
    <select name="vyberObrazka">
    <option value="springer1.jpg">Obrázok 1</option>
    <option value="springer2.jpg">Obrázok 2</option>
    <option value="springer3.jpg">Obrázok 3</option>
    </select>

    <input type="submit" value="Vyber" />
</div>
<div style="padding:10px;">
    @if(imagePath != ""){
    
    }
</div>
</form>
</body>
</html>
```



Dynamické zobrazovanie obrázkov

Upload obrázkov na server

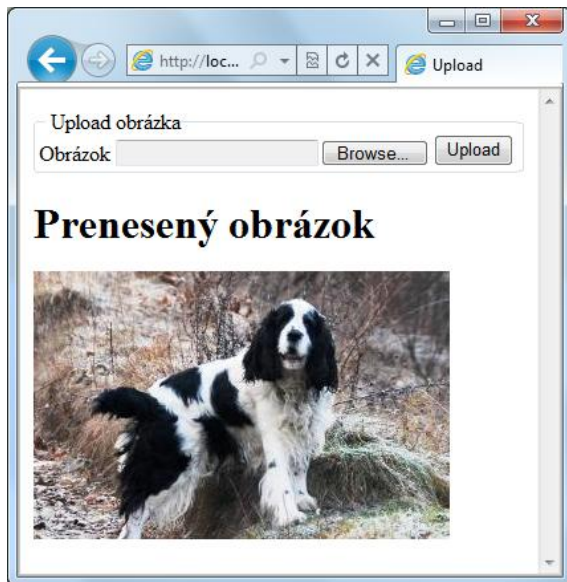
V predchádzajúcej stati sme ukázali upload súboru z počítača klienta na server. Ak prenášate obrázky môžete ich následne po prenose ihneď zobraziť. Naším cieľom je teda preniesť obrázok a na stránke pre klienta vygenerovať HTML tag pre zobrazenie obrázka:

```
@{ WebImage obr = null;
    var MenoSuboru = "";
    var obrDir = "";

    if(IsPost){
        obr = WebImage.GetImageFromRequest();
        if(obr != null){
            MenoSuboru = Guid.NewGuid().ToString() + "_" +
                Path.GetFileName(obr.FileName);
            obrDir = @"Images\" + MenoSuboru;
            obr.Save(@"~\" + obrDir);
        }
    }
}
```

```
<html>
<head><title>Upload</title></head>
<body>
<form action="" method="post" enctype="multipart/form-data">
<fieldset>
<legend> Upload obrázka </legend>
<label for="Image">Obrázok</label>
<input type="file" name="Image" />
<input type="submit" value="Upload" />
</fieldset>
</form>
<h1>Prenesený obrázok</h1>
    @if(obrDir != "")
    {<div class="result">

</div>
    }
</body>
</html>
```



Upload obrázkov

Pre viac know-how ohľadne práce s obrázkami odporúčame pozrieť si vzorovú aplikáciu zo šablóny Gallery.

Prehrávanie videa

Doplnok Video umožňuje prehrávanie videa pomocou prehrávačov formátov:

- Adobe Flash(.swf),
- Windows MediaPlayer(.wmv),
- Microsoft Silverlight(.xap).

Pre fungovanie prehrávania videa je potrebné nainštalovať doplnok ASP.NET Web HelpersLibrary.

```
<body>
    @Video.MediaPlayer(
        path: "Wildlife.wmv",
        width: "400", height: "250",
        autoStart: true, playCount: 2,
        uiMode: "full",
        stretchToFit: true,
        enableContextMenu: true,
        mute: false, volume: 75)
</body>
```



Prehrávanie videa

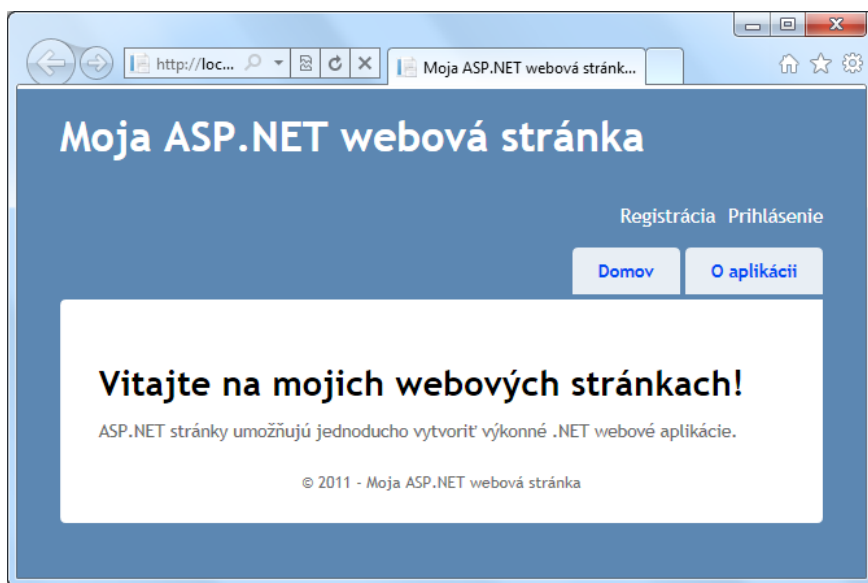
Len pre zaregistrovaných členov

Väčšina webových aplikácií umožňuje voľný pohyb návštevníkov takmer po celom webovom sídle. Je to pochopiteľné, ich prevádzkovatelia sa pomocou webovej aplikácie snažia niečo propagovať. Personálne stránky zas vo väčšine prípadov propagujú hobby svojho majiteľa. Vo všeobecnosti býva mottom webovej aplikácie zverejňovať a zviditeľňovať. V takýchto prípadoch sa otázkou autentifikácie a autorizácie vôbec nemusíte trápiť. Sú však aj aplikácie, kedy chcete niektorých používateľov presunúť do určitej „privátnej zóny“, napríklad pre prezeranie rodinných fotografií, či zverejňovanie informácií len pre nejakú komunitu.

Z iných webových aplikácií poznáte ako to funguje. Používateľ ak chce pristupovať k privátnym informáciám, musí sa prihlásiť spravidla zadaním prihlasovacieho mena a hesla. Tento proces sa nazýva **autentifikácia** a odohráva sa v prihlasovacom dialógu. V zásade sú dve cesty, ako sa používateľ môže k tomuto dialógu nasmerovať. Buď na úvodnú stránku aplikácie, kde je povolený anonymný prístup umiestnite prepojenie na prihlásenie, alebo nepriamo, ak sa používateľ pokúsi o prístup priamo na chránenú stránku, bude presmerovaný najskôr na autentifikačný dialóg. Na základe autentifikačných údajov sa zistí, či príslušný používateľ má alebo nemá k požadovanej stránke prístup. Tento proces sa nazýva **autorizácia**.

Využite šablónu aplikácie StarterSite

Programátori webových stránok nám iste dajú za pravdu, že jednou z najčastejšie sa vyskytujúcich úloh pri riešení každej aplikácie je vytvorenie používateľského rozhrania pre prihlásenie sa používateľa k aplikácii a veci s tým súvisiace, teda upozornenie v prípade zadania nesprávnych údajov a podobne. Pre uľahčenie týchto rutinných záležitostí obsahuje WebMatrix **StarterSite** šablónu, kde sú už prakticky všetky funkcie súvisiace s prihlasovaním a registráciou nových členov implementované.



Hlavná stránka šablóny StarterSite

Po spustení „prázdnej“ aplikácie zistíte, že v ľavom hornom rohu stránky sú prepojenia na registráciu nových členov a prihlásenie členov, ktorí sú už zaregistrovaní. Pomocou prepojenia Register zaregistrujte niekoľko (cvičných) používateľov.

Formulár na vytvorenie nového používateľa vyžaduje zadanie používateľského mena a hesla. Používateľské meno je v tvare adresy elektronickej pošty.

Moja ASP.NET webová stránka

Registrácia Prihlásenie

Domov O aplikácii

Registrácia konta

Použite formulár pre vytvorenie konta.

Formulár pre prihlásenie

E-mail:

Heslo:

Potvrdenie hesla:

Pre aktiváciu CAPTCHA verifikácie [nainštalujte ASP.NET Web Helpers Library](#) odstráňte komentár z ReCaptcha.GetHtml a nahrad'te 'PUBLIC_KEY' vašim verejným kľúčom. V hornej časti kódu stránky, odstráňte komentár z ReCaptcha.Validate a nahrad'te 'PRIVATE_KEY' vašim privátnym kľúčom, a odstráňte komentár z premennej captchaMessage.

Registrujte sa pre získanie reCAPTCHA kľúčov na reCAPTCHA.net.

Pri registrácii je možné využívať aj reCAPTCHA.

reCAPTCHA – užitočná ochrana pred spamovými robotmi

Môžete použiť aj systém na ochranu pred spamovými robotmi. Pod skratkou CAPTCHA (<http://www.google.com/recaptcha>) sa skrýva zložitý pojem CompletelyAutomatedPublicTuring test to tellComputers and HumansApart, alebo po našom automatický verejný Turingov test na odlíšenie počítačov od ľudí. V princípe ide o počítačom generovaný test, ktorý počítač vyriešiť nevie, ale človek áno. Používa sa hlavne pri registrácii e-mailových adries ako ochrana proti spamom. Všetko vysvetlí obrázok.

Zadajte znaky, ktoré vidíte

[Nové](#) | [Zvuk](#) | [Pomocník](#)

Obecton century

Typický príklad CAPTCHA z registračnej stránky Hotmail.sk.

Dovolíme si uviesť malú štatistiku k tomuto najpoužívanejšiemu filtru vo svete, ktorý bezpečne rozlíši väčšinou spamových robotov od ľudí. Denne sa na Internete zadá 60 miliónov týchto slov, priemerný čas na napísanie správneho slova je 10 sekúnd a súčet časov medzi zobrazením a zapísaním správnej odpovede je denne 150 000 hodín. Rozhodne to nie je zbytočne vynaložené úsilie. [Napísaním jedného z dvoch slov pomáhate pri digitalizácii kníh](#). Ide o úlohu nazvanú reCAPTCHA. Jedno zo slov, ktoré prepisujete, totiž pochádza z kníh, ktoré sa digitalizujú takže okrem ochrannej funkcie plní reCAPTCHA aj prepis tých slov z kníh, ktoré sa nedajú rozpoznať ORC algoritmom. Požívateľovi sa zobrazujú dve slová, z ktorých jedno je známe a druhé je zatiaľ nerozpoznané. Prvé z nich má bezpečnostnú funkciu, takže ak ho používateľ správne zadá, dá sa predpokladať, že správne zadal aj druhé slovo z knihy. Napríklad na našom obrázku je prvé slovo z digitalizovanej knihy a druhé bolo vygenerované počítačom.

Ak chcete reCAPTCHA využívať, vytvorte si konto na <http://www.google.com/recaptcha>. Nakoľko pri vytváraní konta musíte zadať názov domény, musíte mať jasno, kde budete svoju aplikáciu na webe hostovať. Pri registrácii získate PUBLIC_KEY. Tento verejný kľúč zadajte do kódu v súbore **Register.cshtml** (nachádza sa v podpričinku Account). Dva bloky kódu na overenie reCAPTCHA sa nachádzajú na začiatku a konci súboru a sú implicitne v komentári, ktorý musíte odstrániť:

