

## Tvorba webových stránok pomocou .NET frameworku

Madaras Samuel · Informačné technológie, Študentské práce

04.02.2011



.NET je vrstva medzi operačným systémom a programovacím jazykom, ktorá podporuje niekoľko programovacích jazykov. V praxi sa používajú najviac C# a Visual Basic. V tomto referáte sa zameriam na jazyk C#. Výhodami .NET sú automatická správa pamäte, procesorovo nezávislý Common Intermediate Language (jazyk, ktorý je výstupom kompilátora .NET) a niekoľko tisíc definícií tried prístupných pre všetky programovacie jazyky.

Tieto triedy uľahčujú prácu s vláknami, databázami, XML dokumentmi atď., ale najmä samotnú tvorbu aplikácií. Programový kód je kompilovaný do CIL (Common Intermediate Language), vďaka čomu je beh aplikácií rýchlejší. Kompilácia je ľahko spustiteľná priamo vo vývojovom prostredí.

### ASP.NET

ASP (Active Server Pages) je súčasťou .NET frameworku. Slúži na tvorbu dynamických webových aplikácií a webservisov. Stránky sú generované server-side kódom. Podporuje objektové programovanie a oddelenie programového kódu od web stránky. Zdrojový kód stránky nie je na jednom mieste ale je rozdelený do viacerých súborov podľa obsahu. Rozloženie elementov a html značky sú v súbore stranka.aspx, programový kód pre obsluhu eventov je v stranka.aspx.cs a tieto súbory sú spolu prepojené. Je možné vkladať kód aj do súboru stranka.aspx medzi značky `<%# %>`.

### Požiadavky

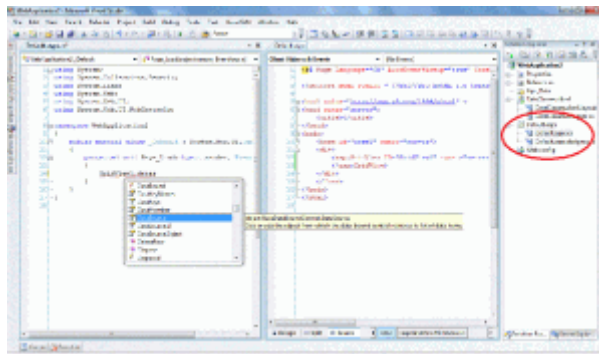
Stránky bežia na serveri IIS od Microsoftu (je navrhnutý špeciálne pre .NET aplikácie), prípadne na serveri Apache. .NET framework však vyžaduje na počítači nainštalovaný operačný systém Windows. Existuje aj multiplatformová, open source-ová implementácia .NET frameworku pre Unix, Windows, MacOS a ďalšie operačné systémy s názvom Mono.

### Vývojové prostredie

Najlepšie vývojové prostredie pre ASP.NET je Microsoft Visual Studio. Umožňuje jednoducho kompilovať alebo publikovať stránky na server jedným klikom, mapovanie databázových tabuliek na triedy drag-and-drop prenášaním a omnoho viac. Debugger ukazuje hodnotu premenných pri prejení myšou na premennú a ak je to objekt, je jednoducho zistiť hodnoty atribútov. Rozsiahly help je precízne napísaný pre každú

vstavanú triedu, obsahuje množstvo príkladov a je ľahko čitateľný.

Asi najzaujímavejšou funkcionalitou je pomocník (CTRL+SPACE) , ktorý funguje ako autocomplete a zároveň ku každej možnosti ukazuje základný popis (popis správania, argumentov, návratovej hodnoty, aké výnimky môže vytvoriť). Pre nekomerčné účely je možné zadarmo používať Visual Studio Express Edition. Táto verzia obsahuje všetku najbežnejšie používanú funkcionalitu. Na obrázku je ukážka oddelenia funkčného kódu od web stránky a autocomplete pomocník.



### Ako funguje komunikácia

Na klientskej strane je stav všetkých prvkov uložený v hidden field, ktorý sa nazýva view state. Pri spracovaní požiadavky sa tieto hodnoty dosadia do premenných. Pri každej požiadavke na server sa odosielajú:

- View state so stavom jednotlivých prvkov na stránke
- ID prvku, ktorý vyvolal požiadavku
- Argument, ktorý špecifikuje udalosť, ktorá nastala

V .NET sa funkčný kód sústreďuje na riadenie udalostí. Stlačenie tlačidla, AJAX request, návšteva stránky... to sú udalosti. Keď nastane konkrétna udalosť, spustí sa Event handler, ktorý ju riadi. Dobrým príkladom je Page\_Load. Táto udalosť nastane vždy pri načítaní stránky. Kód v tomto handleri sa teda vykoná vždy. Ak ale chceme vykonať niečo iba pri stlačení konkrétneho tlačidla, vytvoríme si Event handler.

ASPX kód:

```
<asp: Button ID="Tlacidlo" runat="server" Text="stlac ma"
OnClick="TlacidloStlacene" />
```

C# kód:

```
protected void TlacidloStlacene (object sender, EventArgs e)
{
    Tlacidlo.Text = "dakujem za stlacenie";
}
```

Tento príklad po stlačení tlačidla zmení jeho text. Podobným spôsobom vieme spracovať napríklad vybratie prvku v tabuľke, kde v argumente handlera budú všetky potrebné hodnoty (napríklad aj hodnota primárnych kľúčov práve vybraného prvku).

