



Fachhochschule Köln
Cologne University of Applied Sciences

Institut für Informatik

Webbasierte Anwendungen 2
Prof. Dr. Kristian Fischer

Dokumentation

Phase 1

Abgabe: 12.04.2013

Matthias Rößler
Matrikelnummer: 11083608
Medieninformatik Bachelor
4. Semester

Aufgabe 1

Erklären Sie kurz die Begriffe Wohlgeformtheit, Validität und Namespace im Bezug auf XML und XML-Schema.

Wohlgeformtheit

Ein „Well Formed“ XML Dokument hat eine korrekte Syntax.

Einige Regeln zur korrekten Syntax:

- ein XML Dokument muss und darf nur ein Root-Element (Wurzelknoten) besitzen
- XML Elemente mit Inhalt müssen ein Schließen-Tag haben
- XML Tags sind case sensitive d. h. es wird zwischen Groß- und Kleinschreibung unterschieden
- XML Dokumente müssen Wohlgeschachtelt (properly nested) sein. D. h. bspw., dass das zuletzt öffnende Tag als erstes wieder geschlossen werden muss.
- Werte von Attributen müssen in Anführungszeichen stehen (quoted).

Validität

Ein „valides“ XML Dokument ist „wohlgeformt“ und hält zusätzlich die Regeln einer XML-DTD oder eines XML Schemas ein.

Namespace

In XML Dokumenten werden die Elementnamen von Entwicklern vergeben. Dies kann zu Konflikten führen wenn aus verschiedenen XML Dokumenten Elemente mit gleichem Namen genutzt werden.

Um diese Konflikte zu verhindern können verschiedene Prefixe vor den Elementnamen genutzt werden. Diesen muss ein Namespace zugeordnet werden, was entweder im Root-Element oder dem ersten Eltern-Element mit jeweiligem Prefix geschehen muss.

Aufgabe 2

a)

Erzeugen Sie ein XML-Dokument, dass die Daten des folgenden Formulars vollständig erfasst:

<http://www.gm.fh-koeln.de/~vsch/anmeldung/gruppenanmeldung.html>

Füllen Sie das Dokument mit einem Beispieldatensatz. Achten Sie darauf, dass über das Formular mehrere Personen gleichzeitig erfasst werden können.

Wichtig: Es sollte nicht die HTML-Struktur der Webseite in der XML-Datei abgebildet werden, sondern die zu übertragenden Daten.

XML-Dokument: aufgabe2.xml

Das XML-Dokument hat das Root-Element „Formular“. Formular kann mehrere Elemente „Person“ enthalten, wobei über das Attribut „Gruppenleiter“ (1 ja, 0 nein) festgelegt wird ob es sich bei der Person um den Gruppenleiter handelt. Die Elemente „Person“ enthält jeweils die Elemente „Vorname“, „Nachname“, „Email“, „GDatum“, „Erfahrung“, „Schlagzeug“ und sofern es sich um den Gruppenleiter handelt das Element „Anmerkung“.

b)

JSON-Dokument: aufgabe2.json

Während im XML-Dokument aus Teilaufgabe a mehrere Personen durch mehrfachnutzung des Elements „Person“ abgebildet werden konnten, muss bei der Object Notation ein Array genutzt werden. Die drei Datensätze (Objekte) werden hier also in einem Array abgebildet.

Aufgabe 3

a)

Gegeben ist folgendes Rezept:

<http://www.chefkoch.de/rezepte/24641006006067/Lenchen-s-Schokoladenkuchen.html>

Entwickeln Sie ein XML-Dokument, in dem die Daten des Rezeptes abgebildet werden. Achten Sie darauf, dass das Dokument semantisch möglichst reichhaltig ist. Bei dieser und den folgenden Aufgaben lassen sie bitte die Daten in der Marginalspalte auf der rechten Seite weg.

XML-Dokument: aufgabe3.xml

Das Root-Element lautet „Chefkoch“, es enthält ein oder mehrere Elemente „Rezept“ welche jeweils mit dem Attribut „id“ einen eindeutigen identifier besitzt. Das Rezept enthält eine Überschrift und ein Bild. Diese beiden Informationen wurden in ein „info“-Element verschachtelt. Das Element „Zutatenliste“ enthält ein oder mehrere Elemente „Zutat“ welches wiederum jeweils ein Element „Menge“, „Einheit“ und „Bezeichnung“ besitzt. Nach dem Element „Zutatenliste“ folgt das Element „Zubereitung“ welches die Elemente „Arbeitszeit“, „Schwierigkeitsgrad“, „Brennwert“ und „Anleitung“ enthält.

b)

Betrachten Sie nun andere Rezepte auf der Webseite <http://www.chefkoch.de>. Beschreiben Sie welche Gemeinsamkeiten die Rezepte hinsichtlich ihrer Daten haben und worin Sie sich unterscheiden.

