Portál pre odborné publikovanie ISSN 1338-0087

Ext.NET – Pohodlie hrubého klienta v ASP.NET aplikácii

Gubáš Marcel · Informačné technológie, Študentské práce

27.01.2012

Tradičný hrubý klient bol dlhé obdobie štandardom pre informačné systémy. Poskytuje pohodlie užívateľského rozhrania, ktoré je rýchle a spoľahlivé. Menšie a stredné firmy vyvíjajúce softvérové aplikácie na kľúč však čelia tlaku neustálych aktualizácii svojho produktu a špeciálne v prostredí .NET zosúladeniu verzií frameworku na vývojovom prostredí s verziami na klientskych staniciach.

Webové riešenia poskytujú univerzálnosť používanej verzie, pretože na zobrazenie dát používajú štandardizovaný a rozšírený http/https protokol s html tagovacím jazykom a aplikácia je nasadená centrálne na jednom mieste. Pohodlie práce vo web prostredí klesá úmerne so zložitosťou aplikácie a najpalčivejším problémom je rýchlosť odozvy na akciu užívateľa. Riešenie prináša technológia AJAX, ktorá je s knižnicou Ext.NET prístupná aj v ASP.NET prostredí.

1. Ext.NET v skratke

Používanie AJAXu bolo samozrejme možné aj pred príchodom tejto knižnice, ale Ext.NET priniesol nevídané pohodlie použitia. Jedná sa o ASP.NET wrapper pre JavaScript komponenty užívateľského rozhrania. Na jeho používanie stačí stiahnut .dll súbory Ext.Net.dll, Ext.Net.Utilities.dll a .dll súbor tretej strany potrebný na serializáciu v JavaScripte Newtonsoft.Json.dll. Po zareferencovaní do projektu napríklad vo Visual Studiu je knižnica plne k dispozícii. Na začiatok deklaratívnej časti aplikácie je potrebné vsunúť nasledujúcu definíciu:

```
<%@ Register Assembly="Ext.Net" Namespace="Ext.Net" TagPrefix="ext" %>
```

Po preklenutí obdobia učenia sa a spoznávania API poskytuje Ext.NET vysokú efektivitu práce hlavne z dôvodu rovnakej filozofie používania ako má štandardná knižnica UI prvkov v ASP.NET. Súbory sú voľne stiahnuteľné a použiteľné pre nekomerčné účely na stránke výrobcu http://www.ext.net.

2. Dáta sú základom aplikácie

Reprezentácia dát je základným prvkom väčšiny aplikácií, pretože hlavná úloha informačných systémov je narábanie s informáciami. V ASP.NET sa používa bezstavový protokol a akcie s dátami sa vykonávajú až pri post back stránky. Tento

fakt má za následok, že akákoľvek logická operácia nad dátami ako validácia vstupov či selektívne zobrazovanie dát musia byť vykonané na servery, pričom to nie je potrebné. Na server patria len operácie modifikácie dát (kam sa radí aj vytváranie a mazanie) a validácia, ktorá je z bezpečnostných dôvodov nevhodná na klientsku stranu.

Ext.NET prináša akúsi simuláciu stavového protokolu. Vďaka použitému AJAXu dokáže vykonať post back na server na pozadí bez spomaľovania práce na klientskej strane. Základné funkčné prvky je možné zabezpečiť JavaScriptom. Aby sa minimalizoval počet skrytých post backov, používa Ext.NET špecifický systém uchovávania dát. Jednou z prístupných komponent je Store. Jedná sa o základný prvok pri narábaní s dátami. Je možné ho prirovnať k ObjectDataSource, ktorý je používaný v tradičnom ASP.NET.

Definuje sa v deklaratívnej časti aplikácie a v code behind má k dispozícii atribút DataSource a metódu DataBind(). Do DataSource je možné priradiť kolekciu entít vystupujúcu ako dátový zdroj. Obyčajne stačí ak kolekcia implementuje rozhranie IEnumerable a obsahuje jednoduché entity (dáta sú uložené v Property). V prípade zložitých entít je potrebné dodefinovať v deklaratívnej časti pre RecordField atribút ServerMapping s konkrétnou cestou k hodnote v objektovej reprezentácii.

```
store.DataSource = this.Data;
store.DataBind();
```

Store v sebe zahŕňa funkciu dátového zdroja prístupného na klientskej strane a je univerzálny pre všetky prvky užívateľského rozhrania. Je možné ho deklarovať samostatne, alebo ako vnorený atribút ktorejkoľvek komponenty starajúcej sa o zobrazovanie dát (tabuľka, combo box, ...). Dokáže pracovať s tromi rôznymi typmi dátových zdrojov:

```
<ext:ArrayReader>
<ext:JsonReader>
<ext:XmlReader>
```

Pri použití Store sa dáta v ňom uchované serializujú a posielajú na klientsku stranu, čiže užívateľ má k dispozícii kompletnú sadu dát nad ktorou už môže nerušene pracovať. Preto je dôležité pred naplnením Store vyriešiť niektoré základné body.

- Autorizáciu dát
- Kvantitu dát

3. Základné prvky zobrazenia

V Ext.NET knižnici existuje jeden objekt, ktorý uchováva dáta. Volá sa Store a bol

popísaný v predchádzajúcej kapitole. Komponenty zobrazujúce dáta sú len renderovacie mechanizmy.

