

HANSER

# PHP objektorientiert

Peter Lavin

Konzepte, Techniken, Code

ISBN 3-446-40762-6

Leseprobe

Weitere Informationen oder Bestellungen unter  
<http://www.hanser.de/3-446-40762-6> sowie im Buchhandel

## RSS

Leider gibt es zahlreiche Versionen von RSS. Wählen wir einen pragmatischen Ansatz und ignorieren wir die verworrenen Details der historischen Entwicklung von RSS. Wenn etwas neu ist, sollte man am besten immer mit einem ganz einfachen Beispiel beginnen, und die einfachste Version von RSS ist Version 0.91. Diese Version gilt zwar offiziell als veraltet, wird aber trotzdem weiterhin in großem Stile eingesetzt, und Kenntnisse ihrer Struktur stellen eine gute Grundlage für die Migration auf Version 2.0 dar – Ihre Mühen sind also nicht umsonst. Ich werde Ihnen ein Beispiel für eine RSS 0.91-Datei vorstellen. Ehrlich gestanden handelt es sich dabei sogar um genau den RSS-Feed, den wir zur Anzeige von Neuigkeiten auf einer Webseite verwenden werden.

## 12.2 Struktur einer RSS-Datei

---

Wie wir es bereits früher mit eigenem Code getan haben, wollen wir auch den RSS-Code Schritt für Schritt durchgehen und an den passenden Stellen Anmerkungen fallen lassen.

Die allererste Komponente einer XML-Datei ist die Versionsdeklaration. Diese Deklaration zeigt eine Versionsnummer und kann – wie im folgenden Beispiel – zudem Angaben zur Zeichenkodierung enthalten.

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

Auf die XML-Versionsdeklaration folgt in der nächsten Codezeile das allererste Element des Dokuments. Der Name dieses Elements definiert den Typ des XML-Dokuments. Aus diesem Grund heißt dieses Element auch *Dokumentelement* oder *Wurzelement*. Keine Überraschung: Unser Dokumenttyp ist RSS. Dieses Einleitungselement definiert die RSS-Versionsnummer und hat ein passendes schließendes Tag, welches das Dokument auf ähnliche Weise abschließt, wie `<html>` und `</html>` zum Öffnen und Schließen einer Webseite verwendet werden.

```
<rss version="0.91">
```

Ein korrekt formatiertes RSS-Dokument erfordert genau ein `channel`-Element. Dieses Element wird die Metadaten zum Feed sowie die eigentlichen Daten aufnehmen, aus denen der Feed besteht. Ein `channel`-Element erfordert drei Unterelemente: `title` (Titel), `link` (Verknüpfung) und `description` (Beschreibung). In unserem Code werden wir das Element `channel title` extrahieren, um einen Header für unsere Webseite zu erstellen.

```
</channel>
<channel>
  <title>About Classical Music</title>
  <link>http://classicalmusic.about.com/</link>
  <description>Get the latest headlines from the About.com Classical
Music Guide Site.</ description>
</channel>
```

Die Unterelemente `language` (Sprache), `pubDate` (Veröffentlichungsdatum) und `image` (Bild) enthalten optionale Metadaten zum Kanal.

```
<language>en-us</language>
<pubDate>Sun, 19 March 2006 21:25:29 -0500</pubDate>
<image>
  <title>About.com</title>
  <url>http://z.about.com/d/lg/rss.gif</url>
  <link>http://about.com/</link>
  <width>88</width>
  <height>31</height>
</image>
```

Das nachfolgende Element `item` ist das, woran wir eigentlich interessiert sind. Die drei erforderlichen Elemente eines Items sind diejenigen, die auch hier auftauchen: `title`, `link` und `description`. Dies ist der Teil des RSS-Feeds, der den Inhalt unserer Webseite bilden wird. Wir werden mit den Elementen `title` und `link` ein HTML-Tag vom Typ `anchor` erzeugen und auf dieses die Beschreibung folgen lassen.

```
<item>
  <title>And the Oscar goes to...</title>
  <link>http://classicalmusic.about.com/b/a/249503.htm</link>
  <description>Find out who won this year's Oscar for Best Music...
  </description>
</item>
```