Gemeinsamkeiten

Informationen die alle Rezepte enthalten sind Überschrift, URL zum Bild, eine Zutatenliste und eine schriftliche Anleitung.

Unterscheidungen

Es gibt zwischen einzelnen Rezepten einige Unterschiede hinsichtlich ihrer Daten, so kann bei einem Rezept die Zutatenliste viele Elemente enthalten bei anderen weniger, oft sind Kommentare von Usern vorhanden manchmal aber auch nicht. Bei einigen Rezepten können weitere Daten wie „Backzeit“ oder „Ruhezeit“ auftauchen.

c)

Arbeiten Sie die Kriterien heraus, die für die Entwicklung einer XML-Schema-Datei beachtet werden müssen. Die Schema-Datei soll die Struktur für eine XML-Datei definieren, in der mehrere unterschiedliche Rezepte gespeichert werden können.

Ziel ist es, dass das XML-Schema möglichst restriktiv ist, so dass in der XML-Datei möglichst semantisch sinnvolle Daten bezüglich der Rezepte gespeichert werden können. Ziehen Sie beim Aufstellen der Kriterien u.A. folgende Fragestellungen in Betracht:

- Welche Daten müssen in simple und welche in complex-types abgebildet werden?
- Für welche Daten ist die Abbildung in Attributen sinnvoller?
- Welche Datentypen müssen für die Elemente definiert werden?
- Welche Restriktionen müssen definiert werden?

simple-/complex-types

Die Elemente „Chefkoch“, „Rezept“, „info“, „Zutatenliste“, „Zutat“, „Zubereitung“ und „Kommentare“ müssen complex-type sein, da sie jeweils weitere Elemente enthalten.

Alle anderen Elemente sind simple-type.

Attribute

Für das Element „Rezept“ ist ein Attribut „id“ vorgesehen, damit jedes Rezept eindeutig identifiziert werden kann.

Datentypen

Für das Bild-Element ist eine „anyURI“ vorgesehen. Ich habe mich für die Angabe von Referenzen zu den Bildern entschieden, da so die Datenmenge im XML gering gehalten werden kann. Außerdem soll der Inhalt in einer späteren Aufgabe über die Konsole wiedergegeben werden. Der Type „base64“ wäre hier nicht gut lesbar. Sofern Nummernwerte in den Elementen angegeben werden sollen, ist der Type jeweils „integer“ oder „dezimal“, je nach dem ob ganze Zahlen oder Dezimalzahlen erwartet werden. In allen anderen Fällen ist der Type „string“. Enthält ein Element eine Restriktion wird der Type nicht über das Attribut „type“ sondern in der Restriktion mit „base“ angegeben.

Restriktionen

- Menge hat die Restriktion nicht kleiner als 0 sein zu dürfen
- Für die Einheiten ist eine vordefinierte Liste an Werten erlaubt. Dies verhindert, dass für eine oder ähnliche Einheit verschiedene Bezeichnungen vorhanden sind (bspw. Killogramm, kg, Gramm. usw.).
- Die Arbeitszeit darf nicht kleiner als 0 sein.
- Auch für den Schwierigkeitsgrad gibt es eine vordefinierte Liste. Auch hier sollen zu ähnliche Angaben wie „mittel“ und „normal“ vermieden werden.
- Brennwert darf nicht kleiner als 0 sein.

Aufgabe 4

Java-File: /src/com/w3schools/Main.java

Aufgabe 5

Bei JSON ist die XML-typische Trennung von Daten und Code nicht zwingend. JSON ist ausführbarer JavaScript-Code. Dank der einfachen Schreibweise ist JSON für den Mensch leichter verständlich als XML. JSON enthält wesentlich weniger Overhead als XML, da bei XML viele Tags und Attribute nötig sind. XML ist eine Auszeichnungssprache und damit vielseitiger als JSON. Vorteil von XML ist, dass Elemente und Attribute durch die Verwendung von Namespaces eindeutig benannt werden. Durch den Namespace erhalten XML-Dokumente einen Kontext in dem sie verwendet werden.