```
// Validate the user's captcha answer
// if (!ReCaptcha.Validate("PRIVATE_KEY")) {
//     captchaMessage = "Captcha response was not correct";
//     isValid = false;
// }
...
// Validate the user's captcha answer
@*@ReCaptcha.GetHtml("PUBLIC_KEY", theme: "white")
@if (!captchaMessage.IsNullOrEmpty()) {
<label class="validation-error">@captchaMessage</label>
}*@
```

Ak tento systém overovania nechcete použiť, dajte do komentára blok kódu, ktorý vypisuje oznam na stránke pre registráciu nových používateľov:

```
<li class="recaptcha">
...
</li>
```

Aby sa na „súkromnú“ stránku aplikácie dostali len zaregistrovaní používatelia, treba na jej začiatok pridať kód:

```
@if (!WebSecurity.IsAuthenticated)
{
    Response.Redirect("~/Account/Login");
}
```

Môžete si to vyskúšať napríklad na stránke About.cshtml, ktorá je súčasťou projektu StarterSite.

Tip: O tom, či je formulár na vytvorenie nového používateľa funkčný sa môžete presvedčiť nahliadnutím do tabuliek databázy, ktorá je súčasťou aplikácie.

Role

Ak začnete vytvárať štruktúru používateľov pre zložitejší web, čoskoro pridáte na to, že sa dajú špecifikovať skupinky používateľov, ktoré majú rovnaké práva. Pre takéto skupinky môžete definovať takzvané role. Viacero používateľov môže byť priradených k jednej role, ale aj opačne, jeden používateľ môže mať prístup k viacerým rolám.

Určite vás pri testovaní aplikácie zo šablóny StarterSite napadla otázka, kam sú údaje získané pri registrácii používateľov ukladané. Sú v lokálnej databáze, ktorá je umiestnená v podpriečinku „App_Data“.

Prepnite sa do režimu Databases. V databáze (súbor s príponou.sdf) sú tabuľky:

- UserProfile,
- webpages_Roles,
- webpages_UserInRoles,
- webpages_Membership.

Tieto tabuľky sa vytvorili automaticky pri generovaní projektu zo šablóny.

Email	UserId
lubolacko@hotmail.sk	2
llacko@pcrevue.sk	3
*	


Tabuľka UserProfile

Pridajte role do tabuľky webpages_Roles. **Zadáвайте len názov role do stĺpca RoleName. Hodnota do stĺpca RoleId bude vložená automaticky.**

RoleId	RoleName
7	Veduci
6	Člen
8	Organizátor
NULL	NULL

Pridávanie rolí do tabuľky webpages_Roles

Tabuľka **webpages_UserInRoles** pomocou atribútov **UserId** a **RoleId** určuje vzťah medzi používateľmi v tabuľke **UserProfile** a rolami v tabuľke **webpages_Roles**.

	UserId	RoleId
	2	6
	3	7
*		

Pridávanie používateľov do rolí v tabuľke webpages_UserInRoles

Ak chcete povoliť na niektorú stránku prístup len pre používateľov určitej role, umiestnite na začiatok stránky kód v tvare:

```
@if ( Roles.IsUserInRole("nazov_role"))
{
    Welcome @WebSecurity.CurrentUserName!
}
else {
    Response.Redirect("/StrankaNaPresmerovanie");
}
```

Obnovenie zabudnutého hesla

Ak používateľ zabudne heslo, môže požiadať o jeho obnovenie. Nové heslo mu bude automaticky poslané e-mailom.

Forget Your Password?

Password Reset Instructions Form

Password recovery is disabled for this website because the SMTP server is not configured correctly. Please contact the owner of this site to reset your password.

Aby fungovala obnova hesla, musíte v aplikácii zadať SMTP Server

Parametre pre riadenie prístupu sa nastavujú v súbore **_AppStart.cshtml**:

```
@{
    WebSecurity.InitializeDatabaseConnection("StarterSite", "UserProfile", "UserId", "Email",
true);
    // WebMail.SmtpServer = "mailserver.example.com";
    // WebMail.EnableSsl = true;
    // WebMail.UserName = "username@example.com";
    // WebMail.Password = "your-password";
    // WebMail.From = "your-name-here@example.com";
}
```

Práca s údajmi v databáze

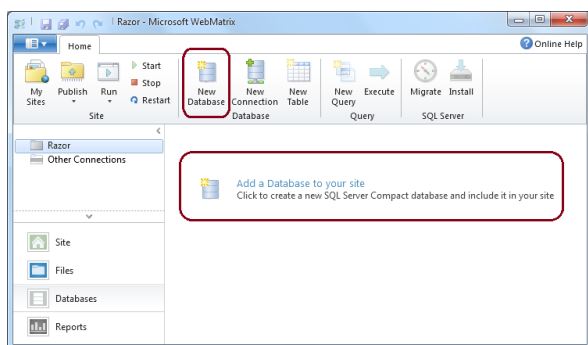
Aplikácie vytvorené v prostredí WebMatrix môžu využívať súborovú „embedded“ databázu **SQLServer Compact Edition**. V praxi to znamená, že pre jednoduchšie projekty nepotrebuje databázový server, čo znamená úsporu času a nákladov.

Táto koncepcia zároveň umožňuje jednoduchú migráciu v prípade rastu biznisu, nakoľko SQLServer Compact Edition je kompatibilný s ostatnými edíciami databázového servera SQL Server.

Vytvorenie databázy

Môžete vytvoriť novú databázu priamo v prostredí WebMatrix pomocou integrovaného nástroja pre SQLServer Compact Edition.

- V ľavom dolnom okne zvolíte pracovný priestor **Databases**.
- Na páse kariet kliknite na tlačidlo **New database**. Nová databáza bude mať rovnaký názov ako názov projektu.



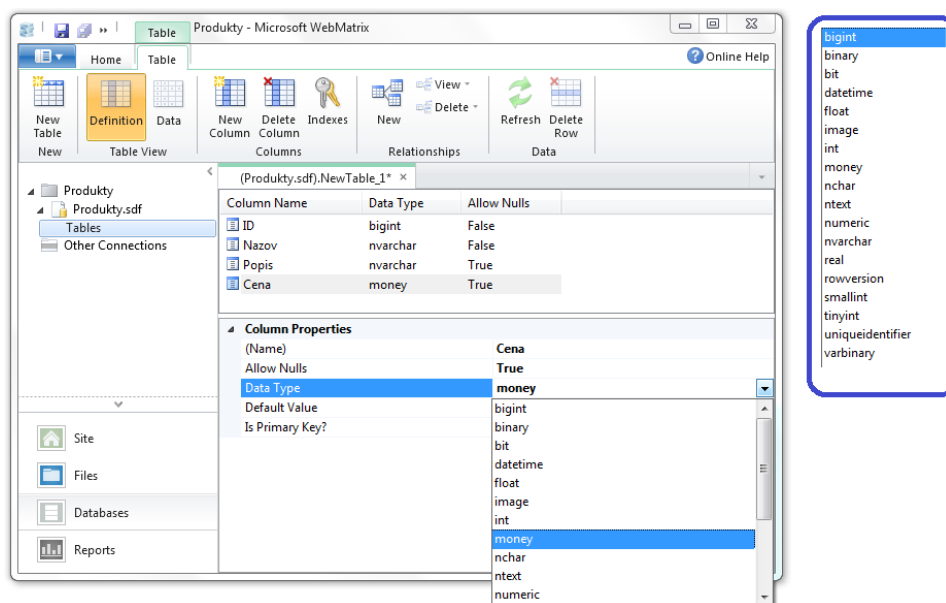
Vytvorenie novej databázy, ktorá bude súčasťou webovej aplikácie

Môžete taktiež použiť aj existujúcu súborovú SQLServer Compact Edition databázu napríklad z inej WebMatrix aplikácie.

Návrh štruktúry databázovej tabuľky

Po vytvorení databázy je potrebné pridať tabuľky s dátami. V ľavom hornom okne rozbaľte symbol databázy, v našom prípade Produkty.sdf a kliknite na tlačidlo **New Table**. WebMatrix otvorí okno pre návrh štruktúry databázy. V tomto okne zadávaním jednotlivých atribútov vytvárate štruktúru databázy.

Ku každému atribútu je potrebné priradiť aj zodpovedajúci dátový typ, podľa toho aký druh údajov bude v príslušnom stĺpci databázovej tabuľky uložený.

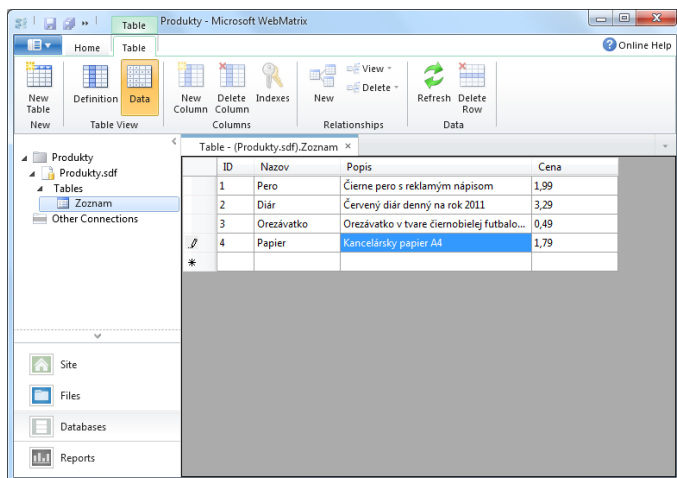


Návrh štruktúry databázovej tabuľky

Každá tabuľka by mala obsahovať primárny kľúč, podľa ktorého sa dá príslušný záznam jednoznačne identifikovať. Pri produkte to môže byť napríklad katalógové číslo.

Naplnenie databázovej tabuľky údajmi

Databázovú tabuľku môžete naplniť údajmi buď staticky, teda ručným vložením, prípadne importom, alebo dynamicky priamo z webovej aplikácie. Ručné vkladanie sa hodí pre údaje, ktoré sa príliš často nemenia, napríklad katalóg ponuky. **Tento spôsob využili pre uloženie katalógu cukroví aj tvorcovia vzorovej aplikácie zo šablóny Bakery.**



„Ručné“ vkladanie údajov do databázovej tabuľky

V našom prípade sme do databázovej tabuľky Zoznam zadali túto vzorku údajov:

ID	Názov	Popis	Cena
1	Pero	Čierne pero s reklamným nápisom	1,99
2	Diár	Červený diár denný na rok 2011	3,29
3	Strúhadlo	Strúhadlo v tvare čiernobielej futbalovej lopty	0,49
4	Papier	Kancelársky papier A4	1,79

Razor využíva pre prístup k údajom v databáze objekt Database.

K dispozícii sú dva spôsoby, ako sa pripojiť k databáze:

- **Database.Open,**
- **Database.OpenConnectionString.**