Hier ist nur ein Item aufgeführt, es können beliebig viele erscheinen. In einem typischen RSS-Feed sind etwa zwanzig Items enthalten.

```
</channel>
</rss>
```

Hiermit wird das `channel`-Element und nachfolgend das `rss`-Element abgeschlossen. Diese Tags sind ordnungsgemäß verschachtelt, und für jedes Tag ist ein passendes End-Tag vorhanden. Wir können also festhalten, dass dieses XML-Dokument *wohlgeformt* ist.

### Den Feed lesen

Um diesen Feed lesen zu können, übergeben wir seinen URI an die Funktion `simplexml_load_file` und erzeugen ein Objekt `SimpleXMLElement`. Dieses Objekt verfügt über vier eingebaute Methoden und so viele Eigenschaften oder Datenmitglieder wie seine XML-Quelldatei.

```
<?php
//point to an xml file
$feed = "http://z.about.com/6/g/classicalmusic/b/index.xml";
//create object of SimpleXMLElement class
$xml = simplexml_load_file($feed);
```

Wir können die Methode `attributes` verwenden, um die RSS-Versionsnummer aus dem Wurzelement zu extrahieren.

```
foreach ($xml->attributes() as $key => $value){
    echo "RSS $key $value";
}
```

Der `channel title` kann auf OO-Weise als verschachtelte Eigenschaft referenziert werden. Beachten Sie jedoch, dass wir `$xml->channel->title` aus einem Segment in Anführungszeichen heraus nicht referenzieren können, da es sich um einen komplexen Ausdruck handelt. Eine alternative Syntax unter Verwendung geschweiffter Klammern ist im nachfolgenden Kommentar enthalten.

```
echo "<h2>" . $xml->channel->title . "</h2>\n";
// Das folgende funktioniert nicht
//echo "<h2>$xml->channel->title</h2>\n";
//may use the syntax below
//echo "<h2>{$xml->channel->title}</h2>\n";echo "<p>\n";
```

Wie Sie sich wohl schon denken können, unterstützt `SimpleXMLElement` die Iteration.

```
//iterate through items as though an array
foreach ($xml->channel->item as $item){
    $strtemp = "<a href=\"{$item->link}\">".
        "{$item->title}</a> {$item->description}<br /><br />\n";
    echo $strtemp;
}
?>
</p>
```

Ich sagte ja bereits, dass es ganz einfach ist, aber ich könnte wetten, dass Sie nicht erwartet haben, derart wenige Codezeilen vorzufinden. Mit nur grundlegendstem Verständnis der Struktur einer RSS-Datei ist es uns gelungen, einen RSS-Feed in eine Webseite einzubetten.

Die SimpleXML-Erweiterung zeichnet sich vor allem unter Umständen wie diesen aus, bei denen die Dateistruktur bereits im Vorhinein bekannt ist. Wir wissen, dass wir es mit einer RSS-Datei zu tun haben und dass diese Datei bestimmte Elemente enthalten muss, wenn sie wohlgeformt ist. Andererseits wäre, wenn wir das Dateiformat, mit dem wir es zu tun haben, nicht kennen, die SimpleXML-Erweiterung völlig ungeeignet. Ein `SimpleXMLElement` kann eine XML-Datei nicht abfragen, um ihre Struktur zu bestimmen. Wie sich dem Namen bereits entnehmen lässt, ist SimpleXML die am einfachsten zu verwendende XML-Erweiterung. Für komplexere Interaktionen mit XML-Dateien müssen Sie entweder die DOM- (Document Object Model) oder die SAX-Erweiterungen (Simple API for XML) verwenden. In jedem Fall ist PHP 5 mit der SimpleXML-Erweiterung seiner Herkunft treu geblieben und hat eine einfache Möglichkeit bereitgestellt, Dinge zu tun, die andernfalls ziemlich komplex wären.