

INSY

Einführung Datenaustauschformate — XML, JSON, CSV

Dominik Dolezal

Höhere Lehranstalt für Informationstechnologie

30. Jänner 2017

Datenaustauschformate

CSV

JSON

XML

Ein Webserver ist zuständig für die Verwaltung von Adressdaten. Wie können die Daten (z.B. Liste aller Adressen) *strukturiert* an einen Client übertragen werden?



Klassisches **tabellenbasiertes** Austauschformat

f. name	name	address	city	state
James	Smith	6649 N Blue Gum St	New Orleans	LA
Jenna	Darakjy	4 B Blue Ridge Blvd	Brighton	MI
Art	Venere	8 W Cerritos Ave 54	Bridgeport	NJ
Lenna	Paprocki	639 Main St	Anchorage	AK

- ▶ Comma-separated **values**
- ▶ Simple Textdateien mit speziellen Regeln
- ▶ Format für
 - ▶ Austausch zwischen Anwendungen
 - ▶ Übertragung im Web
 - ▶ Import-/Export-Funktionalitäten

Beispiel

f. name	name	address	city	state
James	Smith	6649 N Blue Gum St	New Orleans	LA
Jenna	Darakjy	4 B Blue Ridge Blvd	Brighton	MI
Art	Venere	8 W Cerritos Ave 54	Bridgeport	NJ
Lenna	Paprocki	639 Main St	Anchorage	AK

```
f. name;name;address;city;state  
James;Smith;6649 N Blue Gum St;New Orleans;LA  
Jenna;Darakjy;4 B Blue Ridge Blvd;Brighton;MI  
Art;Venere;8 W Cerritos Ave 54;Bridgeport;NJ  
Lenna;Paprocki;639 Main St;Anchorage;AK
```

```
f. name;name;address;city;state  
James;Smith;6649 N Blue Gum St;New Orleans;LA  
Jenna;Darakjy;4 B Blue Ridge Blvd;Brighton;MI  
Art;Venere;8 W Cerritos Ave 54;Bridgeport;NJ  
Lenna;Paprocki;639 Main St;Anchorage;AK
```

Sieht hier jemand einen Widerspruch?



f. `name;name;address;city;state`

- ▶ CSV ist *nicht* standardisiert!
- ▶ Lediglich allgemeine RFC-Beschreibung (RFC 4180)
- ▶ Nicht einmal der Zeichensatz ist geregelt
- ▶ Es gibt jedoch gängige Dialekte
- ▶ Mehrere mögliche Trennzeichen zwischen *Zellen*:
Semikolon ; Komma , Tabulator \t Doppelpunkt : Leerzeichen
- ▶ Mehrere mögliche Trennzeichen zwischen *Zeilen*:
Newline \n oder CRLF (carriage return line feed) \r\n

Wie stellen wir Trennzeichen (z.B. ;) als Text in einer Zelle dar?



James;Peter	Smith	6649 N Blue Gum St	New Orleans	LA
-------------	-------	--------------------	-------------	----

"James;Peter";Smith;6649 N Blue Gum St;...

- ▶ Zellen werden durch Anführungszeichen escaped
- ▶ Je nach Programm werden manchmal auch standardmäßig alle Zellen automatisch escaped

Und wenn ich nun Anführungszeichen als Text verwenden möchte?



James;" Peter"	Smith	6649 N Blue Gum St	New Orleans	LA
----------------	-------	--------------------	-------------	----

"James;" "Peter""";Smith;6649 N Blue Gum St;...

- ▶ Anführungszeichen werden fürs Escapen verdoppelt

f. name	name	address	city	state
James	Smith	6649 N Blue Gum St	New Orleans	LA
Jenna	Darakjy	4 B Blue Ridge Blvd	Brighton	MI

- ▶ CSV-Dateien sind simple Textdateien mit speziellen Regeln
- ▶ Es gibt verschiedene Dialekte
- ▶ Eignen sich gut für tabellarische Informationen
- ▶ Eignen sich schlecht für
 - ▶ hierarchische Beziehungen
 - ▶ referentielle Beziehungen
 - ▶ Unterstützung von Layout
 - ▶ benutzerdefinierte Regeln (z.B. nur Zahlen, E-Mail-Adressen)

