entwickler.press

XML Standards

Tobias Hauser

schnell + kompakt

Tobias Hauser

XML-Standards

schnell+kompakt

entwickler.press

Tobias Hauser
XML-Standards
schnell+kompakt
ISBN: 978-3-86802-236-0

ein Imprint der Software & Support Media GmbH
2. vollständig aktualisierte Auflage

http://www.entwickler-press.de
http://www.software-support.biz

Ihr Kontakt zum Verlag und Lektorat: lektorat@entwickler-press.de

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Projektleitung: Sebastian Burkart
Korrektorat: Ella Klassen, Katharina Klassen
Satz: Pobporn Fischer
Umschlaggestaltung: Maria Rudi
Belichtung, Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, Paderborn.

Inhaltsverzeichnis

Vorwort

XML ist heute nicht mehr aus dem IT-Alltag wegzudenken. XML werkelt in Konfigurationsdateien und als Im- und Exportformat für verschiedenste Programme. Mit AJAX werden XML-Dokumente übermittelt, Web Services sind selbst XML und transportieren XML-Inhalte. Viele Datenbanken produzieren und speichern mittlerweile ihre Daten in XML-Form.

Obwohl XML erfolgreich eingesetzt wird, kommen viele Anwender und Entwickler oft lange Zeit ohne XML aus. Bis ihnen eine XML-basierte Technologie wie XHTML, XSLT, SVG, MathML, SOAP oder XSL:FO über den Weg läuft und sie sich fragen, wie XML eigentlich funktioniert.

Genau hier kommt dieses Buch ins Spiel. Denn auf den folgenden 100 Seiten der zweiten Auflage lernen Sie XML und die wichtigsten Standards zur Arbeit mit XML-Dokumenten kennen und nutzen. Gemäß des Konzepts dieser Buchreihe „schnell + kompakt" erfahren Sie alles Notwendige in den folgenden Kapiteln:

- Kapitel 1 führt in das Grundkonzept von XML ein, zieht Lehren aus der Historie und gibt einen Überblick über das XML-Universum.
- Kapitel 2 zeigt, wie Sie korrekte XML-Dokumente erstellen, diese mit DTD oder XML Schema strukturieren und anschließend prüfen. Daneben lernen Sie das Konzept der Namensräume (Namespaces) und Zeichensätze kennen.
- Kapitel 3 stellt XSLT vor, die Sprache, um XML-Dokumente in andere XML-Dokumente oder andere Formate wie HTML

umzuwandeln. XPath hilft bei XSLT und erlaubt den direkten Zugriff auf Elemente im XML-Dokument. Zu guter Letzt sorgt XSL:FO dafür, dass aus XML-Dokumenten Druckseiten beispielsweise im PDF-Format werden.

- Kapitel 4 kümmert sich um den XML-Zugriff per Programmierung. Sie lernen verschiedene Zugriffsmethoden kennen und erfahren, in welchen Programmiertechnologien diese funktionieren.

Die Beispiele sind nicht von bestimmten Programmen abhängig, sondern können mit einfachen Texteditoren nachvollzogen werden. Die verschiedenen Programme, um XML-Dokumente anzuzeigen, zu prüfen und umzuwandeln, werden in den jeweiligen Kapiteln vorgestellt. Technologieabhängig sind die Beispiele in Kapitel 4. Dort wird je ein Beispiel in PHP, in Java und in .NET gezeigt. Außerdem erfahren Sie, ob die jeweilige Programmiertechnik auch in den anderen Technologien verfügbar ist.

Unter *http://www.hauser-wenz.de/support/* stehen Ihnen die Beispiele zur Verfügung. Dort finden Sie außerdem Errata und Updates zum Buch. In unserem Weblog unter *http://www.hauser-wenz.de/blog/* gibt es regelmäßig neue Einträge zu den aktuellen Themen der Webentwicklung, darunter natürlich auch XML. Für Anregungen und Kritik zum Buch oder zur von Christian Wenz und mir mitgestalteten Buchreihe finden Sie unter *http://www.hauser-wenz.de/support/kontakt/* ein entsprechendes Formular. Selbstverständlich sind auch Fragen zum Buch willkommen und werden von mir möglichst zeitnah beantwortet. Aus Fairness gegenüber den zahlenden Kunden kann ich leider keine Fragen abseits des Buchinhalts kostenlos beantworten.

Obwohl es ein kurzes Werk ist, hatte ich auch bei diesem Buch Hilfe. Christian sei für seinen kleinen, aber sehr feinen inhaltlichen Beitrag gedankt, Sebastian Burkart für seine Unterstützung.

Viel Vergnügen bei der Lektüre!

Tobias Hauser
Im Mai 2010

Die Idee

XML ist nicht nur eine Technologie, sondern eine Idee. Als Format dient XML dazu, Daten zu speichern und zwischen sehr unterschiedlichen Systemen austauschbar zu machen. Außen herum entsteht ein ganzes Universum. Dieses Universum besteht aus Standards, die von XML abgeleitet sind, und aus Hilfstechnologien, die die Bearbeitung von XML-Dokumenten ermöglichen.

XML selbst ist eine Metasprache. Das heißt, mit XML lassen sich Untersprachen definieren, die ganz bestimmte Zwecke erfüllen. Dokumente in diesen Untersprachen sind immer noch XML-Dokumente, aber sie basieren nur noch auf dem eingeschränkten Befehlssprachschatz der Untersprache. Dieser Aspekt wird im Namen von XML deutlich. Das Akronym steht für **eXtensible Markup Language**. Ins Deutsche übersetzt ergeben sich zwei Bestandteile:

- „eXtensible“ steht für erweiterbar. Dies ist der Aspekt der Metasprache.
- „Markup Language“ steht für Beschreibungssprache. Die Sprache selbst beschreibt die Inhalte.

