

XML – Merkblatt 3 – W3C Schema

Einführung XML Schema (XML schema declaration, XSD) ist ein XML-Dialekt, mit dem gültige XML-Dokumente beschrieben werden können. Im Gegensatz zu DTDs gibt es in XML-Schema eine Fülle zusätzlicher Datentypen wie Zahlen. Der Standard unterstützt Namensräume. XML-Schema ist der bevorzugte Standard, um Syntax und Datentypen von XML-Dokumente zu definieren. Er ist sehr umfangreich, so dass hier nur eine kurze Einführung gegeben werden kann.

XML Schema Beispiel Das Wurzelement eines XML Schemas ist `schema`. Als Namensraum ist `http://www.w3.org/2001/XMLSchema` festgelegt.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="mein.namensraum"
           targetNamespace="mein.namensraum">
  <xs:element name="sport-events">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="city" type="xs:string" />
        <xs:element name="event" type="event-type" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="xs:decimal"
                    use="optional" default="1.0"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="event-type">
    <xs:all>
      <xs:element name="name" type="xs:normalizedString"/>
      <xs:element name="start" type="xs:date"/>
      <xs:element name="code" type="xs:int" minOccurs="0"/>
    </xs:all>
  </xs:complexType>
</xs:schema>
```

Mit dem optionalen Attribut `targetNamespace` wird der Namensraum für die zu definierenden Elemente und Attribute definiert.

Innerhalb von `schema` sind (globale) Elementdefinitionen mit `element` und Typdefinitionen mit `complexType` (für komplexen Inhalt mit inneren Elementen und Attributen) oder `simpleType` (für nicht komplexen Inhalt) erlaubt. Das Wurzelement ist immer als globales Element deklariert.

`element` hat unter anderem die folgende Attribute

- ein verpflichtendes `name` für den Elementnamen,

- ein optionales **type** für den Inhalt und
- zwei optionale, welche die Mindest- (**minOccurs**, Standardwert 1) und Höchstanzahl (**maxOccurs**, Standardwert 1) dieses Element definiert. **unbounded** definiert eine unbegrenzte Anzahl Vorkommen.

Das Inhaltsmodell eines Elements muss entweder als inneres Element mit **complexType** (anonymer Typ) oder über das Attribut **type** mit einem Namen angegeben werden. Dieser Name wird mit **complexType** innerhalb des Wurzelements definiert. Analog bei einfachen Typen mit **simpleType**.

Innerhalb von **complexType** sind folgende Elemente erlaubt:

- **sequence** (alle darin definierten Elemente müssen im XML-Dokument hintereinander folgen),
- **all** (die Reihenfolge der definierten Element ist beliebig) oder
- **choice** (genau eines der definierten Element ist erlaubt).

Innerhalb dieser Elemente kann wieder **element** verwendet werden. Attribute lassen sich mit beliebig vielen **attribute**-Elementen vor dem schließenden **complexType** definieren.

Atomare Typen Der XML-Schema-Standard besitzt eine Fülle von vordefinierten *atomaren* Typen, wie **string**. Da diese Werte dem XML-Schema Namensraum zugeordnet sind, ist in dem Beispiel der entsprechende Präfix **xs:** den Typen vorangestellt.

Typ	Wertebereich	Beispiel
string	alle Zeichenketten	
normalizedString	Zeichenkette ohne Leerzeichen am Anfang oder Ende	
byte, short, int, long	8-, 16-, 32-, 64-Bit 2er-Komplement	10 -55
unsignedByte, ..., unsignedLong	8, ..., 64-Bit Dualzahl	10 55
decimal	Dezimalzahl mit Vor- und Nachkommaanteil	17.50 -65.00
float,	IEEE 754 single,	0.123 123E-3 NaN
double	double precision	-5498.231 INF -INF
date	ISO Zeitcode	2017-08-28
time	ISO Uhrzeitcode	17:55:10
	mit Zeitzone	17:55:10+01:00
dateTime	ISO-8601	2017-08-28t17:55:10-05:00
boolean		true false

Attribute Attributenamen werden mit **name** und deren Wertemenge mit einem nicht komplexen **type** definiert. Mit dem optionalen Attribut **use** (**required**, **fixed**, Standardwert **optional**) wird das Vorkommen analog zu DTDs definiert. Der Standardwert des Attributs kann mit **default** angegeben werden. Das **attribute**-Element kennt noch viele synonyme Angaben der Attributsemantik.

Einfache Typen Einfache Typen (simple types) lassen sich aus anderen einfachen oder atomaren Typen (außer **boolean**) durch eine Einschränkung des Wertebereich ableiten. Folgender einfacher Typen **customer-id** ist eine Zeichenkette, die aus genau 10 Zeichen (inklusive Leerzeichen) besteht *und* zum mit **pattern** angegebenen regulären Ausdruck passt.

