

XML und JSON, eine Gegenüberstellung

Emanuel Pörnbacher

10. Juni 2015

Inhaltsverzeichnis

1 Einleitung	1
2 Die Formate	2
2.1 Json	2
2.2 Xml	3
3 Gegenüberstellung	3
3.1 Lesbarkeit	3
3.2 Verwendung	3
3.3 Weltoffenheit	4
3.4 Web Unterstützung	4
4 Fazit	4

1 Einleitung

Für Web APIs und Konfigurationen wird XML zunehmend von einfacheren Formaten wie JSON verdrängt. Dieser Artikel vergleicht JSON und XML und zeigt die Stärken der Formate für bestimmte Anwendungsgebiete auf.

2 Die Formate

2.1 Json

JSON (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben und für Maschinen einfach zu parsen (Analysieren von Datenstrukturen) und zu generieren ist. Bei JSON handelt es sich um ein Textformat, das komplett unabhängig von Programmiersprachen ist, aber vielen Konventionen folgt, die Programmieren aus der Familie der C-basierten Sprachen (C, C++, C-Sharp, Java, JavaScript, Perl, Python uvm.) bekannt sind. Diese Eigenschaften machen JSON zum idealen Format für Datenaustausch.

JSON baut auf zwei Strukturen auf:

- **Name/Wert Paare**

In verschiedenen Sprachen wird dies realisiert als ein Objekt (object), Satz (record), Struktur (struct), Wörterbuch bzw. Verzeichnis (dictionary), Hash-Tabelle (hash table), Schlüssel-Liste (keyed list) oder als ein assoziatives Array (associative array).

- **Eine geordnete Liste von Werten**

In den meisten Sprachen wird das als Array (array), Vektor (vector), Liste (list) oder Sequenz (sequence) realisiert.

Hierbei handelt es sich um universelle Datenstrukturen, die von so gut wie allen modernen Programmiersprachen in der einen oder anderen Form unterstützt werden. Es macht somit Sinn, dass ein zwischen Programmiersprachen austauschbares Datenformat auch auf diesen Strukturen aufbaut.

Abbildung 1 zeigt ein Beispiel einer Zusammensetzung eines Cocktails im JSON Format.

Listing 1: Json Example Code

```
1 {"Cocktail": {
2     "id": "2",
3     "description": "Cuba Libre",
4     "price": "4.50",
5     "ingredient": [
6         "Rum",
7         "Cola",
8         "Lime",
9         "Sugar"
10    ]
11 }}
```

2.2 Xml

XML (Extensible Markup Language) ist ein text-basiertes Format für den Austausch strukturierter Information. XML selbst stellt keine Sprache zur Definition von Inhalten dar, sondern lediglich die Grundlage für die Definition einer solchen Sprache. Daher ist bei XML auch kein einziger Element-Typ (Tag) von der Bedeutung her definiert. XML definiert lediglich den Aufbau derartiger Dokumente. Es gibt tausende Markup Sprachen, die auf XML basieren.

Abbildung 2 zeigt den selben Cocktail wie in Abbildung 1 im XML Format.

Listing 2: XML Example Code

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <Cocktail>
3 <id>2</id>
4 <description>"Cuba Libre"</description>
5 <price>4.50</price>
6 <ingredient>"Rum"</ingredient>
7 <ingredient>"Cola"</ingredient>
8 <ingredient>"Lime"</ingredient>
9 <ingredient>"Sugar"</ingredient>
10 </Cocktail>
```

3 Gegenüberstellung

3.1 Lesbarkeit

Der Cocktail im XML Format hat einen größeren Overhead und ist unübersichtlicher als das JSON Dokument. Würden wir unseren Vergleich dabei belassen, so wäre XML der klare Verlierer. Würde man jedoch JSON um Namensräume¹ erweitern, würden es wahrscheinlich genauso unübersichtlich werden wie ein XML Dokument.

3.2 Verwendung

XML ist im Gegensatz zu JSON eine Markupsprache² und kann daher für mehr als nur für das Serialisieren von Objekten eingesetzt werden.

¹(engl. name spaces) sind in XML-Dokumenten ein einfaches Verfahren zur eindeutigen Benennung von Element- und Attributnamen. Dabei werden die Element- und Attributnamen so verknüpft, dass diese durch URI-Referenzen eindeutig identifiziert werden können.

²Auszeichnungssprachen (ML), auch als Beschreibungssprachen bezeichnet, definieren die Bausteine eines Dokuments und legen die Beziehung fest, in denen sie zueinander stehen. Sie sind textbasiert und beschreiben je nach Anwendung den logischen Inhalt, Struktur, Datenaustausch oder werden zur Definition anderer ML benutzt.

Für die Notation von JSON wurde die Syntax von JavaScript verwendet. Somit ist ein JSON Dokument gleichzeitig ein ausführbares JavaScript. Die übliche Trennung von Daten (z.B. XML Dokumente) und Code (z.B. Java) wird damit vermieden. Besonders beliebt ist der Einsatz von JSON für das asynchrone Übertragen von Daten zwischen Servern und Web Seiten. Daher hat es XML als AJAX Format mittlerweile fast verdrängt.

3.3 Weltoffenheit

Durch die Verwendung von Namespaces können Elemente und Attribute in XML weltweit eindeutig benannt werden. Ein XML Dokument kann losgelöst von einer Anwendung interpretiert werden. Bei JSON Dokumenten bekommt der Inhalt erst durch die Anwendung, die das Dokument einliest, einen Sinn. Durch Die Namespaces bekommen XML Dokumente einen Kontext, in dessen Rahmen sie interpretiert werden können.

3.4 Web Unterstützung

Mit XML lässt sich die REST Idee, dass alles eine Resource ist, leicht umsetzen. Über ein XLink Attribute oder ein umschließendes Element kann jeder Information ein Verweis auf eine Resource hinzugefügt werden. Mit JSON geht das nicht, obwohl diese vorgibt eine "Web" Sprache zu sein.

4 Fazit

Die Alleinherrschaft von XML wird langsam aber sicher durch JSON in Frage gestellt. JSON hat gezeigt, dass es für spezielle Aufgaben wie z. B. die Serialisierung von Objekten eine schlankere Alternative gibt und, dass es in Sachen Lesbarkeit auch um einiges besser geht. XML punktet nach wie vor durch seine Vielseitigkeit, Toolunterstützung sowie durch die Vielzahl der Standards, die auf XML aufbauen oder mit XML kombinierbar sind.

Literatur

- [1] <http://json.org/json-de.html> (26.05.2015)
- [2] <http://wiki.selfhtml.org/index.php?title=JavaScript/JSON&oldid=24163>
- [3] http://de.wikipedia.org/w/index.php?title=JavaScript_Object_Notation&oldid=141595587
- [4] <http://wiki.selfhtml.org/index.php?title=XML&oldid=29953>
- [5] http://de.wikipedia.org/w/index.php?title=Extensible_Markup_Language&oldid=140760376