```
@{    var db = Database.Open("Produkty");
```

Príkaz **db.Query** spustí vykonávať dopyt ktorý vráti množinu údajov, v našom prípade zoznam produktov. Nakoľko je vrátených viac záznamov, na čítanie sa využíva cyklus foreach:

```
@foreach(var row in db.Query(SELECT * FROM Zoznam ORDER BY Nazov))
```

Konkrétne atribúty (stĺpce databázovej tabuľky) príslušného záznamu sa vypisujú pomocou syntaxe:

```
@row.NazovAtributu
```

Kompletný kód pre výpis zoznamu produktov vo forme HTML tabuľky:

ZoznamProduktov.cshtml

```
@{    var db = Database.Open("Produkty");
    var sSQL = "SELECT * FROM Zoznam ORDER BY Nazov"; }

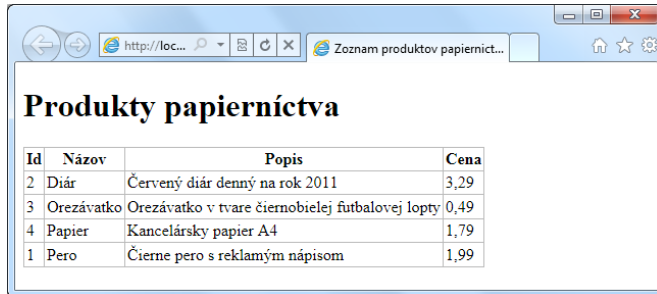
<!DOCTYPE html><html>
<head>
<title>Zoznam produktov papiernictva</title>
<style>
    table, th, td
    {
        border: solid 1px #bbbbbb;
        border-collapse: collapse;
        padding: 2px;
    }
</style>
</head>

<body>
<h1>Produkty papiernictva</h1>
<table>
<thead><tr><th>Id</th>
<th>Názov</th>
<th>Popis</th>
<th>Cena</th></tr>
</thead>
<tbody>
    @foreach(var row in db.Query(sSQL))
    {
<tr><td>@row.ID</td>
```

```

<td>@row.Nazov</td>
<td>@row.Popis</td>
<td>@row.Cena</td></tr>
}
</tbody>
</table>
</body>
</html>

```



The screenshot shows a web browser window with the address bar displaying 'http://loc...' and the page title 'Zoznam produktov papiernictva...'. The main content area has the heading 'Produkty papiernictva' and a table with the following data:

Id	Názov	Popis	Cena
2	Diár	Červený diár denný na rok 2011	3,29
3	Orezávatko	Orezávatko v tvare čiernobielej futbalovej lopty	0,49
4	Papier	Kancelársky papier A4	1,79
1	Pero	Čierne pero s reklamným nápisom	1,99

Zobrazenie obsahu databázovej tabuľky na HTML stránke

Zobrazenie údajov pomocou WebGridHelper

V predchádzajúcom (rozsahom skôr odstrašujúcom) príklade ste videli, ako sa dajú vypísať údaje pomocou „ručne“ skonštruovanej HTML tabuľky. Oveľa jednoduchšie a intuitívnejšie to dokážete pomocou objektu **WebGridHelper**. Tento objekt poskytuje navyše možnosť stránkovania, triedenia, či vlastného formátovania zobrazovaných údajov.

```

@{
    var db = Database.Open("Produkty") ;
    var sSQL = "SELECT * FROM Zoznam ORDER BY Nazov";
    var data = db.Query(sSQL);
    var grid = new WebGrid(data);
}
<!DOCTYPE html>

<html>
<head>
<title>Zoznam produktov papiernictva</title>
</head>
<body>
<h1>Produkty papiernictva</h1>
<div id="grid">
    @grid.GetHtml()
</div>
</body>
</html>

```

ID	Nazov	Popis	Cena
2	Diár	Červený diár denný na rok 2011	3,29
3	Orezávatko	Orezávatko v tvare čiernobielej futbalovej lopty	0,49
4	Papier	Kancelársky papier A4	1,79
1	Pero	Čierne pero s reklamným nápisom	1,99

Zobrazenie obsahu databázovej tabuľky pomocou objektu WebGridHelper

Produkty papierníctva

ID	Nazov	Popis	Cena
2	Diár	Červený diár denný na rok 2011	3,99
3	Orezávatko	Orezávatko v tvare čiernobielej futbalovej lopty	0,45
4	Papier	Kancelársky papier B4	1,39
1	Pero	Čierne pero s reklamným nápisom	1,33
22	Pravítko	Zelené 30 cm	0,4
23	Pravítko	Trojuholníkové	0,3
24	Uhlomer	Trojuholníkový	0,29
5	Značkovače	Sada troch značkovačov	3,79
18	Zošíť	Čistý A5	0,22
19	Zošíť	Linkovaný A5	0,29
1 2 >			

Ak výber obsahuje viac ako 10 záznamov, výpis údajov pomocou objektu WebGridHelper je stránkovaný tak, že na stránke sa zobrazí 10 záznamov. Stránky sa prepínajú v dolnej časti.

Pomocou HTML štýlov je možné meniť aj dizajnový štýl tabuľky:

```
@{
    var db = Database.Open("Produkty") ;
    var sSQL = "SELECT * FROM Zoznam ORDER BY Nazov";
    var data = db.Query(sSQL);
    var grid = new WebGrid(data);
}
<!DOCTYPE html>

<html>
<head>
<title>Zoznam produktov papiernictva</title>
<style>
        .table {border : 1px solid #809FFF;
                background-color: #E6FFFF;
                color: #6078BF;
                border-collapse: collapse;
        }

        .header{background-color: #6078BF;
                color: white;
        }

        .alternate{background-color: #EEE6FF;}
    </style>
</head>
<body>
```

```

<h1>Produkty papierníctva</h1>
<div id="grid">
    @grid.GetHtml( tableStyle : "table",
                    alternatingRowStyle : "alternate",
                    headerStyle : "header")

</div>
</body>
</html>

```

Produkty papierníctva

ID	Nazov	Popis	Cena
2	Diár	Červený diár denný na rok 2011	3,99
3	Orezávatko	Orezávatko v tvare čiernobielej futbalovej lopty	0,45
4	Papier	Kancelársky papier B4	1,39
1	Pero	Čierne pero s reklamným nápisom	1,33
22	Pravítko	Zelené 30 cm	0,4
23	Pravítko	Trojuholníkové	0,3
24	Uhlomer	Trojuholníkový	0,29
5	Značkovače	Sada troch značkovačov	3,79
18	Zošit	Čistý A5	0,22
19	Zošit	Linkovaný A5	0,29

Zmena štýlu tabuľky

V predchádzajúcom príklade, pri implicitne nastavených parametroch sa zobrazili všetky stĺpce databázovej tabuľky. Ak chcete zobraziť len niektoré, je potrebné ich pri vytváraní objektu WebGrid vymenovať.

Nahradíte v príklade riadok

```
var grid = new WebGrid(data);
```

riadkom

```
var grid = new WebGrid(data, columnNames : new [] {"Nazov", "Cena"});
```

Produkty papierníctva

<u>Nazov</u>	<u>Cena</u>
Diár	3,99
Orezávatko	0,45
Papier	1,39
Pero	1,33
Značkovače	3,79

Selekcia – do výpisu sú vybrané len niektoré stĺpce

Vkladanie údajov z aplikácie

Vkladanie sa obvykle aktivuje zo stránky na ktorej je výpis dosiaľ uložených údajov. Preto do kódu z predchádzajúcej state vložte riadok:

```
<a href="Pridaj.cshtml">Pridaj produkt do katalógu</a>
```

Čiže kompletný kód výpisu bude:

```
@{
    var db = Database.Open("Produkty") ;
    var sSQL = "SELECT * FROM Zoznam ORDER BY Nazov";
    var data = db.Query(sSQL);
    var grid = new WebGrid(data);
}

<html lang="en">
<head>
<meta charset="utf-8" />
<title></title>
</head>
<body>
<h1>Produkty papierníctva</h1>
<div id="grid"> @grid.GetHtml() </div>
<a href="Pridaj.cshtml">Pridaj produkt do katalógu</a>
</body>
</html>
```

Produkty papierníctva

<u>ID</u>	<u>Nazov</u>	<u>Popis</u>	<u>Cena</u>
2	Diár	Červený diár denný na rok 2011	3,29
3	Orezávatko	Orezávatko v tvare čiernobielej futbalovej lopty	0,49
4	Papier	Kancelársky papier A4	1,79
1	Pero	Čierne pero s reklamným nápisom	1,99
Pridaj produkt do katalógu			

Doplnenie výpisu z databázy o prepojenie na stránku pre vloženie nového záznamu

Pridaný riadok odkazuje na stránku **Pridaj.cshtml** na ktorej bude formulár na zadanie údajov, ktoré sa budú vkladať do databázy a kód pre vloženie:

```
@{
    var Nazov = ""; var Popis = ""; var Cena = "";
    if (IsPost)
    {
        Nazov = Request["formNazov"];
        Popis = Request["formPopis"];
        Cena = Request["formCena"];
        var SQLINSERT = "INSERT INTO Zoznam (Nazov, Popis, Cena) VALUES (@0, @1, @2)";
        var db = Database.Open("Produkty");
        db.Execute(SQLINSERT, Nazov, Popis, Cena);
    }
}

<html>
<head>
<meta charset="utf-8" />
<title></title>
</head>
<body>
<h1>Pridanie produktu do databázy</h1>
```

```

<form action="" method="post">
<p>Nazov:<input type="text" name="formNazov" /></p>
<p>Popis:<input type="text" name="formPopis" /></p>
<p>Cena:<input type="text" name="formCena" /></p>
<p><input type="submit" value="Pridaj produkt" /></p>
</form>
</body>
</html>

```

Pridanie produktu do databázy

Nazov:

Popis:

Cena:

Formulár na zadávanie údajov

Formulár môžete dizajnovovo vylepšiť pomocou HTML tagu<fieldset>:

