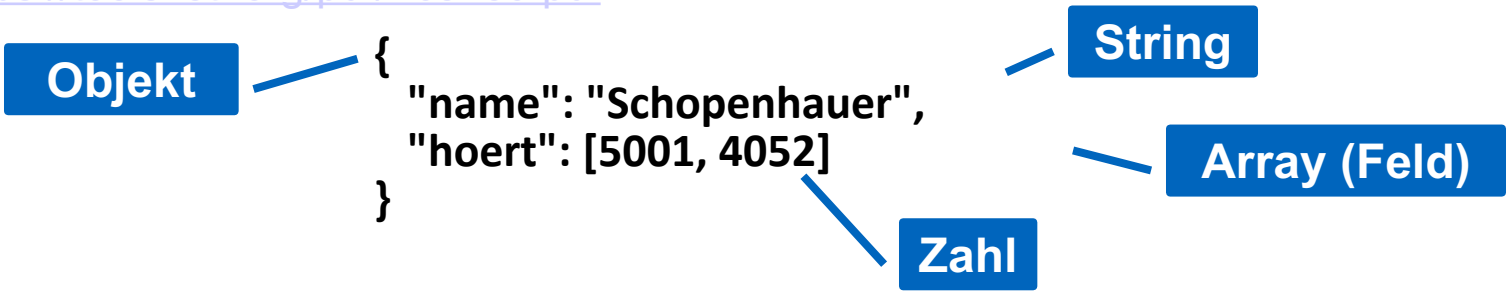
ERDB – JSON

Maximilian E. Schüle
i3erdb@in.tum.de

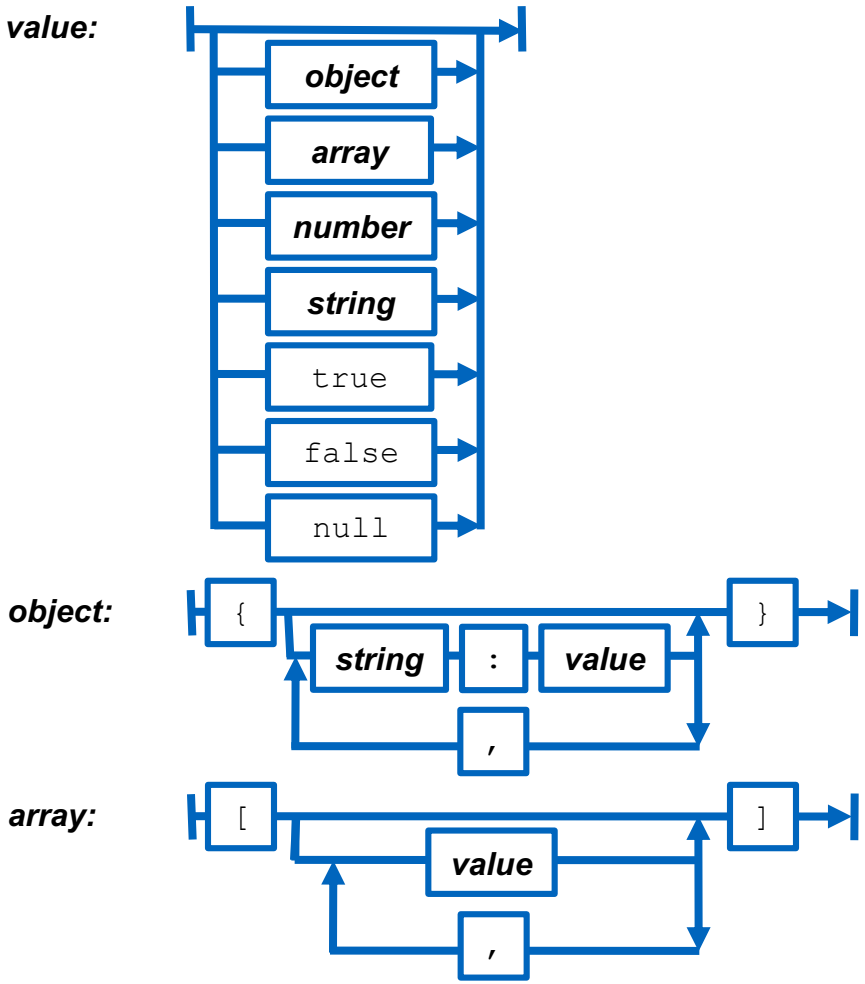
JSON

- Ziel: menschen- und maschinenlesbares Datenaustauschformat
- Semistrukturiert, kein festes Schema, ähnlich zu XML (und ersetzt es schrittweise)
- Ursprung: ECMAScript (1999), ähnlich zu JavaScript; standardisiert 2017 [1] [2] [3]
- Key-Value-Paare (vgl. NoSQL)
- Strukturtoken: { } [ ] : ,
- Verwendung in Datenbanksystemen für komplexe Datentypen oder zum Datenaustausch

[1] https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf
[2] https://json.org
[3] https://tools.ietf.org/pdf/rfc8259.pdf



JSON: Syntax



Gültige JSON-Ausdrücke:

```
{
  "name": "Schopenhauer"
}
```

```
"hoert": [5001, 4052]
```

```
42
```

```
"Schopenhauer"
```

```
true
```

JSON: Vor- und Nachteile zu XML

- XML schwierig zu parsen, JSON intuitiv
- XML: Position der Elemente wichtig; JSON: nur innerhalb Arrays
- JSON unterscheidet nicht zwischen Attributen und Elementen
- In JSON ebenfalls gemischte Datentypen möglich
  - Deshalb JSON Schema https://json-schema.org/

"hoert": [5001, "V4052"]

Zahl | String

```
{
 "$schema": "http://json-schema.org/schema#",
 "title": "Universitaet",
 "type": "object",
 "required": [
  "name",
  "UniLeitung",
  "Fakultaeten"
 ],
 "properties": {
  "Name": {
   "type": "string",
