

C# - RSS-Feed mit XPath verarbeiten

Im Folgenden wollen wir mithilfe von C# und XPath den RSS-Feed von der Tagesschau-Website (Tagesschau.de) verarbeiten und uns stündlich die Titel der aktuellen Tagesthemen ausgeben lassen.

Vorbereitung

Zu Beginn sollte sichergestellt werden, dass die zusätzlich benötigten Namespaces eingebunden werden:

Quellcode

1. using System.Timers;
2. using System.Xml.XPath;

Timer bereitstellen

Damit wir stündlich den aktuellen RSS-Feed erhalten, bietet es sich an einen Timer zu verwenden, der alle 60 Minuten ein Ereignis auslöst.

Quellcode

```
1. try
2. {
3. // neue Instanz der Timer-Klasse erzeugen
4. Timer timer = new Timer();
5. // hier legen wir unser Timer-Intervall fest
6. timer.Interval = TimeSpan.FromMinutes(60).TotalMilliseconds;
7. // Ereignishandler an Ereignis binden
8. timer.Elapsed += new ElapsedEventHandler(DisplayFeedTitles);
9. // Timer auslösen
10. timer.Start();
11. // zum beenden des Programms 'q' drücken
12. while (Console.Read() != 'q') ;
13. }
14. catch (Exception error)
15. {
16. Console.WriteLine("Es ist ein Fehler aufgetreten: {0}", error.Message);
17. }
```

Alles anzeigen

Der Code sollte denke ich leicht verständlich sein. Die Angabe des Intervalls hätte man natürlich auch direkt in Millisekunden oder aber auch mittels

Quellcode

1. TimeSpan.FromHours(1).TotalMilliseconds;

angeben können. Wer mit Ereignissen und Delegationen noch nicht ganz vertraut ist, der sollte sich mal den dazugehörigen Abschnitt im Galileo Openbook [Visual C# 2008](#) durchlesen.

RSS-Feed mit XPath verarbeiten

Jetzt kommt der eigentlich Code, der dafür verantwortlich ist uns die jeweiligen Titel der aktuellen Tagesthemen auf der

Inhaltsverzeichnis

- [1 Vorbereitung](#)
- [2 Timer bereitstellen](#)
- [3 RSS-Feed mit XPath verarbeiten](#)

Konsole auszugeben.

Quellcode

```
1. public static void DisplayFeedTitles(object source, ElapsedEventArgs e)
2. {
3. // neue Instanz der XPathDocument-Klasse erstellen - hier übergeben wir dem Konstruktor direkt die Feed-Adresse
4. XPathDocument xpathDocument = new XPathDocument("http://www.tagesschau.de/xml/rss2");
5. // Erzeugung eines Objektes zum navigieren durch die Knoten im Feed (XML-Knoten)
6. XPathNavigator xpathNavigator = xpathDocument.CreateNavigator();
7. // alle Titel selektieren, die sich innerhalb der Eltern-Knoten <item></item> befinden
8. XPathNodeIterator xpathIterator = xpathNavigator.Select("//item/title");
9. // prüfen, ob überhaupt Titel gefunden wurden
10. if (xpathIterator.Count > 0)
11. {
12. Console.WriteLine("Die aktuellen Tagesthemen:");
13. // durch die Titel-Knoten durchiterieren und den aktuellen Wert/Text des jeweiligen Titel-Knoten ausgeben
14. while (xpathIterator.MoveNext())
15. {
16. Console.WriteLine(xpathIterator.Current.Value);
17. }
18. }
19. else
20. {
21. Console.WriteLine("Keine Tagesthemen vorhanden");
22. }
23. xpathNavigator = null;
24. xpathDocument = null;
25. }
```

Alles anzeigen

Man könnte hier nun noch viele Sachen erweitern und z.B. auch den entsprechenden Link zum jeweiligen Titel mit ausgeben lassen. Wer sich schon mal mit dem XMLTextReader in C# beschäftigt hat, für den ist die Variante XML-Dateien mit XPath zu verarbeiten vielleicht eine gute Alternative, um performanteren Code zu schreiben.

Nebenbei sei gesagt, dass in meinem Beispiel bei jedem Aufruf von DisplayFeedTitles() neue Instanzen der jeweiligen XPath-Klassen erzeugt werden. Hier sollte man natürlich noch nachbessern, so dass nur eine einzige Instanz erzeugt wird. Der XPathNavigator braucht dann nur immer beim Aufruf von DisplayFeedTitles() zurück auf den Stammknoten gesetzt werden:

Quellcode

```
1. xpathNavigator.MoveToRoot();
```