

Semistrukturierte Daten

by

Dr. Günter Kolousek

Semistrukturierte Daten

- ▶ strukturierte Daten
 - ▶ Begriff bezieht sich meist auf relationales Datenmodell
 - ▶ Strukturierung in Relationen, d.h. jedes Tupel hat den gleichen Aufbau
- ▶ semistrukturierte Daten
 - ▶ Daten nicht nach dem relationalen Datenmodell
 - ▶ Schema nicht notwendigerweise vorhanden
 - ▶ tragen einen Teil der Strukturinformation mit sich

Anwendungen & Formate

- ▶ Anwendungen
 - ▶ flexible und unregelmäßige Daten
 - ▶ viele optionalen Werte
 - ▶ unterschiedliche Datentypen
- ▶ Formate
 - ▶ XML
 - ▶ JSON
 - ▶ YAML
 - ▶ CSV

- ▶ JavaScript Object Notation
- ▶ Unterstützung für "alle" Programmiersprachen vorhanden
- ▶ Anwendungen
 - ▶ Konfiguration
 - ▶ Datenspeicherung
 - ▶ Kommunikation
 - ▶ JSON-RPC
 - ▶ AJAX (Asynchronous JavaScript and JSON)

Definition in BNF

```
<object> ::= {} | { <members> }  
<members> ::= <pair> | <pair>, <members>  
<pair> ::= <string> : <value>  
<array> ::= [] | [ <elements> ]  
<elements> ::= <value> | <value>, <elements>  
<value> ::= <string> | <number> | <object> |  
           <array> | true | false | null
```

Beispiel

```
{  
  "key1" : 1,  
  "key2" : "abc",  
  "key3" : null,  
  "key4" : [-1, true, [false, 2.5e-10, {}]],  
  "key5" : { "key6" : { "key7" : "a\tb\tc\n" } }  
}
```

Erweiterungen

- ▶ JSON Pointer
 - ▶ zum Identifizieren eines Wertes innerhalb eines JSON-Dokuments
- ▶ Beispiel
 - ▶ JSON-Dokument

```
{  
  "foo" : ["bar", "baz"],  
  "pi"  : 3.1416  
}
```
 - ▶ `/foo` → `["bar", "baz"]`
 - ▶ `/foo/0` → `"bar"`
 - ▶ `/foo/1` → `"baz"`
 - ▶ `/pi` → `3.1416`

Erweiterungen – 2

- ▶ JSON Schema
 - ▶ zum Beschreiben des Formats von JSON-Dokumenten
- ▶ Beispiel

- ▶ Schema

```
{  
  "$schema" : "http://json-schema.org/schema#",  
  "allof" : [  
    { "type" : "number" },  
    { "type" : "string" }  
  ]  
}
```

- ▶ gültige JSON-Dokumente

- ▶ 42
 - ▶ -1
 - ▶ 2.5e8
 - ▶ "abc"

- ▶ RFC 4180
- ▶ Jede Zeile enthält einen Datensatz (getrennt durch CRLF)
- ▶ Letzte Zeile kann mit einem CRLF abgeschlossen sein
- ▶ Eine Headerzeile kann vorhanden sein und enthält die Namen der Felder; gleiche Anzahl wie in Records (SOLL)
- ▶ Innerhalb jedes Records dürfen mehrere Felder enthalten sein (getrennt durch ein ,)
- ▶ Jedes Feld kann in " eingeschlossen sein
- ▶ Felder die ein CRLF, ein " oder ein , enthalten sollen in " eingeschlossen sein
- ▶ Wenn " in einem Feld vorkommen, dann mittels einem weiteren " maskieren