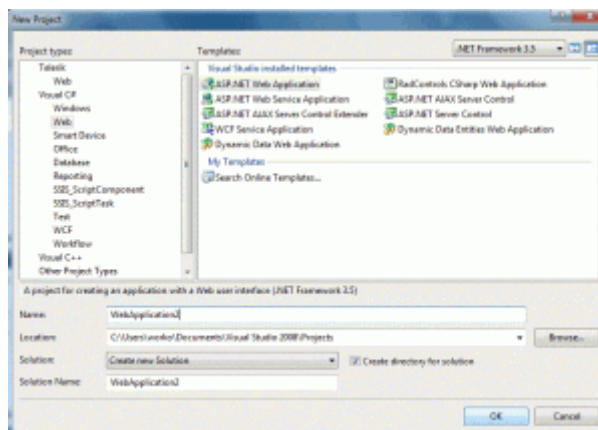
## Životný cyklus stránky

- Page request - Cyklus začína vyžiadaním stránky. Keď je stránka vyžiadaná, ASP.NET určí, či stránka musí byť vytvorená, alebo môže byť odoslaná kešovaná verzia bez spúšťania celého cyklu.
- Start - Nastavia sa základné atribúty stránky. Vytvoria sa objekty Request a Response. Request a Response obsahujú okrem iného všetky parametre odoslaného formulára, prístup k Session State či Cookies. V tejto fáze sa nastaví sa IsPostBack atribút podľa toho, či je požiadavka nová alebo post-back.
- Initialization - Prvky na stránke sa stávajú dostupnými podľa ich unikátneho ID. Ak je požiadavka post-back, hodnoty prvkov na stránke ešte neboli načítané z view state.
- Load - Ak je požiadavka post-back, hodnoty prvkov na stránke sa načítajú z view state.
- Postback event handling - Ak je požiadavka post-back, volajú sa handlersy eventov, ktoré nastali. Potom sa spustia všetky validátory na stránke.
- Rendering - Na začiatku sa uloží nový view state. Potom stránka volá metódu Render pre každý prvok na stránke. Tá vypíše prvky do objektu Response.OutputStream.
- Unload - Potom, ako je stránka odoslaná klientovi, sa zrušia vlastnosti stránky ako Response a Request. Nakoniec sa vykoná cleanup.

## Tvorba jednoduchovej webovej aplikácie

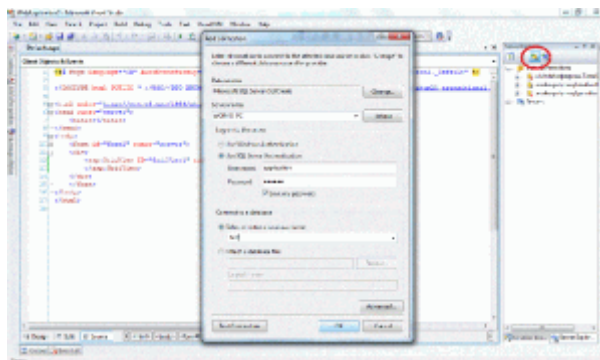
Na príkladoch ukážem jednoduchosť tvorby dynamických stránok vo Visual Studiu, konkrétne jazyku C#.

### 1. Vytvorenie projektu



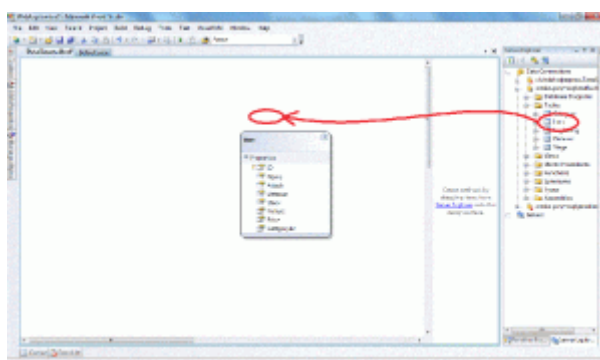
### 2. Pripojenie na databázu

Na obrázku bol do web stránky pridaný element <asp:GridView>, ktorý predstavuje tabuľku. Tento element bude zobrazovať dáta zo svojho DataSource. Spôsob ich zobrazenia je možné meniť.

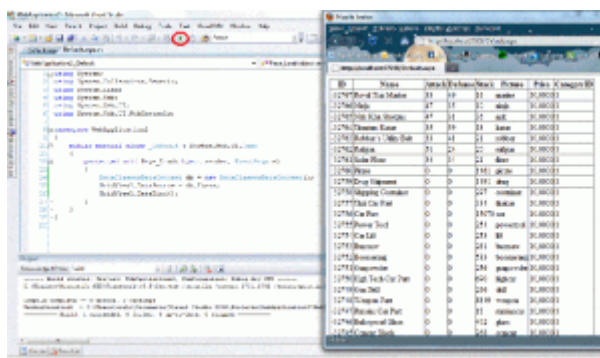


Výsledkom tohto jednoduchého formuláru je connection-string vo web-config, ktorý je možno aj písať ručne alebo prepisovať podľa potreby (napríklad zmena hesla alebo adresy serveru).

### 3. Mapovanie tabuliek na triedy



Prenesením tabuľky zo server exploreru do DataClasses triedy sa automaticky vytvorí trieda s názvom tejto tabuľky obsahujúca všetky jej typy. Vytvorí sa kompletne mapovanie a môžeme hneď začať pristupovať k dátam. Na obrázku je zároveň ukázaná jednoduchosť spustenia kompilácie a debugera.



Postup pri tvorbe takejto jednoduchej aplikácie pozostáva z 3-4 krokov. Nepoznám žiadne vývojové prostredie pre PHP, kde by bolo možné takto jednoducho pristupovať k dátam v databáze. Všetko, aj hlavička tabuľky, je generované automaticky ak to nenastavíme sami tak ako chceme.

