Databáze Caché
CSP Custom Tags

vlastní značky

CSP custom tags

- vývoj vlastních tagů pro CSP stránky
- možnost přidat novou funkcionalitu, ale zachovat stejnou syntaxi
- možnost vyvíjet znovupoužitelné komponenty

CSP kompilátor

- rozpoznání #( )# výrazů připojených v CSP dokumentu
- rozpoznání konkrétních elementů (HTML/XML) a nahrazení předdefinovanými akcemi (analogie XSL)

Příklad: vlastní element

```
<html>                    <html>
<body>                    <body>
<my:COMPANY>   ⇨          <b>Párky Praha</b>
</body>                   </body>
</html>                   </html>
```

Element <my:COMPANY> je při kompilaci stránky nahrazen názvem firmy.

Příklad: vlastní element (2)

- Aby byl element kompilátorem rozpoznán, je třeba vytvořit patřičné pravidlo, které se při kompilaci provede.

```
<csr:RULE name="myCOMPANY" match="my:COMPANY" empty>
<csr:ACTION>
<b>Párky Praha</b>
</csr:ACTION>
</csr:RULE>
```

- Název pravidla podléhá stejným omezením jako název třídy v Caché.

<csr:ACTION>

- specifikuje akci, která bude provedena, pokud pravidlo vyhovuje na nějaký element.
- obsah je interpretován jako HTML (až na dvě výjimky)
  - výrazy #()# a ##()##, element <script>
  - další <csr:> elementy
- nemůže obsahovat takový CSP tag, pokud v daném kontextu není možné provést transformaci

Načtení souboru pravidel

- v terminálu příkazem
  - – načtení pravidel do aktuálního kontextu

  Do $system.CSP.LoadRule("/myrules/company.csr")
- Caché Studio
  - – nahrát soubor a zkompilovat
  - – sledovat, která pravidla se používají souborem *rulemgr.csp* dodaným v CSP Samples

Milan Kryl (c) 2004 MFF UK

Tag matching

- atribut MATCH obsahující 1 nebo více elementů, oddělené /

Hodnota v MATCH	Pravidlo aplikováno
AAA	Kdekoliv je tag <AAA></AAA>
AAA/BBB	Kdykoliv <AAA> obsahuje <BBB> (<AAA><BBB></BBB></AAA>)
AAA/*/BBB	Kdykoliv <AAA> obsahuje libovolně hluboko zanořený <BBB> (<AAA><C><D><BBB> </BBB></D></C></AAA>)
AAA[CCC]	Pokud <AAA> má atribut CCC s lib. hodnotou. (<AAA CCC="10"></AAA>)
AAA[CCC=22]	Pokud <AAA> má atribut CCC s hodnotou 22. (<AAA CCC="22"></AAA>)
AAA[CCC=22]/*/BBB	<BBB> zanořený lib. hluboko v elementu <AAA> s atributem CCC rovným 22.

Milan Kryl (c) 2004 MFF UK

Test: pochopení MATCH

- text text text **text** text
- text **text** text
- text text text text text
- Označ elementy v elementech
  1. MATCH = "p/b"
  2. MATCH = "p/*/b" ⬅
  3. MATCH = "p/i/b"

Milan Kryl (c) 2004 MFF UK

Test pochopení MATCH (2)

- **text** text
- <a href="/">**text** text
- <a href="/uroven/">**text** text
- Označit všechny odkazy na hl. stránku "/"
  1. MATCH = "p/b/a"
  2. MATCH = "p/a[href=/]/"
  3. MATCH = "p/*/a[href=/]/" ⬅

Milan Kryl (c) 2004 MFF UK

Programový kód v ACTION

- akce v pravidlech mohou obsahovat programový kód, který bude vykonán
  - – když je pravidlo aplikováno (run-time)
  - – když je pravidlo kompilováno (compile-time)

Milan Kryl (c) 2004 MFF UK

Programový kód v ACTION (2)

- vykonání při spuštění - syntaxe #(expr)#
  - – pravidlo

  ```
  <csr:RULE name="TODAY" match="TODAY" empty>
  <csr:ACTION>
  Dnes je: <b>#($ZDATE($H))#</b>
  </csr:ACTION></csr:RULE>
  ```

  - – pro kód

  ```
  <TODAY>
  ```
- vyvolá příkaz $ZDATE pro element <TODAY> při zaslání stránky

Milan Kryl (c) 2004 MFF UK

Programový kód v ACTION (3)

- vykonání při kompilaci - syntaxe ##(expr)##
  - pravidlo

```
<csr:RULE name="TODAY" match="TODAY" empty>
<csr:ACTION>
Dnes je: <b>##($ZDATE($H))##</b>
</csr:ACTION></csr:RULE>
```