```

<form action="" method="post">
<fieldset>
<legend>Údaje o produkte</legend>
<p>Nazov:<input type="text" name="formNazov" /></p>
<p>Popis:<input type="text" name="formPopis" /></p>
<p>Cena:<input type="text" name="formCena" /></p>
<p><input type="submit" value="Pridaj produkt" /></p>
</fieldset>
</form>

```

Pridanie produktu do databázy

Údaje o produkte

Nazov:

Popis:

Cena:

Formulár na zadávanie údajov graficky zvýraznený pomocou tagu<fieldset>

Úpravy údajov z aplikácie

Podobne ako vkladať, je z aplikácie možné aj editovať údaje. V tomto prípade je úloha rozdelená do dvoch krokov. V prvom formulári musí používateľ vybrať záznam, ktorý bude editovať a následne aplikácia prejde do druhého – editačného formulára.

Formulár pre výber záznamu:

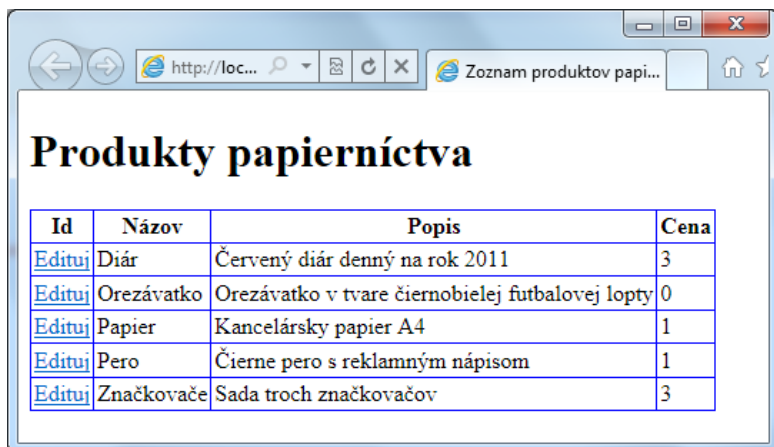
```

@{
    var db = Database.Open("Produkty");
    var sSQL = "SELECT * FROM Zoznam ORDER BY Nazov";
}

<!DOCTYPE html><html>
<head>
<title>Zoznam produktov papiernictva</title>
<style>
    table, th, td {border: solid 1px blue; border-collapse: collapse; padding: 2px; }
</style>
</head>

<body>
<h1>Produkty papiernictva</h1>
<table>
<thead><tr><th>Id</th>
<th>Názov</th>
<th>Popis</th>
<th>Cena</th></tr>
</thead>
<tbody>
        @foreach (var row in db.Query(sSQL))
        {
<tr><td><a href="@Href("~/Edituj", row.Id)">Edituj</a></td>
<td>@row.Nazov</td>
<td>@row.Popis</td>
<td>@row.Cena</td></tr>
        }
</tbody>
</table>
</body>
</html>

```



Id	Názov	Popis	Cena
Edituj	Diár	Červený diár denný na rok 2011	3
Edituj	Orezávatko	Orezávatko v tvare čiernobielej futbalovej lopty	0
Edituj	Papier	Kancelársky papier A4	1
Edituj	Pero	Čierne pero s reklamným nápisom	1
Edituj	Značkovače	Sada troch značkovačov	3

Formulár pre výber záznamu

Formulár pre zmenu údajov:

```

@{
    var Nazov = ""; var Popis = ""; var Cena = "";
    var id = UrlData[0];
    var SQLSELECT = "SELECT * FROM Zoznam where ID=@0";
    var db = Database.Open("Produkty");
    var Produkt = db.QuerySingle(SQLSELECT, id);
    var ProduktNazov = Produkt.Nazov;
    var ProduktPopis = Produkt.Popis;
    var ProduktCena = Produkt.Cena;
    if (IsPost)
    {
        Nazov = Request["formNazov"];
    }
}

```

```

        Popis = Request["formPopis"];
        Cena = Request["formCena"];
        var SQLUPDATE = "UPDATE Zoznam Set Nazov=@0, Popis=@1, cena=@2 WHERE id=@3";
        db.Execute(SQLUPDATE, Nazov, Popis, Cena, id);
    }
}

<html>
<head>
<meta charset="utf-8" />
<title></title>
</head>
<body>
<h1>Zmena údajov o produkte</h1>
<form action="" method="post">
<p>Nazov:<input type="text" name="formNazov" value="@ProduktNazov"/></p>
<p>Popis:<input type="text" name="formPopis" value="@ProduktPopis"/></p>
<p>Cena:<input type="text" name="formCena" value="@ProduktCena"/></p>
<p><input type="submit" value="Uprav" /></p>
</form>
</body>
</html>

```

Zmena údajov o produkte

Nazov:

Popis:

Cena:

Formulár pre editovanie údajov

Vymazávanie údajov z aplikácie

Vymazávanie je jednoduchšie, než vkladanie, takže stačí zjednodušiť predchádzajúci príklad. Stránka pre výber záznamu, ktorý je potrebné vymazať zostáva prakticky bezo zmeny, akurát sa zmení popis prepojenia z „Edituj“ na „Vymaz“:

```

@{
    var db = Database.Open("Produkty");
    var sSQL = "SELECT * FROM Zoznam ORDER BY Nazov";
}

<!DOCTYPE html><html>
<head>
<title>Zoznam produktov papiernictva</title>
<style>
    table, th, td {border: solid 1px blue; border-collapse: collapse; padding: 2px; }
</style>
</head>

<body>
<h1>Produkty papiernictva</h1>
<table>
<thead><tr><th>Id</th>
<th>Nazov</th>

```

```

<th>Popis</th>
<th>Cena</th></tr>
</thead>
<tbody>
    @foreach(var row in db.Query(sSQL))
    {
<tr><td><a href="@Href("~/Vymaz", row.Id)">Vymaz</a></td>
<td>@row.Nazov</td>
<td>@row.Popis</td>
<td>@row.Cena</td></tr>
    }
</tbody>
</table>
</body>
</html>

```

Stránka, pomocou ktorej sa fyzicky vykoná vymazanie nebude obsahovať zadávací formulár, ale bude vyžadovať potvrdenie vymazania:

```

@{
    var db = Database.Open("Produkty");
    var id = UrlData[0];
    if (id.IsEmpty()) {
        Response.Redirect(@Href("/Vymazavanie"));
    }
    var prod = db.QuerySingle("SELECT * FROM Zoznam WHERE ID = @0", id);
    if (IsPost && !id.IsEmpty())
    {
        var deleteQueryString = "DELETE FROM Zoznam WHERE ID=@0";
        db.Execute(deleteQueryString, id);
        Response.Redirect("/Vymazavanie");
    }
}

<html>
<title>Vymazanie produktu</title>
</head>
<body>
<h1>Potvrdenie vymazania produktu</h1>
<form method="post" action="" name="form">
<p>Naozaj chcete vymazať tento produkt?</p>

<p>Nazov: @prod.Nazov <br />
    Popis: @prod.Popis <br />
    cena: @prod.Cena</p>
<p><input type="submit" value="Vymaz" /></p>
</form>
</body>
</html>

```

Potvrdenie vymazania produktu

Naozaj chcete vymazať tento produkt?

Nazov: Zošit
 Popis: Čistý A4
 cena: 0,79

Vymaz

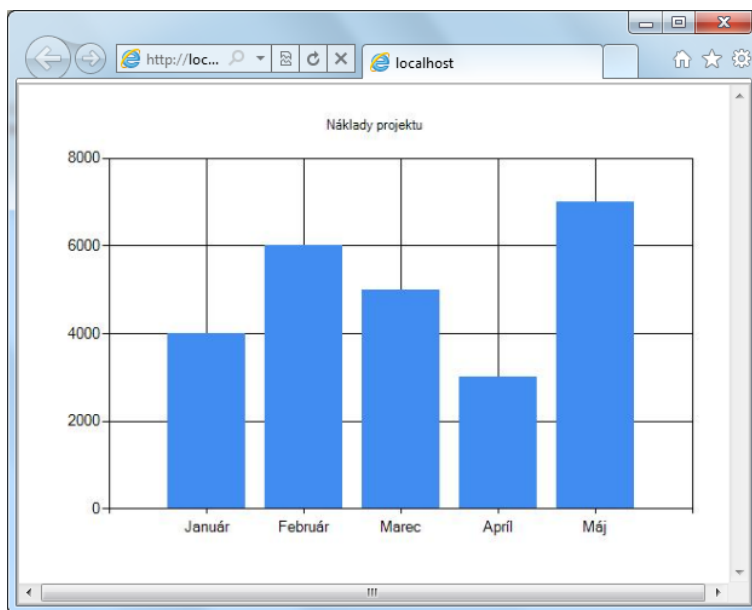
Stránka pre potvrdenie vymazania

Zobrazenie údajov vo forme grafov

Hlavne v biznis aplikáciách, ale aj v rôznych štatistikách sa čoraz viac uplatňuje takzvaná biznis grafika, to znamená zobrazenie údajov v grafickej forme.

Graf vytvorený z explicitne zadáných údajov:

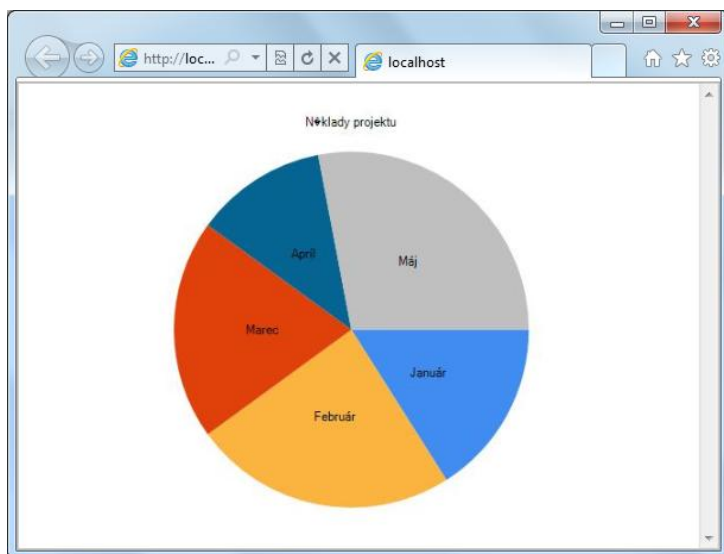
```
@{
    var key = new Chart(width: 600, height: 400)
        .AddTitle("Náklady projektu")
        .AddSeries(
            name: "Mesiac",
            xValue: new[] { "Január", "Február", "Marec", "Apríl", "Máj" },
            yValues: new[] { "4000", "6000", "5000", "3000", "7000" })
        .Write();
}
```



Zobrazenie údajov vo forme grafov

Koláčový graf:

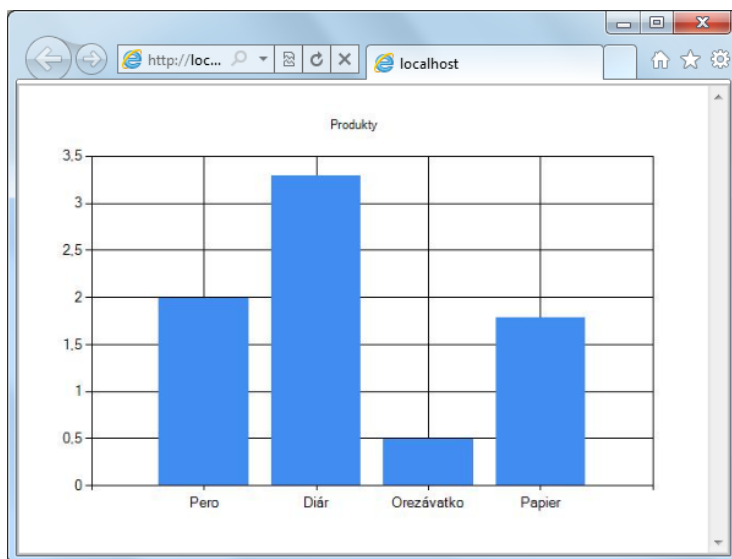
```
@{
    var key = new Chart(width: 600, height: 400)
        .AddTitle("Náklady projektu")
        .AddSeries("Default", chartType: "Pie",
            xValue: new[] { "Január", "Február", "Marec", "Apríl", "Máj" },
            yValues: new[] { "4000", "6000", "5000", "3000", "7000" })
        .Write();
}
```



Zobrazenie údajov vo forme koláčového grafu

Oveľa častejšie sa vo forme grafu zobrazujú údaje z databázových tabuliek:

```
@{
    var db = Database.Open("Grafy");
    var data = db.Query("SELECT Nazov, Cena FROM Zoznam");
    var key = new Chart(width: 600, height: 400)
        .AddTitle("Produkty")
        .DataBindTable(dataSource: data, xField: "Nazov")
        .Write();
}
```

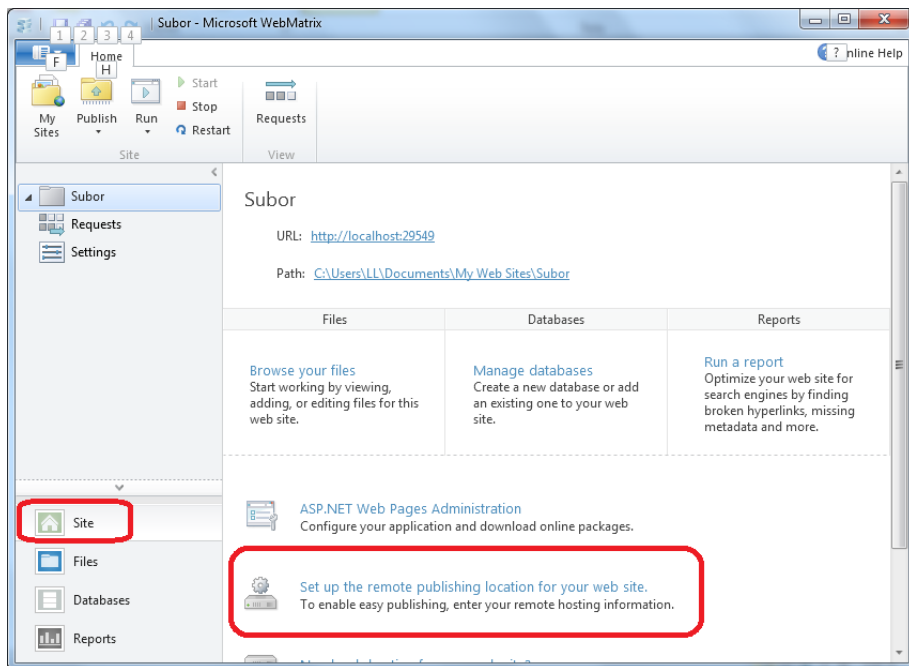


Zobrazenie údajov z databázy

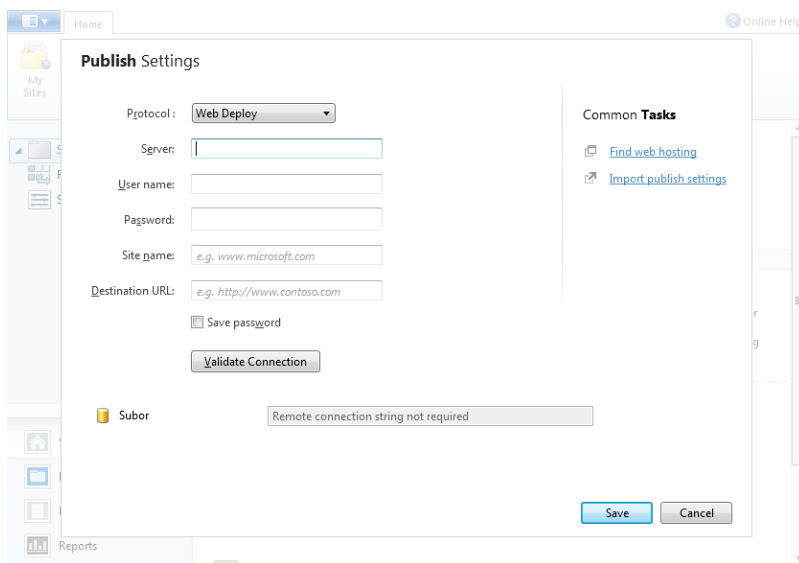
Publikovanie webovej aplikácie na server

Publikovanie webovej aplikácie na server je finálnym, no určite nie záverečným krokom pri tvorbe aplikácie. Po odladení na lokálnom počítači „idete s kožou na trh“ a sprístupníte svoju aplikáciu na webe pre všetkých.

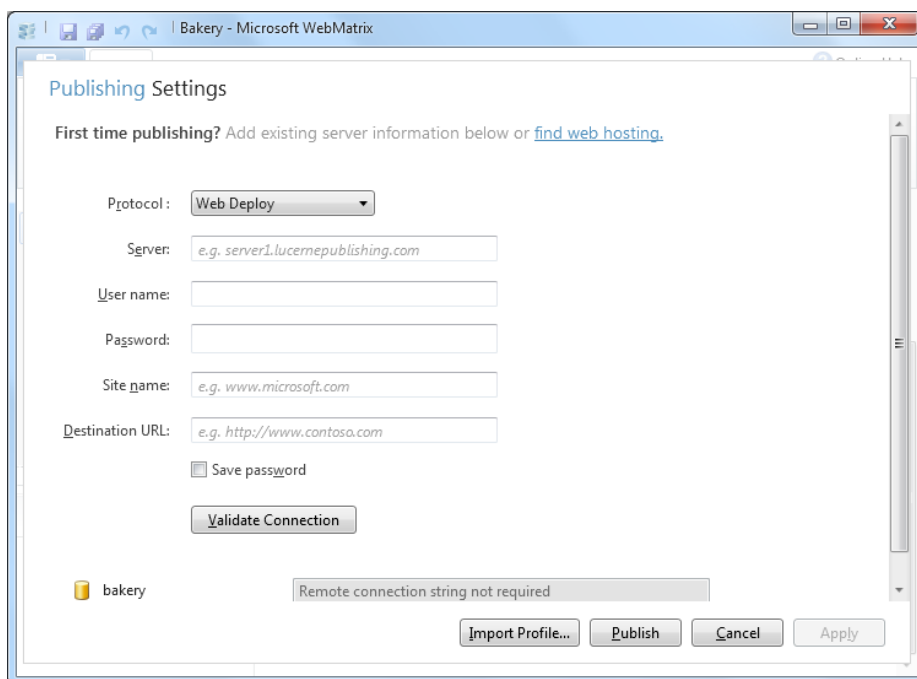
Prvým krokom je výber poskytovateľa webhostingu, kde bude aplikácia umiestnená.



Publikovanie sa zaháji v pracovnom priestore Site



Pre publikovanie je potrebné nastaviť parametre, ktoré získate od správcu webhostingu



Dialóg pre publikovanie webovej aplikácie

Postúpte na vyššiu métu – Visual Web Developer 2010 Express.

Vývojové prostredie Visual Web Developer 2010 Express poskytuje vynikajúce možnosti pre vývoj rôznych typov webových aplikácií. Je kompatibilné s komerčnou verziou nástroja VisualStudio, takže znalosti získané pri vývoji personálnych stránok a hobby aplikácií môžete prípadne neskôr zúročiť v profesionálnejších aplikáciách.

Vstupný bod pre prevzatie vývojových nástrojov pre webové aplikácie je na adrese <http://www.microsoft.com/web/default.aspx>.

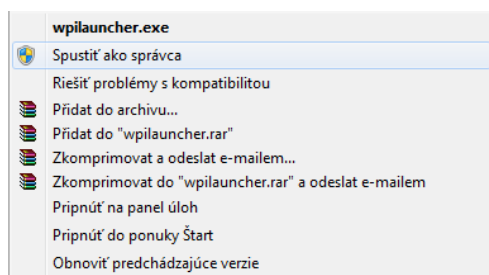
Prípadne si môžete vývojárske nástroje prevziať priamo na adrese <http://www.microsoft.com/web/downloads/platform.aspx>.

Na PC musíte mať niektorý z podporovaných operačných systémov: Windows 7, Windows Vista, Windows Vista SP1, Windows XP SP2+, Windows Server 2003 SP1+, Windows Server 2008, Windows Server 2008 R2.

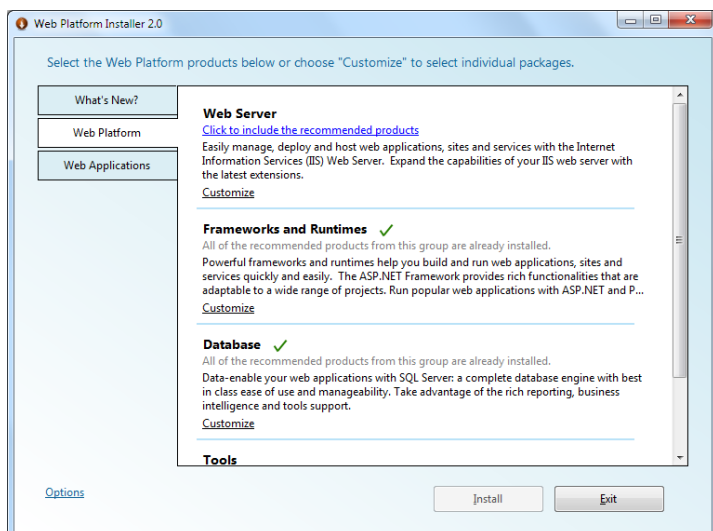
V rámci balíka **Microsoft Web Platform** sa nainštalujú produkty:

- Visual Web Developer 2010 Express,
- Internet InformationServices (IIS),
- SQL Server Express,
- .NET Framework.







Pre spustenie inštalácie sú potrebné administrátorské privilégia, preto odporúčame prevziať inštalačný súbor wpilauncher.exe (ktorý má len niekoľko megabajtov a iniciuje inštaláciu z webu) na lokálny disk a tam ho spustiť v režime „spustiť ako správca“.



Inštaláciu je potrebné spustiť v režime „spustiť ako správca“



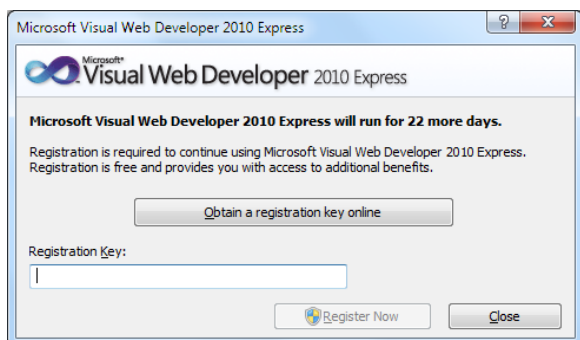
Web PlatformInstaller 2.0 – úvodná obrazovka

 <p>Web Server Install the latest version of IIS, including latest IIS Web extensions like IIS Media Services.</p>	 <p>Framework Install the latest version of the .NET Framework. This includes everything you need to work with ASP.NET.</p>
 <p>Database Install the latest version of SQL Server 2008 Express. This includes both the database engine and tools.</p>	 <p>Tools Install the latest version of Visual Web Developer Express, Our full featured free web development tool.</p>
 <p>Web Apps Install free popular ASP.NET and PHP web apps such as Umbraco and WordPress. Browse the Windows Web Application Gallery</p>	 <p>Extras Web PI also includes the latest community version of PHP for Windows and a trial version of Expression Web 3!</p>

Web PlatformInstaller 2.0 – komponenty ktoré sa nainštalujú ako jeden balík

Vývoj webovej aplikácie v prostredí nástroja Visual Web Developer 2010 Express

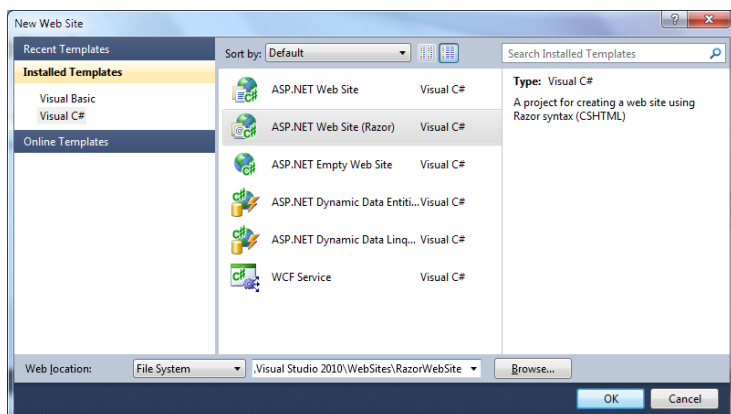
Produkt je potrebné po inštalácii aktivovať, inak bude funkčný len obmedzený počet dní. Na nutnosť aktivácie budete po prvý krát upozornení na konci inštalácie a potom periodicky, keď sa doba do aktivácie bude chýliť ku koncu.



Visual Web Developer 2010 Express je potrebné zaregistrovať, inak do 30 dní expiruje. Kód získate na webe po prihlásení pomocou LiveID konta.

Projekt ASP.NET aplikácie