- ▶ **JavaScript Object Notation**
- ▶ Ebenfalls nur eine RFC-Beschreibung
- ▶ JSON-Dokumente sind (mit kleinen Ausnahmen) valider JavaScript-Code, welcher 1:1 in JavaScript-Objekte umgewandelt werden kann
- ▶ Wird aber auch in (fast allen) anderen Programmiersprachen verwendet
- ▶ Hauptanwendungsgebiete sind JavaScript-Applikationen, Ajax, Webapplikationen und Webservices

```
[{  
  "first_name": "James",  
  "last_name": "Smith",  
  "address": "6649 N Blue Gum St",  
  "city": "New Orleans",  
  "state": "LA",  
  "zip": 70116,  
  "female": false,  
  "phones": ["504-621-8927", "504-845-1427"]  
}, {  
  "first_name": "Jenna",  
  "last_name": "Darakjy",  
  "address": "4 B Blue Ridge Blvd",  
  "city": "Brighton",  
  "state": "MI",  
  "zip": 48116,  
  "female": true,  
  "phones": ["810-292-9388", "810-374-9840"]}]
```

- ▶ JSON kennt Datentypen
 - ▶ Strings, Boolean, Zahlen
 - ▶ null
 - ▶ Arrays (in eckigen Klammern [])
 - ▶ Objekte (in geschwungenen Klammern {})
- ▶ Objekte beinhalten eine (ungeordnete) Liste von Eigenschaften
- ▶ Eine Eigenschaft besteht aus einem Schlüssel (Zeichenkette) und einem Wert
- ▶ Hierarchien können dargestellt werden
- ▶ Einfache Repräsentation (Serialisierung) von Objekten der objektorientierten Programmierung
- ▶ Es sind jedoch keine komplexen Zusammenhänge darstellbar oder fortgeschrittene Validierungen möglich

- ▶ Extensible **M**arkup **L**anguage
- ▶ Eine Auszeichnungssprache
- ▶ Ebenfalls Textdokumente, die „menschenslesbar“ sein sollen
- ▶ Strukturierte Darstellung von Informationen (ggf. auch inkl. benutzerdefinierten Regeln)
- ▶ Ebenfalls zur Serialisierung von Objekten geeignet
- ▶ Anwendung bei Webapplikationen, Webservices, Konfigurationen, ...
- ▶ Spezifikation von der W3C


```
<?xml version="1.0"?>
<menu>
  <food calories="650">
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
      Two of our famous Belgian Waffles with plenty of real
      maple syrup
    </description>
  </food>
  <food calories="900">
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>
      Light Belgian waffles covered with strawberries
    </description>
  </food>
</menu>
```

```
<menu>
  <food calories="650">
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
      Two of our famous Belgian Waffles with plenty of real
      maple syrup
    </description>
  </food>
</menu>
```

- ▶ Es gibt immer genau ein Wurzelement
- ▶ Elemente werden geöffnet und müssen wieder geschlossen werden
- ▶ Elemente können weitere Elemente sowie Attribute beinhalten

XML vs. HTML

HTML beschreibt Layout des Inhalts, XML beschreibt Struktur/Semantik des Inhalts

```
<h1>HandyKatalog</h1>
<h2>Nokia 8210</h2>
<table border="1">
  <tr>
    <td>Batterie</td><td>900mAh</td>
  </tr>
  <tr>
    <td>Gewicht</td><td>141g</td>
  </tr> ...
</table>
```



```
<HandyKatalog>
  <Hersteller name="Nokia">
    <Modell name="8210">
      <Batterie>900mAh</Batterie>
      <Gewicht>141g</Gewicht>
      ...
    </Modell>
  </Hersteller>
</HandyKatalog>
```

- ▶ **D**ocument-**O**bject-**M**odel
- ▶ Ebenfalls W3C-Standard
- ▶ Für den programmgesteuerten Zugriff auf XML-Dateien
- ▶ Darstellung des Dokuments als Baumstruktur
- ▶ Gesamtes Dokument wird in den Speicher geladen
- ▶ Wahlfreier Zugriff möglich

```
<bookstore>  
  <book category="COOKING">  
    <title lang="en">Everyday Italian</title>  
    <author>Giada De Laurentiis</author>  
    <year>2005</year>  
    <price>30.00</price> ...
```

Softwareentwicklung 4

Document Object Model

Dominik Dolezal

Höhere Lehranstalt für Informationstechnologie

13. Februar 2017

Datenaustauschformate

- ▶ CSV
 - ▶ Einfaches tabellenbasiertes Format nach RFC-Empfehlung
 - ▶ Keine Datentypen, Beziehungen, Layouts, Regeln
- ▶ JSON
 - ▶ RFC-Empfehlung mit hierarchischem Aufbau
 - ▶ Datentypen, Listen und Objekte als Key-Value-Pairs
 - ▶ Keine Validierung möglich
- ▶ XML
 - ▶ Komplexer W3C-Standard für saubere wohlgeformte Dokumente
 - ▶ Beliebige verschachtelte Zusammenhänge
 - ▶ Validierungen möglich