  - pro kód

```
<TODAY>
```

- vyvolá příkaz $ZDATE pro element <TODAY> v době kompilace

Programový kód v ACTION (4)

- výrazy provedené v době kompilace se stávají statickou částí CSP stránky.
- je možné kombinovat "statické" a "dynamické" vykonávání kódu:

  Stáří stránky (dny): #($H - ##($H)##)#.
- vnitřní část je vyhodnocena při kompilaci a vnější se volá při zaslání stránky ze serveru

Programový kód v ACTION (5)

- dále je možno vkládat více řádků kódu za pomoci elementu <script> a atributu runat (server/compiler)

```
<csr:RULE name="BIGLIST" match="BIGLIST" empty>
<csr:ACTION>
<ul>
<script language="CACHE" runat=server>
For i = 1:1:100 {
    Write "<li>Položka " _ i _ $C(13,10)
}
</script>
</ul>
</csr:ACTION>
</csr:RULE>
```


Server Document Object Model

- vytvářen při kompilaci CSP dokumentu (analogie DOM)
- dva typy základních atomů

  %CSP.Rule - elementy HTML

  %CSP.TextAtom - vše ostatní co není element
- kvůli efektivitě se %CSP.Rule vytváří pouze pokud je zmíněno v některém z pravidel, jinak je vše %CSP.TextAtom

Server DOM (2)

```
<html>
<body>
Ahoj!
<MYTAG MSG="Vítejte">
</body>
</html>
```

Při kompilaci vytvořeno

- html
  - body
    - TextAtom s textem Ahoj!
    - element se jménem MYTAG a atributem MSG rovným Vítejte

Server DOM (3)

- po vytvoření DOMu je celý stromeček procházen (do hloubky)
  - na každý uzel typu %CSP.Rule je voláno pravidlo a renderován výsledek
  - %CSP.TextAtom jsou přenášeny přímo do výsledného souboru
- při volání pravidla se na akt. zpracovávaný uzel odkazuje za pomoci ##this proměnné

Další csr: elementy (vnitřní)

- csr:DEFAULT
  - vypíše element na výstup (při změnách některých atributů, ale zachování tagu)
- csr:CHILDREN
  - vypíše všechny potomky tagu, na který bylo vyvoláno pravidlo
- csr:SECTION
  - výstup na jiné místo do výsledného dokumentu (definice funkce pro form do hlavičky)

Milan Kryl (c) 2004 MFF UK

csr:DEFAULT

```
<csr:RULE name="REDTABLE" match="TABLE" >
<csr:ACTION>
<script language="CACHE" runat="COMPILER">
   // pro tento element nastav bgcolor atribut na red
   Do ##this.SetAttribute("BGCOLOR","red")
</script>
<csr:DEFAULT>
</csr:ACTION>
</csr:RULE>
```

Milan Kryl (c) 2004 MFF UK

csr:SECTION

```
<csr:RULE name="MYBUTTON" match="FORM/*/MYBUTTON" empty>
<csr:ACTION>
<csr:SECTION NAME=HEAD>
<script language="JavaScript">
function MyButton() {
    alert('Tlačítko MyButton stisknuto!');
    return true;
}
</script>
</csr:SECTION>

<input type="button" value='##(##this.GetAttribute("VALUE"))##'
onclick="MyButton();"></input>

</csr:ACTION>
</csr:RULE>
```

Milan Kryl (c) 2004 MFF UK

Další csr: elementy (vnější)

- csr:CLASS
  - zpřístupnění další metod vně pravidla
- csr:DESCRIPTION
  - umožňuje k pravidlu přidat vysvětlující popis
- csr:ATTRIBUTE
  - název, popis a typ atributů vlastního elementu
  - <csr:attribute name=Type description="Specify the default Content-Type" type="contentType:STRING">

Milan Kryl (c) 2004 MFF UK

Třídy pravidel

- pro každé kompilované pravidlo vytvořena třída, volaná při platnosti pravidla
- pravidla mohou být
  - výkonná
  - přímo vytvářena jako třídy
  - třídy lze zobrazit a vytvářet za pomoci Studia

Milan Kryl (c) 2004 MFF UK

Struktura třídy pravidel

- renderovaná třída obsahuje metody
  - RenderStartTag
  - 1 a více CompilerMethod (pro kód runat=compiler)
  - RenderEndTag
- Render[Start|End]Tag obsahují kód pro přímý zápis do CSP stránky.