```
<xs:simpleType name="customer-id2">
  <xs:restriction base="xs:string">
    <xs:length value="10"/>
    <xs:pattern value="[A-Z][0-9]+"/>
  </xs:restriction>
</xs:simpleType>
```

Der folgende Typ **country-code** hat als Wertebereich die drei Zeichenketten DE, UK und FR.

```
<xs:simpleType name="country-code">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DE"/>
    <xs:enumeration value="UK"/>
    <xs:enumeration value="FR"/>
  </xs:restriction>
</xs:simpleType>
```

Elemente mit Attributen bestehen immer aus komplexen Inhalt. Ein Element mit einfachen Typ und einem Attribut kann wie folgt definiert werden:

```
<xs:element name="internationalPrice">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="currency" type="xs:string"/>
        <xs:attribute name="value" type="xs:decimal"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Verweis auf XML-Schema Um auf das zugehörige XML-Schema zu verweisen, kann das Attribut `schemaLocation` oder `noNamespaceSchemaLocation` (kein `targetNamespace` im XML-Schema definiert) verwendet werden. Diese beiden Attribute sind dem Namensraum `http://www.w3.org/2001/XMLSchema-instance` zugeordnet.

```
<mein-element-irgendwo-im-dokument
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation
    ="mein.namensraum example.xsd dein.namensraum beispiel.xsd">
</mein-element-irgendwo-im-dokument>
```

Mit `schemaLocation` kann auf mehr als ein XML-Schema verwiesen werden, um verschiedene XML-Dialekte zu mischen. Als Wert muss eine Folge von Paaren mit Namensraum und URI des XML-Schemas angegeben werden. Damit ein XML-Prozessor im XML-Dokument die lokalen Elemente richtig zuordnen kann, müssen die Namensräume korrekt deklariert und verwendet werden.

```
<messages xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="mein.namensraum" xmlns:dein="dein.namensraum"
  xsi:schemaLocation
    ="mein.namensraum example.xsd dein.namensraum beispiel.xsd">
  <text>Hello<dein:text>Hi</dein:text></text>
```

`text` ist als innere Element von `messages` in `example.xsd` deklariert. Um den XML-Dialekt von `beispiel.xsd` darin syntaktisch verwenden zu können, ist der Typ von `text` als `anyType` deklariert. `text` in `beispiel.xsd` ist ein dort global definiertes Element.

Ergänzende Literatur In [1, Abschnitt 3.3.2] gibt es eine Übersicht über den XML-Schema-Standard. Für Details sei hier auf die Einführung zur XML Schema-Spezifikation [2] verwiesen. Für die korrekte Verwendung von Namensräumen unbedingt noch [3, Abschnitt 10.3] lesen!

Literatur

- [1] Serge Abiteboul, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset, and Pierre Senellart. *Web Data Management*. Cambridge University Press, New York, NY, USA, 2011. <http://webdam.inria.fr/Jorge/files/wdm.pdf>.
- [2] David C. Fallside and Priscilla Walmsley. XML Schema Part 0: Primer Second Edition. Technical report, W3C, December 2008. <https://www.w3.org/TR/xmlschema-0/>.
- [3] Eric van der Vlist. *XML Schema*. O'Reilly Media, Inc., 1 edition, 2002.