

Programmierpraktikum 2012

Austauschbare Datenformate und Parsing

**Gruppe 45
Philipp Radermacher**

XML (**E**xtensible **M**arkup **L**anguage)

- ist ein Standard des World Wide Web Consortium
- XML-Dokumente sind sowohl maschinen- als auch menschenlesbar
- die XML definiert den Aufbau dieser Dokumente
- anwendungsunabhängiges Datenformat erlaubt den Datenaustausch zwischen Anwendungen

XML (**E**xtensible **M**arkup **L**anguage)

Aufbau

- XML-Dokumente bestehen aus Elementen und aus textuellen Inhalten
- ein Dokument besitzt genau ein Wurzelement (=äußerste Element)
- Elemente mit Inhalt besitzen einen Beginn- und einen End-Auszeichner
z. B. <element>Element1</element>
- Attribute können Elementen zugeordnet werden
z. B. <element attr1=„wert1“ attr2=„wert2“>
- Namen dürfen frei gewählt werden, müssen sich aber an Regeln halten
(alphan. Zeichen Unicode, Sonderzeichen _-., Trennzeichen :,
keine Ziffern oder - am Wortanfang)
- Ein XML-Dokument heißt wohlgeformt, wenn es alle XML-Regeln einhält.
- Zusätzlich: Verarbeitungsanweisungen an den Parser durch <?Anweis?>
z. B. <?xml version="1.0">

XML (**E**xtensible **M**arkup **L**anguage)

Beispiel

<Finale>

Verein="FC Bayern München"

Begegnung= „Champions League Finale 2012“

Gegner="FC Chelsea"

Entfernung_zum_Tor_in_Metern="11"

<Schuetze

<Name>

<Name>Arjen**</Name>**

<Name>Robben**</Name>**

</Name>

Trikotnummer="10"

</Schuetze>

Verwandelt="false"

Auswirkung="Das Land Bayern legalisiert die Robbenjagd"

</Finale>

JSON (JavaScript Object Notation)

- kompaktes Datenformat zum Datenaustausch zwischen Anwendungen
- Datenformat ist für den Menschen lesbar
- besonders häufige Verwendung im Internet / durch Browser, das JSON-Strings direkt in JavaScript-Objekte konvertiert werden können.
- unterstützt die Datentypen Zahlen, Wahrheitswerte (boolean), Strings, Arrays, null und Objekte
- können durch `eval()` interpretiert werden

JSON (JavaScript Object Notation)

Beispiel

```
{  
  „Verein“: „FC Bayern München“,  
  „Begegnung“: „Champions League Finale 2012“,  
  „Gegner“: „FC Chelsea“,  
  „Entfernung_zum_Tor_in_Metern“: 11,  
  „Schuetze“: {  
    „Name“: [ „Arjen“, „Robben“ ],  
    „Trikotnummer“: 10,  
  },  
  „Verwandelt“: false,  
  „Auswirkung“: „Das Land Bayern legalisiert die  
  Robbenjagd“,  
}
```

SAX-Parser

- Simple API for XML
- Ereignisorientiertes Parsing \Rightarrow liest das Dokument sequentiell
- Meldet Ereignisse während des Parsens anhand bestimmter Bedingung (Start eines Elements, ...) an das Anwendungsprogramm mittels Callbacks.
- Im Normalfall wird kein (vollständiger) Baum im Speicher aufgebaut!
 - \Rightarrow ereignisbasierte API sind einfacher als baumbasierte, liegen aber auf einer tieferen Ebene!

DOM-Parser

- Document Object Model-Parser
- die zentrale Datenstruktur ist ein Baum
(Knoten der Klasse Node bzw. Unterklassen davon)
 - gesamtes Dokument als Baum im Speicher
 - vorteilhaft für Anwendungen, die auf das gesamte Dokument zugreifen müssen.
 - Programmierpraktikum!

XML

DOM-Parser

<Finale>

Verein="FC Bayern München"

Begegnung= „Champions League Finale 2012“

Gegner="FC Chelsea"

Entfernung_zum_Tor_in_Metern="11"

<Schuetze

Name=„Arjen Robben“

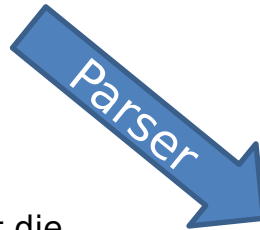
Trikotnummer="10"

</Schuetze>

Verwandelt="false"

Auswirkung="Das Land Bayern legalisiert die Robbenjagd"

</Finale>



Finale

Verein

Begegnung

Gegner

Entfernung
zum Tor

Schütze

Verwandelt

Auswirkung

Name

Trikotnr.

Quellen

- <http://www.w3.org/XML/>
- [**http://openbook.galileocomputing.de/java7/index.html**](http://openbook.galileocomputing.de/java7/index.html)
- <http://de.wikipedia.org/wiki/Xml>
- http://de.wikipedia.org/wiki/Simple_API_for_XML
- http://de.wikipedia.org/wiki/Document_object_model
- <http://www.w3.org/DOM/>
- http://de.wikipedia.org/wiki/JavaScript_Object_Notation
- <http://docs.oracle.com/javase/6/docs/technotes/guides/xml/index.html>