

## Abfrage von Twitter-Nachrichten mit XSLT und PHP

# Spatz in der Hand

Nicht nur die Spatzen zwitschern munter drauflos. Auch beim Microblogging-Dienst Twitter [1] erfreut sich das „Zwitschern“ einer stetig wachsenden Beliebtheit. Da Twitter die bis zu 140 Zeichen langen Kurzmitteilungen (Tweets) auch über XML-Schnittstellen ausliefert, soll hier eine praktikable Möglichkeit zur Verarbeitung mit XSLT und PHP vorgestellt werden.

von Thomas Meinike

Twitter ermöglicht den externen Datenzugriff sowohl über JSON als auch in mehreren XML-Darreichungsformen [2] wie RSS und Atom Feeds sowie über ein generisches und vergleichsweise umfangreiches XML-Format. Letzteres hat folgende formale Struktur:

```
<statuses type="array">
<status>
<created_at>Fri Jul 03 16:04:41 +0000 2009
</created_at>
<id>2455715311</id>
<text>Firefox- schlägt Safari-Download nicht http://bit.ly/UKgY0</text>
<!-- ... -->
<user>
<id>5895222</id>
<name>entwickler</name>
<!-- ... -->
</user>
</status>
</statuses>
```

Innerhalb des Wurzelements *statuses* werden die einzelnen Tweets jeweils in ein *status*-Kindelement verpackt, dem wiederum die eigentlich relevanten Informationen in den Elementen *created\_at* und *text* untergeordnet sind. Mehr braucht man für die Einbindung der letzten getwitterten Neuigkeiten in die eigene Website nicht. Das genannte XML-Format ist über die Adressierung `http://twitter.com/statuses/user_timeline/Username.xml?count=anzahl` zugänglich. Als Username dient der jeweilige Twitter-Anmeldungsname (etwa *entwickler*) und die gewünschte Anzahl abzufragender Tweets kann über den Parameter *count* eingestellt werden, z. B. *count=5*. Statt

*xml* kann die Adresse auch *rss* oder *atom* bzw. *json* enthalten. Das RSS-Format ist besonders kompakt. Es liefert pro Tweet ein *item*-Element, das wiederum die benötigten Inhalte kapselt:

```
<item>
<title>entspricht description</title>
<description>entwickler: Firefox- schlägt Safari-Download nicht http://bit.ly/UKgY0</description>
<pubDate>Fri, 03 Jul 2009 16:04:41 +0000</pubDate>
<!-- ... -->
</item>
```

Ein Tweet lässt sich also durch die Auswertung der Elemente *description* und *pubDate* in Zielformate übernehmen. Die Abbildungen 1 und 2 zeigen die Browseransicht der aufgerufenen Ressourcen.

### Tweets abfragen und darstellen

Im Grunde ist es sehr einfach, Informationen über die genannte Adressierung von

Twitter abzufragen. Im Beispiel kommt PHP 5 zum Einsatz, um einerseits mit *file\_get\_contents(\$url)* die XML-Dokumente abzuholen und andererseits mittels XSLT-Funktionen die Transformation in eine (X)HTML-Ausgabe anzustoßen.

Es wurden XSLT-Stylesheets für die genannten XML- und RSS-Formate erstellt. Darin werden im Wesentlichen die Tweet-Texte ausgelesen und die Datumswerte mit Zeichenkettenfunktionen wie *substring()* verarbeitet. Außerdem erfolgt die Umwandlung der in den Nachrichten enthaltenen URLs in aktive Links (*a*-Elemente) mit einem als *tweet\_text* benannten und rekursiv aufgerufenen XSLT-Template. Im Beispielcode wird lediglich ein HTML-Fragment, `<div id="twitter_news">...</div>`, zur Einbindung in weiteren HTML- bzw. PHP-Code erzeugt. Dieses Fragment enthält eine passende Überschrift und innerhalb einer *ul*-Liste pro Tweet ein *li*-Element.

### Listing 1

#### PHP-Funktion TwitterNews()

```
function TwitterNews()
{
// Username ($user) und Abfragetyp ($type) sowie
// Tweet-Anzahl ($tweets) anpassen:
$user="Username";
$type="rss"; // "rss" bzw. "xml"
$tweets=5;

$xmlurl="http://twitter.com/statuses/user_
timeline/" . $user . "." . $type . "?count=" . $tweets;

$xml=document=""; // wird von Twitter geladen;
$xml=document=file_get_contents($xmlurl);
// XSLT-Stylesheets auf dem eigenen Server ablegen:
$xmlstylesheet="twitter_" . $type . ".xsl";

// Transformation
if($xml=document && $xsl=document)
{
$xmllobj=new DomDocument();
$xmllobj->loadxml($xml=document);
$xsllobj=new DomDocument();
$xsllobj->loadxml(file_get_contents($xsl=document));

$xslstref=new XSLTProcessor;
$xslstref->importStylesheet($xsllobj);
$xml_output=$xslstref->transformToXML($xmllobj);

return $xml_output;
}
}
```