Bevor Sie mehr zur Standardisierung erfahren, werfen Sie im nächsten Abschnitt einen Blick auf ein erstes XML-Dokument.

1.1 Ein XML-Dokument

Grundlage des XML-Universums sind XML-Dokumente. Deswegen erfahren Sie hier zuerst ein paar grundlegende Informationen, bevor wir über andere Technologien schreiben. Mehr von XML sehen Sie dann in Kapitel 2.

Basis von XML sind Tags, „Befehle“ in spitzen Klammern:

```
<kontakt>
  <vorname>Tobias</vorname>
  <nachname>Hauser</nachname>
</kontakt>
```

In den Tags stehen wiederum andere Tags oder Text als Inhalt. Die Tags beschreiben normalerweise den Inhalt. Der Begriff „Tag“ ist zwar englisch, hat sich im Deutschen aber vor allem durch HTML als gängiger Ausdruck durchgesetzt. Ein Tag (oder auch Element) besteht aus verschiedenen Elementen:

- Das Start-Tag ist das öffnende Tag. Es kann beliebig viele Attribute enthalten. Ein Attribut besteht aus dem Attributnamen, hier beispielsweise `id`, und einem Wert, hier `1`, der immer in Anführungszeichen steht:

```
<kontakt id="1" art="privat">
  <vorname>Tobias</vorname>
  <nachname>Hauser</nachname>
</kontakt>
```

- Der Inhalt eines Tags kann wieder ein Elements ein. Hier ist das beim `<kontakt>`-Tag der Fall. Inhalt kann auch Text bzw. Inhalt sein, hier beispielsweise bei `<vorname>`.
- Das End-Tag schließt das Tag. Es beginnt immer mit einem führenden Schrägstrich. Hat ein Tag keinen Inhalt, kann es auch direkt im Start-Tag wieder geschlossen werden:

```
<kontakt id="1" art="privat">
  <vorname>Tobias</vorname>
  <nachname>Hauser</nachname>
  <telefon nummer="00000"/>
</kontakt>
```

HINWEIS: Neben Tags und Text kann ein XML-Dokument auch noch Kommentare enthalten. Diese Kommentare werden bei der Verarbeitung des Dokuments ignoriert oder als Kommentare ausgewiesen. Sie gleichen den Kommentaren für HTML und beginnen mit `<!--` und enden mit `-->`.

Um ein XML-Dokument als solches zu erkennen, gibt es das XML-Tag zu Beginn des Dokuments. Dieses Tag heißt auch „Processing Instruction“ oder „XML-Deklaration“.

```
<?xml version="1.0" ?>
<!-- XML-Inhalte -->
```

PROFITIPP: Processing Instructions können nicht nur für das XML-Dokument selbst angegeben werden, sondern auch mitten im XML-Dokument stehen und von anderen Technologien genutzt werden. PHP verwendet beispielsweise auch Processing Instructions mit `<?php ?>`.

Der Name „Processing Instruction“ stammt daher, dass dieses Tag für Programme, die das XML-Dokument abarbeiten, entsprechende Informationen zur Verfügung stellt. Zu diesen Informationen gehören:

- Die XML-Versionsnumme: Hier stehen aktuell 1.0 und 1.1 zur Wahl (siehe Abschnitt „Ursprung und Standards“):

```
<?xml version="1.0" ?>
```

- Der für das XML-Dokument verwendete Zeichensatz: Der Zeichensatz gibt an, welche Sonderzeichen und sonstigen Schriftzeichen eingesetzt werden. Die bekannteste Kodierung ist UTF-8. Sie ist eine Variante des Unicode-Zeichensatzes (*http://www.unicode.org/*) und wird auch bei der Internet Engineering Task Force standardisiert (*http://www.ietf.org/rfc/rfc3629.txt*). Bei UTF-8 werden beliebige Zeichen mit speziellen Bytekombinationen aus vier Bytes gebildet. Eine Alternative ist ISO-8859-1 (auch Latin-1): Dieser Zeichensatz kodiert nur 256 Zeichen, enthält aber die wichtigsten:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

- Die Angabe, ob ein XML-Dokument alleine, sprich ohne Strukturinformationen, steht oder nicht:

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes" ?>
```

HINWEIS: Einzig das Attribut für die Version ist Pflicht. Kodierung und `standalone` können weggelassen werden. Sind sie aber vorhanden, so ist die Reihenfolge `version`, `encoding` und dann `standalone` vorgeschrieben.

Wenn Sie ein XML-Dokument fertiggestellt haben, bilden alle Tags gemeinsam eine hierarchische Dokumentenstruktur. Essenziell ist dabei, dass es nur ein Wurzelelement gibt. Als Wurzelelement wird das oberste Tag der Hierarchie bezeichnet:

```
<?xml version="1.0" encoding="UTF-8" ?>
<kontakte>
  <kontakt id="1" art="privat">
    <vorname>Tobias</vorname>
    <nachname>Hauser</nachname>
    <telefon nummer="00000" />
  </kontakt>
  <kontakt id="2" art="geschäftlich">
    <vorname>Christian</vorname>
    <nachname>Wenz</nachname>
    <telefon nummer="11111"/>
  </kontakt>
</kontakte>
```

Listing 1.1: Eine einfache XML-Datei (*kontakt.xml*)

Im obigen Fall ist das Tag `<kontakte>` das Wurzelelement. Betrachtet man das XML-Dokument als hierarchisches Dokument, so sind die einzelnen Tags auch Knoten. Die Inhalte innerhalb der Tags sind ebenfalls Knoten, so genannte Textknoten. Die folgende Abbildung illustriert den hierarchischen Aufbau.