



Fachhochschule Köln
Cologne University of Applied Sciences

WBA 2

Dokumentation Phase 1

Author
Oliver van der Bürie

Aufgabe 1

Was bedeutet Wohlgeformt?

Wohlgeformt bedeutet, dass eine Datei die Regeln von XML korrekt einhält.

- Elemente müssen eine geschlossene Hierarchie haben. `<p></p>` ist nicht erlaubt.
- Elemente müssen immer geschlossen werden. `<p></p>`
- Elemente sind Case sensitive. `<p></P>` führt zu einem Fehler.
- Leere Elemente müssen geschlossen werden. `
`
- Attributwerte müssen in Anführungszeichen stehen. `<p c=1></p>` = Falsch, `<p c="1"></p>` = Richtig
- Es gibt keine booleschen "Schalter". `<option value="gm" checked>foo</option>` ist nicht möglich. Wenn dann `checked="checked"`.

Was bedeutet validiert?

Validität bedeutet, dass eine XML Datei durch ein XML-Schema oder eine DTD semantisch validiert und auf Gültigkeit geprüft wurde.

Beispiel: Einbinden eines Schemas in eine XML Datei

```
<rezepte xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="Aufgabe3c.xsd">
```

Was bedeutet Namespace?

Namespaces bedeutet, dass die entsprechenden Tags zu einem gewissen Namespace gehören und somit eindeutig zuzuordnen sind bzw. einem gewissen Vokabular unterliegen.

`xmlns:draft="http://www.w3.org/2001/someDraft"` wäre die Deklaration des Namespace "draft" mit Verweis auf den entsprechenden URI.

Aufgabe 2

Bei Aufgabe zwei konnte ich insgesamt für jede Person sieben Felder identifizieren. Für jede Person sind klar ersichtlich der Vorname , der Nachname und die E-Mail Adresse. Weiterhin kann jede Person das Geburtsdatum eingeben. Hierbei habe ich mich dafür entschieden das Format als Attribut aus dem Grund mit zu übertragen, da sie doch sehr viele unterschiedliche Zeitformate gibt und es somit zu keiner Verwechslung kommen kann. Bei der Erfahrung habe ich mich für einen Integer als Attribut entschieden. Somit muß ich das Element nicht mehr explizit schließen und erspare mir somit einen gewissen Overhead. Das gleiche gilt für die boolsche Variabel der Ausrüstung. Ich habe mich weiterhin dafür entschieden keinen gesonderten Tag „Gruppenleiter“ einzuführen, sondern dieses über die Gruppen ID zu identifizieren. Zuletzt wird noch einen Kommentar zu allen Personen der Gruppe hinzugefügt.

Aufgabe 3b

Jedes Dokument muss haben:

- Zutaten mit:
 - Menge
 - Einheit
 - Name
- Dauer
- Rezeptname
- Schwierigkeit
- Zubereitungstext
- Brennwert

Ein Dokument kann haben:

- ein oder mehrere Fotos
- Links innerhalb der Zutaten
- Social Network Tags
- Ein oder mehrere Kommentare

Die Kommentare enthalten:

- Nutzername
- Datum
- Nutzerbild
- Nutzertag
- Bewertung
- Kommentar

Beschreibung:

Da ich persönlich kein Mitglied von Chefkoch.de bin, kann ich leider auch keine definitive Aussage über Pflichtfelder treffen. Deshalb habe ich diese nach meiner persönlichen Entscheidung unter "muss" aufgeführt. Für ein Rezept sind zwingend die Zutaten und eine Beschreibung erforderlich, wie diese zubereitet werden. Dieses sollte keiner näheren Begründung bedürfen. Die Dauer ist ein Wert, den nur der Ersteller dieses Rezepts kennt, der aber gerade für die Auswahl eines Rezepts von Bedeutung sein kann. Die Schwierigkeit obliegt ebenfalls dem subjektiven Empfinden des Erstellers und dient des weiteren als Auswahlkriterium. Der Brennwert kann ein Streitpunkt sein, denn einem Normalsterblichen Koch kann eigentlich nicht zugemutet werden, den Brennwert seines Rezepts zu ermitteln. Ich schwanke hierbei persönlich zwischen kann und muss Lösung.