In Listing 1 ist der Code der PHP-Funktion *TwitterNews()* ersichtlich. Diese lässt sich über das Skript *transform.php* einbinden und ausführen (Listing 2). Den für die RSS-Variante entworfenen XSLT-Code enthält Listing 3. Die jeweilige Zuweisung erfolgt in der PHP-Funktion über die Variable *\$type*. Entsprechende Anpassungen sind auch für *\$user* und *\$tweets* vorzunehmen. Die erzeugte HTML-Ausgabe kann durch zusätzliche CSS-Formatierungen an die eigenen Layoutvorgaben angepasst werden.

**Fazit**

Mit den vorgestellten Techniken lassen sich über Twitter verbreitete Mitteilungen sehr einfach verwerten und in Websites integrieren. Somit können Reichweite und Potenzial von Twitter deutlich erhöht werden. Der komplette Beispielcode mit weiteren Hinweisen in *readme.txt* steht auf der Heft-DVD sowie unter [3] zur Verfügung.

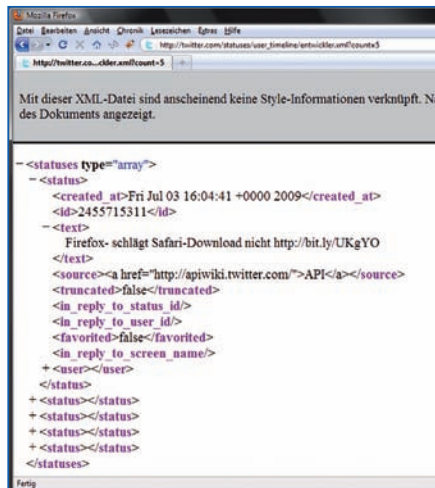


Abb. 1: Das Twitter-XML-Format führt zur Codeansicht



Abb. 2: Als RSS Feed sind die Tweets auch direkt lesbar



**Dr. Thomas Meinike** ist seit 1997 an der Hochschule Merseburg (FH) als Lehrkraft tätig. Seine Arbeitsschwerpunkte sind XML-Anwendungen in der Technischen Dokumentation, Onlinehilfen und Webentwicklung. Er verfasst regelmäßig Fachartikel und hält Vorträge zu Themen im XML-Umfeld, twittert als XMLArbyter. Kontakt: [thomas.meinike@hs-merseburg.de](mailto:thomas.meinike@hs-merseburg.de).

**Links & Literatur**

- [1] Twitter: [twitter.com](http://twitter.com)
- [2] Twitter-REST-API: [apiwiki.twitter.com/](http://apiwiki.twitter.com/)  
Twitter-REST-API-Methode:-statuses-user\_timeline
- [3] Beispielcode: [datenverdrahten.de/xslt/twitter\\_xml.zip](http://datenverdrahten.de/xslt/twitter_xml.zip)

**Listing 2**

**HTML-Ausgabe mit XSL-Transformation**

```
<?php
header("Content-type: text/html; charset=UTF-8");
require_once("transform.php");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="de" lang="de">
<head>
<title>Beispiel zur Abfrage von Tweets via XML/XSLT
und PHP</title>
<meta http-equiv="Content-Type" content="text/
html; charset=UTF-8" />
</head>
<body>
<h1>Beispiel zur Abfrage von Tweets via XML/XSLT und
PHP</h1>
<?php
// Transformation ausführen und Ergebnisfragment
ausgeben
echo TwitterNews();
?>
</body>
</html>
```

**Listing 3**

**XSLT-Stylesheet für RSS-Transformation**

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.
w3.org/1999/XSL/Transform">
<xsl:output method="xml" encoding="UTF-8" omit-xml-
declaration="yes" indent="yes" />
<xsl:variable name="username" select=
"concat(substring-after(rss/channel/link,
'http://twitter.com/'), ': ')" />
<xsl:template match="rss">
<div id="twitter_news">
<h2>Twitter-News</h2>
<ul>
<xsl:apply-templates select="channel/item" />
</ul>
</div>
</xsl:template>
<xsl:template match="item">
<li>
<xsl:call-template name="tweet_text">
<xsl:with-param name="t" select="concat(normalize-
space(substring-after(description,$username)), ' ')" />
</xsl:call-template>
<em><xsl:text>[</xsl:text><xsl:value-of select="sub
string(pubDate,1,16)" /><xsl:text>]</xsl:text></em>
</li>
</xsl:template>
</xsl:stylesheet>
```