Nakoľko publikácia je koncipovaná prakticky, aj s vývojovým prostredím Visual Web Developer 2010 Express sa zoznámite „za pochodu“ pri vytváraní prvej webovej aplikácie. V ponuke pre vytvorenie novej webovej aplikácie je niekoľko šablón viacerých typov aplikácií, vzhľadom na zameranie publikácie využijeme z ponúkaných možností len vytvorenie projektu **ASP.NET Web Site (Razor)**.



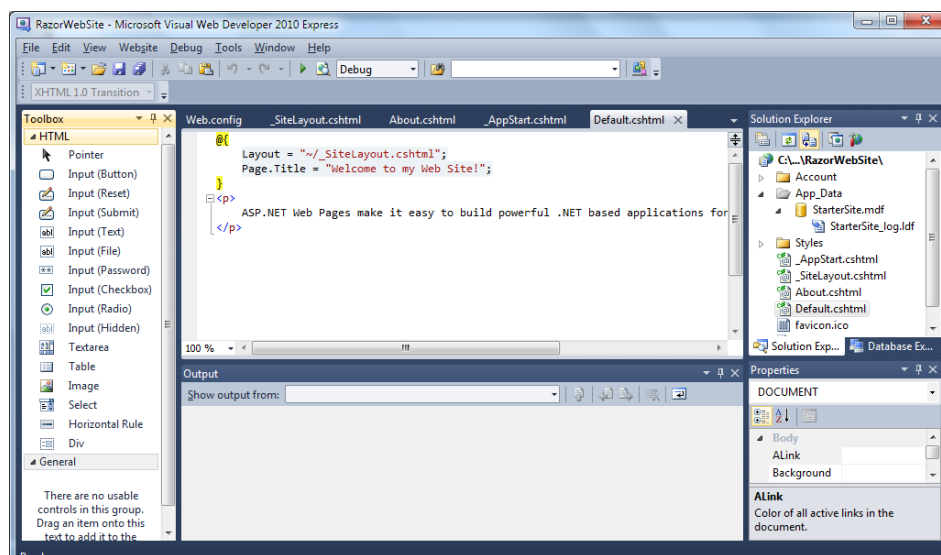
Vytvorenie projektu ASP.NET Web Site (Razor)

Pracovná obrazovka nástroja Visual Web Developer v implicitnej konfigurácii obsahuje stredné pracovné okno, na okrajoch ktorého sú pomocné okná:

- Toolbox,
- Solution Explorer,
- Properties,
- Output.

V ľavej časti pracovnej obrazovky sa nachádza okno Toolbox. Vpravo sú okná Solution Explorer, Database Explorer (prepínajú sa pomocou záložiek v jednom okne) a Properties. Väčšinu pomocných okien (Toolbox, Solution Explorer, Properties, Output...) je možné podľa potreby, alebo podľa rozlíšenia obrazovky vývojárskeho počítača buď operatívne schovávať, alebo pevne „přišpendliť“. Každý z týchto

panelov sa obvykle skladá z niektorých záložiek, ktoré menia jeho aktuálny význam, čím je veľmi efektívne využitá zobrazovacia plocha monitora.

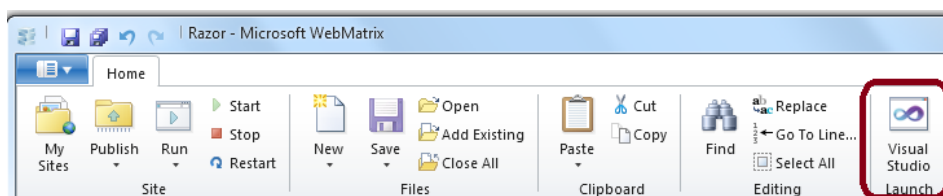


Vývojové prostredie nástroja Visual Web Developer 2010

Postúpte na ešte vyššiu métu – VisualStudio 2010

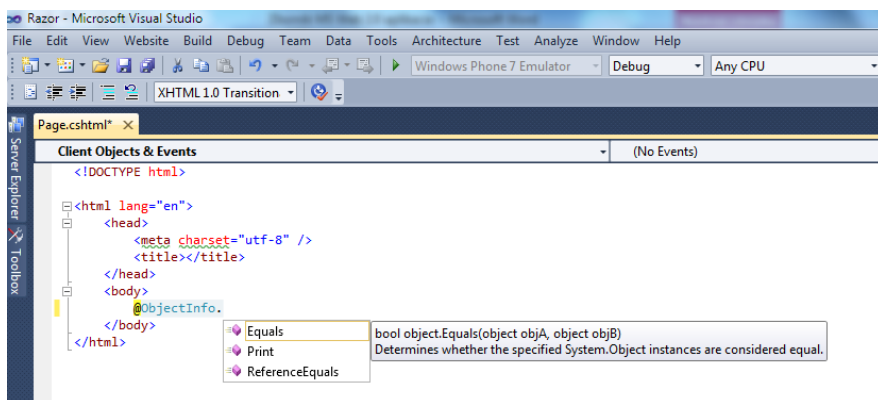
Hoci primárne je táto publikácia určená vývojárom „hobby“ aplikácií, niekedy sa stane, že vaše hobby alebo webová aplikácia prerastie do seriózneho biznisu a vtedy je potrebné webovú aplikáciu škálovať, prípadne migrovať do výkonnejšieho serverového prostredia. Pre vývoj rozsiahlejších, zložitejších a robustnejších webových aplikácií s využitím moderných technológií AJAX a Silverlight je určené vývojové prostredie Microsoft VisualStudio, pričom jeho najnovšia verzia má označenie 2010.

Projekty z nástroja WebMatrix je možné jednoducho migrovať do prostredia VisualStudio a tak sa v prípade osvedčeného webového projektu posunúť na vyššiu profesionálnejšiu métu.



WebMatrix prepojený s vývojovým prostredím VisualStudio, pomocou ikonu s logom VisualStudio môžete „matrixový“ projekt otvoriť v tomto vývojovom prostredí.

Pri písaní a editovaní Razor kódu v textovom režime vám vývojové prostredie poskytne cenné služby v podobe interaktívneho pomocníka IntelliSense. Tento pomocník funguje nielen pre kľúčové slová a objekty programovacích jazykov, ale aj pre HTML dokumenty, ASP.NET tagy a podobne. Napríklad po napísaní názvu objektu `ObjectInfo` vám IntelliSense ponúkne zoznam atribútov a funkcií dostupných pre tento objekt.



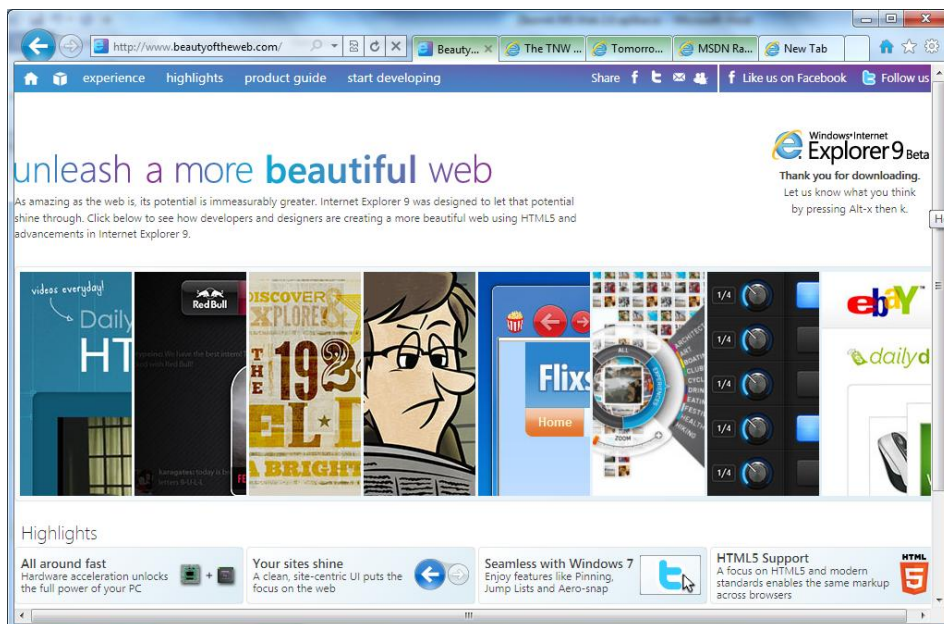
Vo prostredí VisualStudio môžete pri písaní Razor Syntaxe využiť IntelliSense

Internet Explorer 9

Dosiaľ sme sa venovali daniu „za oponou“, teda vývoju ASP.NET webových aplikácií. Alfou a omegou snaženia vývojárov webových aplikácií je komfort a spokojnosť používateľov. A k tomu rozhodujúcou mierou prispieva aj aplikačné prostredie, v ktorom sa webová aplikácia sprostredkováva používateľovi, čiže prehľadávač webového obsahu. Aj napriek možnostiam Webu 2.0 využívajúceho pokročilé technológie AJAX, Silverlight, či Flash, nie je práca cez webové rozhranie v súčasných prehľadávačoch taká komfortná ako cez natívne aplikačné rozhranie. Preto požiadavky používateľov smerujú hlavne k zvýšeniu používateľského komfortu v synergii s Webom 2.0. Reakciou na tieto požiadavky je nová verzia prehľadávača Internet Explorer 9.

Inštalácia

Prvý marketingový dojem spojený s očakávaním sa u používateľa asociuje s názvom stránky produktu www.beautyoftheweb.com, odkiaľ je možné betaverziu prehľadávača Internet Explorer 9 (IE9) prevziať. K dispozícii je pomerne veľa jazykových verzií, na rozdiel od češtiny však slovenčina zatiaľ medzi ne nepatrí. Podporované sú len novšie operačné systémy Vista SP2, Windows Server 2008 SP2 a Windows 7 takže zástancovia „starých dobrých“ Windows XP budú možno trochu sklamaní. Inštalácia si nevyžaduje prakticky žiadnu asistenciu používateľa.