Unter den "kann" Daten befinden sich an erster Stelle die Rezept Fotos. Diese müssen nicht zwingend für ein Rezept vorhanden sein um es kochen zu können. Ebenfalls müssen nicht zwingend Links auf die Zutatenbeschreibungen vorhanden sein. Die SNG Tags habe ich hier ebenfalls mit aufgeführt, da bei einem "Teilen" oder "Liken" eine eindeutige ID für dieses Rezept mit übergeben wird. Diese wird jedoch nicht durch den Nutzer angelegt, sondern meistens durch eine Routine. Somit könnte dieser Punkt auch unter den "muss" Daten stehen. Da die ID allerdings erst nachträglich auf dem Server hinzugefügt wird, kann bei der Übertragung vom Client dieses Feld auch leer sein bzw. es wird leer sein. Das ein Rezept nach dem Erstellen noch keinen Kommentar aufweist sollte sich ebenfalls ohne nähere Begründung erschließen lassen.

Die Anzahl der Personen habe ich bewusst raus gelassen, da diese bei Chefkoch.de durch ein JavaScript aufgerechnet werden. Dies war jedoch auch ein strittiger Punkt, ob die Anzahl zuerst zum Server übertragen wird und dann dort oder ob das Rezept Clientseitig durch ein JavaScript umgerechnet wird. Ich habe mich dafür entschieden dass das der Client macht. Wenn wir uns eine Webseite wie die von Chefkoch.de genauer betrachten, dann können wir feststellen dass diese nicht nur in Deutschland existiert. Weiterhin gehört Chefkoch.de zu den am meisten frequentierten Koch Seiten im Internet. Wenn wir jetzt davon ausgehen das tagtäglich mehrere zehntausend Nutzer diese Seite besuchen und für jeden Besucher der Server ein Rezept umrechnen muß, dann ist das, bei durchschnittlich fünf Zutaten pro Rezept, ein größerer Rechenaufwand der sich wieder in Kosten für die Betreiber niederschlägt.

Ich habe bei meiner Struktur absichtlich die Kommentare von den eigentlichen Rezepten getrennt. Wenn man von verteilten Systemen ausgeht, sollen die Daten den unterschiedlichsten Plattformen mit den unterschiedlichsten Diensten / Anwendungen zur Verfügung stehen. Hierbei stellt sich die Frage, wozu man bei einem Rezept zwingend Kommentare braucht respektive ob sich die Semantik der Daten durch einen Kommentar ändert. Ebenfalls gilt zu prüfen, in wie fern Kommentare bei einem Dienst / einer App angezeigt werden müssen. Ein Beispiel: Eine Website, die auf die Rezeptdaten von Chefkoch.de zurückgreift um die Zutaten einer Einkaufsliste hinzuzufügen, benötigt keine Kommentare.

Aufgabe 5

Warum werden Daten in Formaten wie JSON oder XML gespeichert?

Durch das Speichern der Daten in diesen Formaten können die Daten semantisch geordnet, strukturiert und baumartig sortiert werden.

Dieses wäre mit Daten in loser Form oder als "normale" Textdateien nicht möglich.

Stellen Sie die Formate XML und JSON gegenüber.

XML

XML bietet eine menschenlesbare Struktur für Daten. Diese Struktur kann beliebig, jedoch nach den gegebenen Regeln für XML Dokumente geschachtelt werden. Weiterhin können Metainformationen zu Daten oder Typisierung von Daten in Form von Attributen vergeben werden. Ebenfalls ist die Validierung mittels XML Schemata oder DTD möglich. Der Nachteil von XML ist der teilweise erhebliche Overload an strukturbedingten Auszeichnungen.

JSON

JSON bietet eine baumartige Strukturierung von Daten und kennt nur einfache Datentypen. Daten können hierbei ebenfalls semantisch geschachtelt und geordnet werden. Um ein JSON String zu validieren wird jedoch eine Server.- oder Clientseitige Funktionalität benötigt, die dieses beim Parsen des Strings übernimmt. JSON wird von JavaScript nativ erkannt und eingebunden.

Gegenüberstellung

Um komplexe, typisierte Daten, die validiert werden müssen, zu übermitteln oder zu speichern eignet sich XML hervorragend. Zumal XML auch wesentlich menschenlesbarer ist als JSON. Zur Übermittlung von größeren Datenmengen (bsp.: WebGL-3d-Vektoren) eignet sich JSON wesentlich mehr, da hierbei der Overhead von XML wegfällt.