Milan Kryl (c) 2004 MFF UK

Kód RenderStartTag

```
<csr:ACTION>
<script language="CACHE" runat=server>
    Set myfile="c:\temp.txt"
    Open myfile:("FR":100)
    Use myfile:()
    Read var1
    Close myfile
</script>
</csr:ACTION>
```

Milan Kryl (c) 2004 MFF UK

Výsledná metoda RenderStartTag

```
Method RenderStartTag() As %Status
{
    New element Set element=##this
    Do ..WriteCSPServer(" Set myfile=""c:\temp.txt""",0)
    Do ..WriteCSPServer(" Open myfile:(""FR"":100)",1)
    Do ..WriteCSPServer(" Use myfile:() ",1)
    Do ..WriteCSPServer(" Read var1",1)
    Do ..WriteCSPServer(" Close myfile",1)
    Quit $$$SKIPCHILDREN
}
```

- pokud by pravidlo obsahovalo <csr:CHILDREN> tak je metoda ukončena Quit $$$PROCESSCHILDREN

Milan Kryl (c) 2004 MFF UK

CompilerMethod

- kód

```
<script language="CACHE" runat=compiler>
SET ^client(2,1,1)=..InnerText()
</script>
```

- výsledná metoda - nakopírována do těla

```
Method CompilerMethod1() [ Language = cache ]
{
SET ^client(2,1,1)=..InnerText()
}
```

Milan Kryl (c) 2004 MFF UK

RenderEndTag

- generována, pokud je v pravidle obsažen element <csr:CHILDREN>
- obsahuje kód uvedený za <csr:CHILDREN>

Milan Kryl (c) 2004 MFF UK

Kód RenderEndTag

```
<csr:rule name="iscbarchart" match="isc:barchart"
    language="any">
<csr:action>
<table bgcolor='##(..GetAttribute("BGCOLOR"))##'
    border=0 cellspacing=0 style='border:
    ##(..GetAttribute("BORDER","solid blue"))##;'><tr>
<csr:children>
</tr></table>
</csr:action>
</csr:rule>
```

Milan Kryl (c) 2004 MFF UK

```
Method RenderEndTag() As %Status
{
    New element Set element=##this
    Do ..WriteText("",1)
    Do ..WriteCSPText("</tr></table>",0)
    Quit $$$OK
}
```

Milan Kryl (c) 2004 MFF UK

Metody %CSP.Rule

- GetAttribute(name As %String, default As %String = "")
  - získá hodnotu atributu z aktuálního tagu
- QuoteAttribute(name As %String, default As %String = "")
  - připraví hodnotu včetně provedení #()#, ##()## a ##"##
- GetAttributesOrder(ByRef paramsordered)
  - vrátí všechny atributy podle jejich po řadí v tagu
- IsDefined(name As %String)
  - zjistí zda je daný atribut definován
- InnerText()
  - vrátí obsah od počátečního po koncový element
- AddChildElements(atom As %CSP.AbstractAtom)
  - přidá k aktuálnímu elementu podelement zadaný jako parametr
- SetAttribute(name As %String, value As %String)
  - nastaví atribut na nějakou hodnotu

Milan Kryl (c) 2004 MFF UK

Metody zápisu %CSP.AbstractAtom

- WriteText(line As %String, crlf As %Boolean = 0)
  - zapiš obsah line, crlf = 1 pokud má být zapsán konec řádku
- WriteCSPText(line As %String, crlf As %Boolean = 0)
  - zápis řádku s prováděním ##()##, ##"##, #server, #url, a #()# výrazů
- WriteExpressionText(expr As %String, crlf As %Boolean = 0)
  - zapíše text, který je vrácen zadaným výrazem (měl by být korektně quoted)
- WriteServer(line As %String, keepTogether As %Boolean = 0)
  - zapíše COS příkaz
- WriteCSPServer(line As %String, keepTogether As %Boolean = 0)
  - zapíše COS příkaz s vyhodnocenými ##()##, #()#, a ##"## výrazy.

Milan Kryl (c) 2004 MFF UK

National Language Support (NLS)

- seznam všech nativně podporovaných jazyků

- 5 - Cache standard
- 10 - German1
- 11 - Portuguese1
- 12 - Polish1
- 13 - German2
- 14 - Spanish1
- 15 - Danish1
- 17 - Greek1
- 18 - Czech1
- 19 - Czech2
- 20 - Portuguese2
- 21 - Finnish1
- 22 - Japanese1
- 23 - Cyrillic2
- 24 - Polish2
- 26 - Chinese2

Milan Kryl (c) 2004 MFF UK

Děkuji za pozornost

nějaké otázky?

Milan Kryl (c) 2004 MFF UK