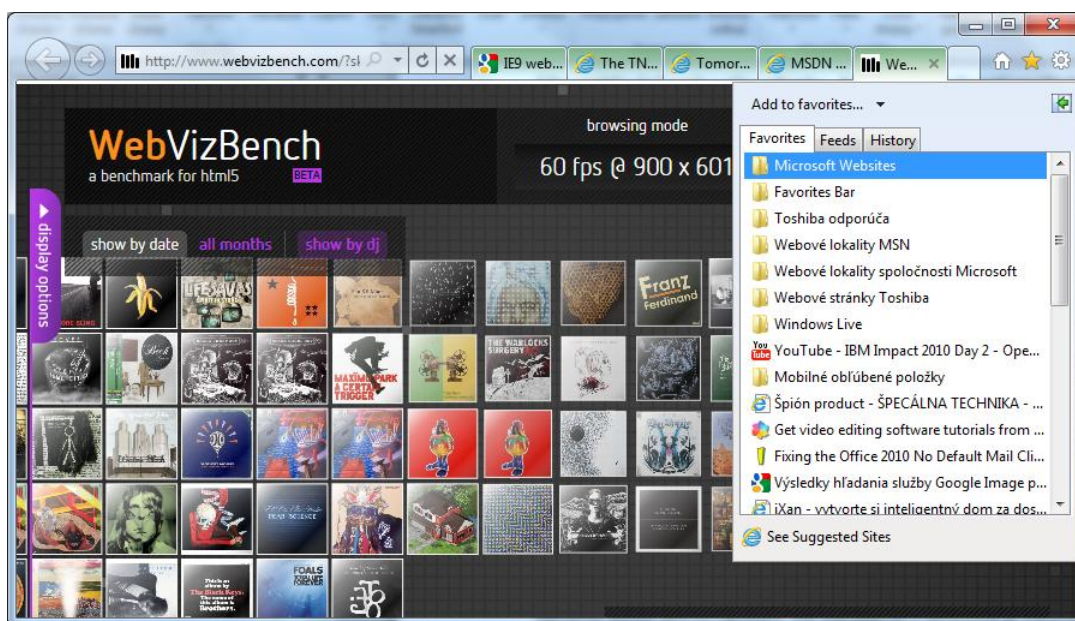
Internet Explorer 9 je možné prevziať na www.beautyoftheweb.com

Maximum priestoru na prezentovaný obsah

Aby na zobrazenie obsahu webovej stránky zostal maximálny možný priestor, muselo čo najviac prvkov používateľského rozhrania nového prehľadávača ustúpiť do pozadia. Je to dôležité nielen kvôli mininotebookom; najrozšírenejšie na multimediálnych notebookoch nižšej a strednej triedy je rozlíšenie 1366 x 768. Aktuálne sa zobrazujú len tie prvky, ktoré sú v danej chvíli potrebné. Dominujú im okno pre zadanie adresy URL a tlačidlá pre navigovanie dopredu a dozadu. Ak na ne kliknete pravým

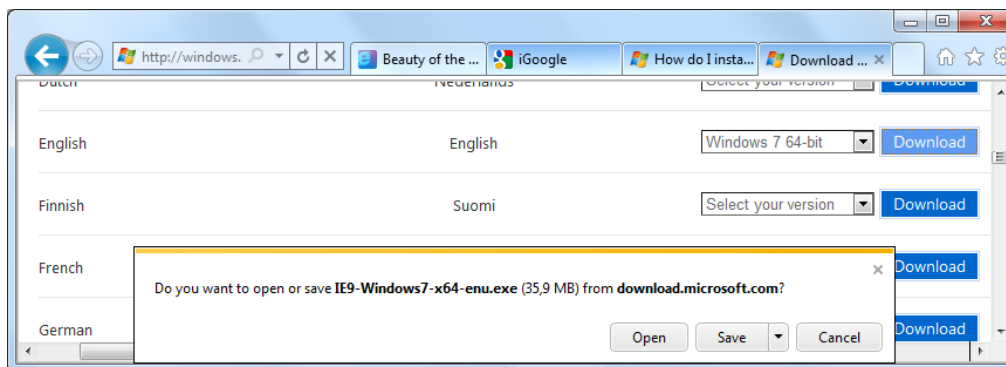
tlačidlom myši, zobrazí sa zoznam naposledy navštívených stránok. Tlačidlo pre návrat na predchádzajúcu stránku je väčšie, nakoľko sa používa oveľa častejšie. Vlastne po novom sa toto univerzálne okno volá „One Box“. Služi totiž aj na zadávanie vyhľadávaných fráz a grafickú, či farebnú signalizáciu špeciálnych režimov a navzájom súvisiacich stránok. Pri vyhľadávaní sa počas písania frázy interaktívne zobrazuje ponuka najnavštevovanejších stránok.

Pomocou tlačidla Alt sa dá zapnúť aj zobrazenie klasickej ponuky.



Okno obsahujúce záložky pre zobrazenie obľúbených stránok, feedov a histórie prehľadávania sa zobrazuje v pravej časti obrazovky, pričom hlavne história sa dá prehľadne usporiadať či už podľa dátumu alebo počtu návštev. Do obľúbených položiek je možné naraz pridať stránky na všetkých momentálne otvorených kartách.

Interaktívny dialóg s používateľom sa v prípade potreby zobrazí dolnej časti obrazovky samostatný panel. Oveľa menej prekrýva obsah webovej stránky, než klasické dialógové okno.

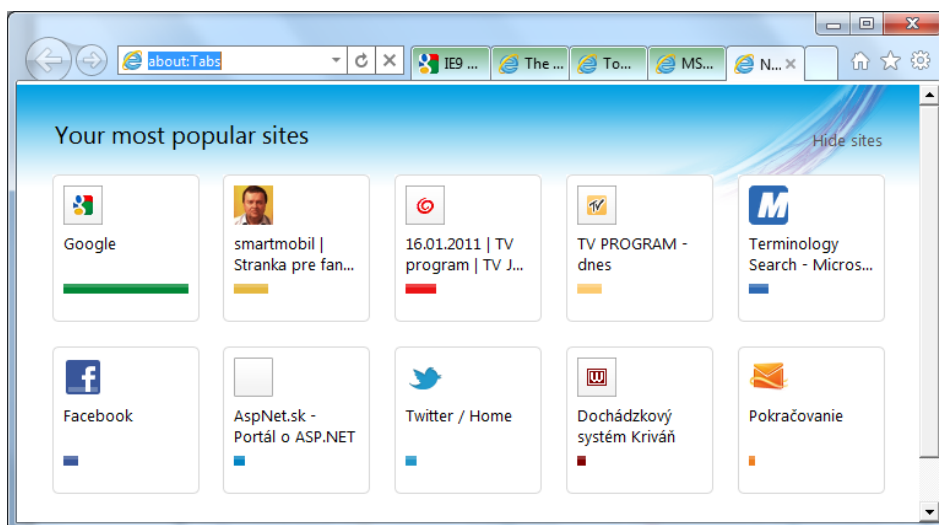


Notifikačný panel s dialógom pre potvrdzovanie akcií vyžadujúcich súhlas, alebo rozhodnutie používateľa

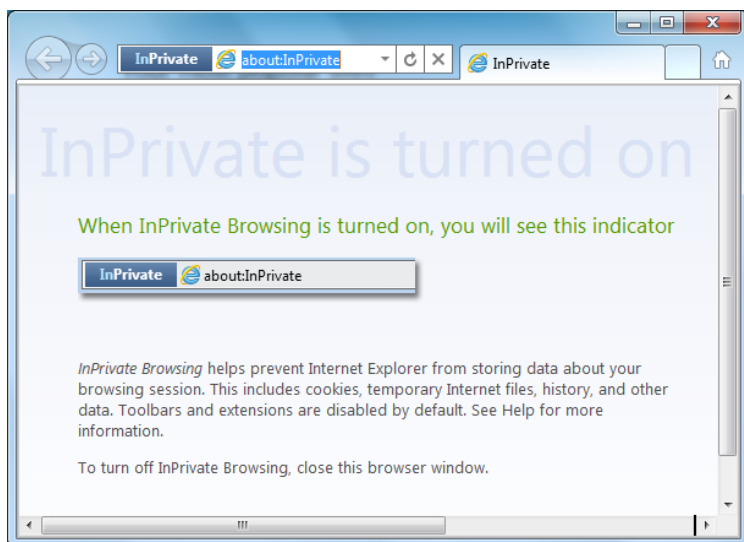
Vráťme sa však k najdôležitejšej časti prezentačnej vrstvy prehľadávača, k oknu pre zobrazenie webovej aplikácie. O engine prezentačného rozhrania, ktorý mal v procese vývoja kódové označenie Chakra sa diskutovalo na rôznych fórach ešte pred prvým predstavením nového prehľadávača a teraz sa ukázalo, že občas publikované informácie o jeho výkone vôbec neboli prehnané. IE 9 naplno využíva nielen výkon viacjadrových procesorov, ale pri zobrazovaní aj hardvérovú akceleráciu na grafickej karte. K vysokej rýchlosti prehľadávača významne prispieva optimalizovaný document-object model (DOM). Internet Explorer 9 je podľa druhu aplikácie viac 8 až 10x rýchlejší ako IE 8.

Rýchlosť prehľadávača si môžete otestovať napríklad na <http://www.webvizbench.com/?skipintro>. Odporúčame spustiť test na starom prehľadávači a následne po nainštalovaní IE9.

Na prvý pohľad v IE9 objavíte prehľadný zoznam vašich najnavštevovanejších stránok stvárnený vo forme veľkých ikon. V dolnej časti každej ikony je stĺpcový graf, ktorý ukazuje, ako často danú stránku navštevujete. Všimnite si prepojenie „InPrivateBrowsing“ na novej karte vpravo dole. Ak aktivujete tento režim, nebude sa zaznamenávať história prezeraných stránok. Tento režim je indikovaný v okne adresy URL. Podobne ako v predchádzajúcej verzii, majú vzájomne súvisiace karty zhodnú farbu záhlavia.



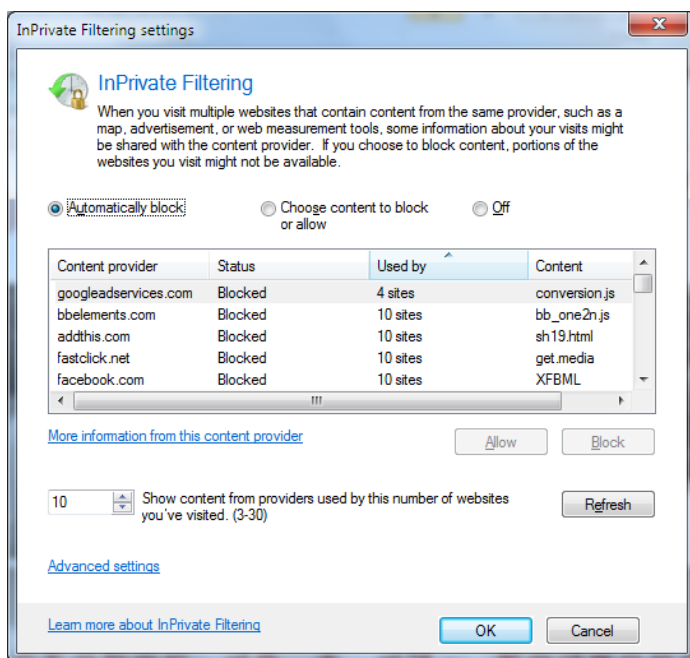
Komfort otvorenia novej záložky zvyšuje prehľadný zoznam ikon vašich najnavštevovanejších stránok



Možnosť práce v privátnom režime

Bezpečnosť

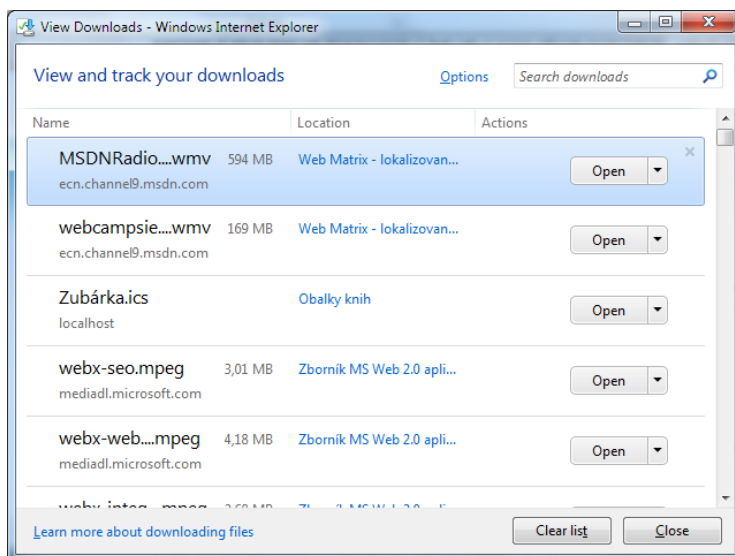
Vývojári sa určite poučili zo skúseností s prehľadávačom Internet Explorer 8, preto sú jeho bezpečnostné nástroje doplnené napríklad o integrovaný Downloadmanager, ktorý má zabudovaný vlastný SmartScreen Filter pre bezpečné preberanie. V prípade straty spojenia alebo problémov na stránke sa automaticky pokračuje v preberaní bez straty údajov.