3.1. Tabuľka

Už pri základnom použití tabuľky je k dispozícii sortovanie stĺpcov a stránkovanie. Nie je potrebné tieto funkčnosti pracne programovať v code behind. Základná definícia obsahuje atribút StoreID, kde je zadané id Store s dátami, ale je možné použiť aj definíciu Store priamo v tele definície tabuľky.

Column model definuje stĺpce tabuľky v súlade s definíciou RecordFields v Store. Prepojenie je zabezpečené pomocou atribútu DataIndex. Je možné definovať spôsob výberu dát nad tabuľkou pomocou SelectionModel atribútu. PagingToolbar zabezpečuje stránkovanie tabuľky a jednoduchou modifikáciou jeho definície v deklaratívnej časti si ho užívateľ dokáže do značnej miery prispôsobiť, ak mu z nejakého dôvodu nevyhovuje základná konfigurácia. Prestránkovanie je zabezpečené pomocou inline JavaScriptu v atribúte Handler, kde sa prisupuje k JS API Ext.NETu.

Vďaka vlastnosti Store, ktorý má k dispozícii kompletnú sadu dát na klientskej strane, nie je zoraďovanie dát podľa stĺpcov či stránkovanie tabuľky pomalé ako pri plnom post back v bežnej ASP.NET technológii. Nespôsobuje ani neobľúbené blikanie stránky pri znovunačítaní.

3.2. Combo box

Výberová ponuka, či rozbaľovacia ponuka patria do širokej sady nepodarených prekladov anglických pomenovaní IT pojmov. Dôležitejšie ako správne pomenovanie combo boxu je poznať spôsob jeho definície v Ext.NETe.

```
<ext:ComboBox
    ID="ComboBox1"
    runat="server"
    StoreID="Store1"
    Width="250"
    Editable="false"
    DisplayField="state"
    ValueField="abbr"
    EmptyText="Vyberte ..."
    SelectOnFocus="true">
    <Template runat="server">
        <Html>
            <tpl for=".">
            <div class="list-item">
             <h3>{state}</h3>
             {nick:ellipsis(8)}, {price:usMoney}
            </div>
            </tpl>
       </Html>
    </Template>
</ext:ComboBox>
```

4. Organizácia rozloženia stránky

Ext.NET neprináša len obohatenie jednotlivých komponent, ale rieši aj celkové rozloženie stránky. Inšpiruje sa pri tom existujúcimi spôsobmi, ktoré sú zavedené v hrubom klientovi.

4.1. ViewPort

Najprv krátka ukážka použitia ViewPortu. Vybrené sú len tie časti, ktoré sú dôležité.

```
<ext:Viewport runat="server" Layout="border">
    <Items>
        <ext:Panel
            Region="North"
                ...
            />
        <ext:Panel
            Region="West"
                ...
        </ext:Panel>
        <ext:TabPanel runat="server" Region="Center">
                ...
        </ext:TabPanel>
        <ext:Panel
            Region="East"
                ...
        </ext:Panel>
        <ext:Panel
            Region="South"
                ...
            />
    </Items>
</ext:Viewport>
```

Pre programátora, ktorý sa niekedy stretol s Java UI knižnicami ako AVG či Swing bude použitá terminológia povedomá. Jeden z možných layout manažérov sa volá border (ako BorderLayout) a používa rovnaké princípy ako jeho vzor v Jave. Priestor stránky vo ViewPorte je rozdelený na 5 častí. Toto delenie je prakticky overené a používa sa na hlavné rozloženie UI aplikácii písaných ako v Jave pod Swingom tak aj v C# pod WinForms. Každá časť má prisúdenú svetovú stranu ku ktorej sa viaže s výnimkou časti center, ktorá vypĺňa priestor medzi nimi. ViewPort umožňuje zadefinovanie výplne týchto vymedzených priestorov v podobe Ext.NET komponent a dokáže im nastaviť veľkosť, či možnosť zmeny veľkosti.

4.2. TabPanel

TabPanel predstavuje vysoký prínos do organizácie formulárov a v počiatkoch sa objavoval hlavne v dialógových oknách obsahujúcivch množstvo ovládacích prvkov,

napr. systémové nastavenia Windows. Neskôr prerazil aj do editačných programov, kde nahradil koncepciu parent window a child Windows, napr. VisualStudio. Len nedávno sa uplatnil vo web prehliadačoch. Jedná sa o záložky, ktoré boli klincom do rakvy nesmrteľného IE6. Ext.NET prináša TabPanel na úroveň web stránky, priamo do renderovaného html kódu.

```
<ext:TabPanel
    ID="TabPanel1"
    runat="server"
    ActiveTabIndex="0"
    Width="600"
    Height="250"
    Plain="true">
    <Items>
        <ext:Panel
            ID="Tab1"
            runat="server"
            Title="Normal Tab"
            Html="My content was added with the Html Property."
            Padding="6"
            AutoScroll="true"
            />
        <ext:Panel
            ID="Tab2"
            ...
            Closable="true"
            />
    </Items>
</ext:TabPanel>
```

5. Zdroje

Uvedené príklady sú len ilustráciou základných možností Ext.NET, v ktorých sa dá zrovnávať jeho prínos oproti ASP.NET komponentám. Ext.NET však umožňuje množstvo pre ASP.NET neprístupných funkčností, kroré sú popísané na domovskej stránke.

1. http://www.ext.net
2. http://examples.ext.net
3. http://docs.ext.net
4. http://www.ext.net/download/