   "description": "Name der Universität"
  },
  ...
 }
}
```

JSON: Uni-Schema (Ausschnitt)

```
{
 "Name": "Virtuelle Universitaet der Grossen Denker",
 "UniLeitung": {"Rektor": "Sokrates", "Kanzler": "Erhard"},
 "Fakultaeten":
 [
  {
   "Name": "Philosophie",
   "Professoren":
   [
    {
     "PersNr": 2125,
     "Name": "Sokrates",
     "Rang": "C4",
     "Vorlesungen":
     [
      {"VorlNr": 5041, "Titel": "Ethik", "SWS": 4},
      {"VorlNr": 5049, "Titel": "Maeeutik", "SWS": 2},
      {"VorlNr": 4052, "Titel": "Logik", "SWS": 4}
     ]
    }
   ]
  }
 ]
}
```

JSON: Auslesen

- Bisher keine standardisierte Abfragesprache wie XPath und XQuery zu XML
- JavaScript: eval(), aber Ausdruck könnte Schadcode enthalten, daher JSON.parse()
- C++: PicoJson
  - https://github.com/kazuho/picojson/blob/master/picojson.h
- Java:
  - import org.json.JSONObject; import org.json.JSONArray;
- JavaScript: JSONPath, jpath, JSPath, uvm. http://www.jsonquerytool.com/
- Bash: jq
  - https://stedolan.github.io/jq/ https://jqplay.org/
  - curl '<dataset url>' | jq '<filter>'

```
$ cat uni.json | jq '.Fakultaeten[].Professoren[].Vorlesungen[] | select (.VorlNr==5041).Titel'
"Ethik"

$ curl https://raw.githubusercontent.com/openfootball/football.json/master/2016-17/de.1.json
 | jq '.rounds[].matches[] | select(.score1==2) | select(.score2==1).team1.name'
```