Okno pre nastavenie možností funkcie InPrivateFiltering

Posilnená spolupráca so systémom Windows 7

Predpokladá sa, že v dobe nástupu finálnej verzie IE9 bude vrcholiť migrácia operačných systémov firemných PC na systém Windows 7. IE 9 naplno využíva možnosti systému Windows 7, napríklad umožňuje premiestniť kartu z prehľadávača do samostatného okna, prípadne ju pomocou funkcie PinnedSites pripnúť na panel úloh, takže jednotlivé webové aplikácie sa dajú otvárať priamo z hlavného panelu systému Windows 7. Tieto funkcie sú možné iba vďaka zmenám, ktoré boli urobené v jadre prehľadávača. Webové aplikácie a s nimi súvisiace procesy na jednotlivých kartách totiž bežia izolovane, takže prípadné problémy niektorej z nich už neohrozia beh ostatných aplikácií.



DownloadManager

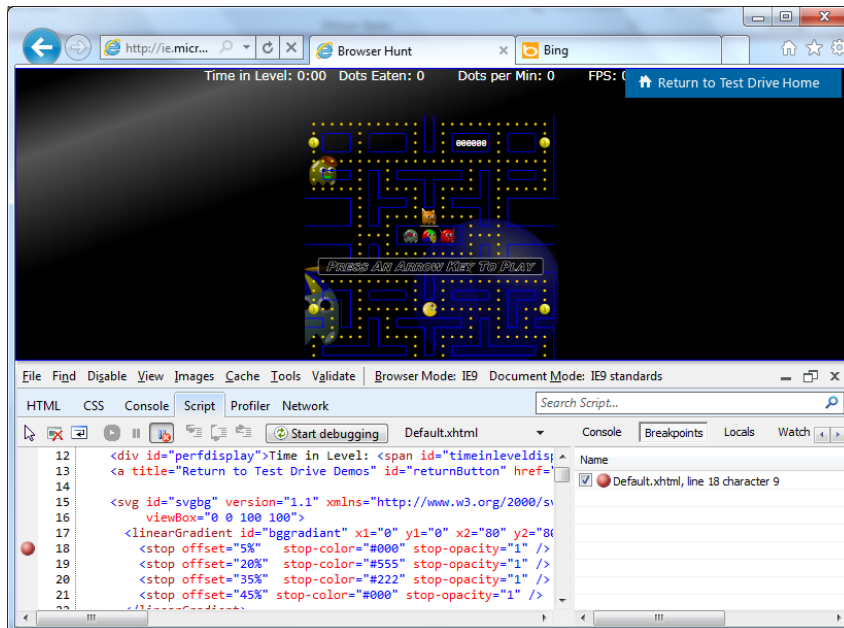
HTML5 verus Silverlight

Už pri prvých informáciách o možnostiach pripravovaného štandardu HTML 5 sa diskutovalo o budúcnosti technológií Flash a Silverlight. Nad budúcnosťou technológie Flash visia otázky hlavne preto lebo jeho podpora nebola implementovaná do čoraz populárnejšieho tabletu Apple iPad a špekuluje sa, či ho dokáže HTML 5 plne nahradiť. V prípade webovej prezentačnej technológie nebude príchod HTML5 znamenať až takú zmenu, aj keď jeho multimediálne funkcie budú so technológiou Silverlight súperiť o priazeň webových vývojárov a dizajnérov. Predpokladá sa, že Silverlight sa bude aj naďalej využívať pre komplexnejšie úlohy, ale aj na streamovanie HD videa a namierené má aj na iné platformy, napríklad Windows Phone 7. Vykresľovanie vektorového grafického obsahu, ktorý súvisí s HTML5 si vyžaduje vysoký výkon. Preto v súvislosti s podporou grafických akceleratorov označuje spoločnosť Microsoft implementáciu nového HTML štandardu na platforme Internet Explorer 9 aj ako „GPU-Powered HTML5“.

HTML ocenia hlavne vývojári

Azda najväčším magnetom priťahujúcim pozornosť vývojárov k novému prehľadávaču je HTML5. Zároveň je tento nový štandard pre nich aj veľkou výzvou, no okrem neho sa môžu realizovať aj prostredníctvom podpory, XHTML, DOM, CSS3, či vektorovej grafiky ScalableVectorGraphics (SVG).

Kláves F12 sprístupňuje vývojársky režim, v ktorom je možné pracovať so zdrojovými kódmi stránky a jej skriptov vrátane trasovania a ladenia.



Pre vývojárov je k dispozícii špeciálny režim v ktorom je možné ladiť a trasovať skripty na stránkach

Podľa spoločnosti Microsoft je Internet Explorer 9 viac než 10x rýchlejší ako IE 8 a tiež rýchlejší ako najnovšie verzie prehľadávačov Firefox a Safari. Inak povedané, staršie verzie vám sprostredkovali 10 percent možností webu a IE9 „odomyká“ zvyšných 90 percent. Najlepšie sa o tom presvedčíte sami na testovacej stránke <http://ie.microsoft.com/testdrive>. Zároveň môžete posúdiť vymoženosti štandardu HTML5.

Polemika miesto záveru

Hneď na úvod sa pokúsime zodpovedať na dve otázky, ktoré vás možno napadnú:

Web 2.0 je o interaktivite, vizualizácii, sociálnych sieťach, multimédiách... ako do toho zapadá vývojové prostredie pre hobby komunitu, ktoré neumožňuje ani poriadny vizuálny návrh?

Odpovieme protiotázkou, najpopulárnejšia platforma pre hobby komunitu je platforma PHP, ktorá je ešte konzervatívnejšia. Tak prečo je potom najpopulárnejšia?

Prvá verzia nástroja WebMatrix pred siedmimi rokmi prišla a postupne upadla do zabudnutia. Čo ak sa história zopakuje a nový WebMatrix postihne rovnaký osud?

Môže nastať aj taká situácia, no ak sa to stane, bude to len preto, že bude k dispozícii lepší nástroj. Možno to bude Express verzia nástroja VisualStudioLightSwitch, možno niečo ešte jednoduchšie a lepšie. Pri prvom WebMatrixe sa nadobudnuté vedomosti dali v plnom rozsahu zúročiť v nástroji VisualStudio, ani v tomto prípade to nebude inak. Skúsenosti s vývojom moderných interaktívnych a sociálnych aplikácií pre Web 2.0 budú v každom prípade na nezaplatenie.

