

JSON

Sahar Nafisi

Data Mining presentation

Supervisor: Dr. S. M. Vahidipour

University of Kashan

Fall 1396

What is JSON

- ▶ JavaScript Object Notation
- ▶ JSON is a (mostly) language-independent way of specifying objects as name-value pairs
- ▶ Is lightweight format for exchanging data between the client and server.
- ▶ Easy for humans to read and write.
- ▶ Used to format data

Why JSON?

- ▶ JSON is recognized natively by JavaScript
- ▶ Simple format
- ▶ The easiness of reading
- ▶ The easiness of using
- ▶ Supported by all programming languages
- ▶ Lighter than XML

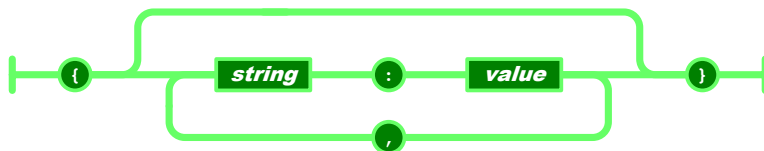
Example

```
{  
  "name": "sahar",  
  "age": 22,  
  "cars": ["pride", "bmw", "peugeot"],  
  "address": {  
    "city": "karaj",  
    "street": "golshar",  
    "alley": "bahar"  
  }  
}
```

JSON Syntax

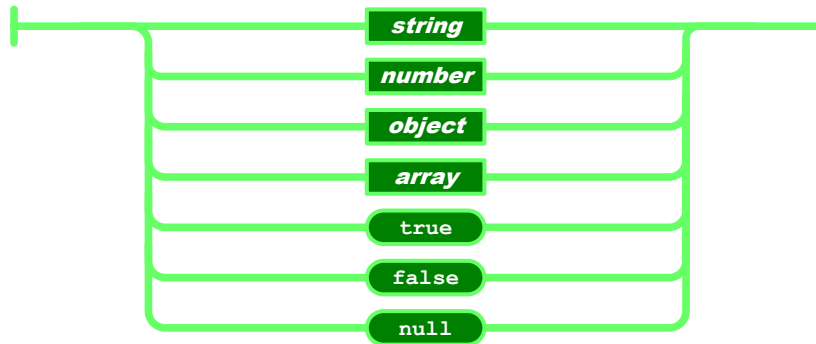
- ▶ An *object* is an unordered set of name/value pairs
 - ▶ The pairs are enclosed within braces, { }
 - ▶ There is a colon between the name and the value
 - ▶ Pairs are separated by commas
 - ▶ Example: { "name": "html", "years": 5 }
- ▶ An *array* is an ordered collection of values
 - ▶ The values are enclosed within brackets, []
 - ▶ Values are separated by commas
 - ▶ Example: ["html", "xml", "css"]

Object



Value

value



JSON Syntax

- ▶ A *value* can be: A string, a number, **true**, **false**, **null**, an object, or an array
 - ▶ Values can be nested
- ▶ *Strings* are enclosed in double quotes, and can contain the usual assortment of escaped characters
- ▶ *Numbers* have the usual C/C++/Java syntax, including exponential (E) notation
 - ▶ All numbers are decimal--no octal or hexadecimal
- ▶ *Whitespace* can be used between any pair of tokens

Accessing Data in JSON

- ▶ The most common way to access JSON data is through dot notation.
- ▶ `var myObject = { 'color' : 'blue' };`
- ▶ `document.writeln(myObject.color);`

Comparison of JSON and XML

- ▶ Similarities:
 - ▶ Both are human readable
 - ▶ Both have very simple syntax
 - ▶ Both are hierarchical
 - ▶ Both are language independent
 - ▶ Both can be used by Ajax
 - ▶ Both supported in APIs of many programming languages

Comparison of JSON and XML

► Differences:

- Syntax is different
- JSON is less verbose
- JSON includes arrays
- Easier for machines to parse and generate
- Names in JSON must not be JavaScript reserved words

JSON	XML
JSON object are type	XML data is typeless
JSON types: string, number, array, boolean	XML data are all string
Data is readily accessible as JSON objects	XML data needs to be parsed
Retrieving value is easy	Retrieving value is difficult
JSON is supported by all browsers	Cross browser XML parsing can be tricky
SimpleAPI	ComplexAPI
Supported by many Ajax toolkit	Not fully supported by Ajax toolkit
Fast object de-serialization in JavaScript	Slower de-serialization in JavaScript
Fully automated way of de-serializing/serializing JavaScript objects	Developers have to write JavaScript code to serialize/de-serialize to/from XML

Pros	Cons
Fast to parse	No namespace support, hence poor extensibility
Good support for all browsers	Limited development tools support
Supported by many languages	No support for formal grammar definition
Concise format: name/value pair - based approach	
Easy to read	
Easy to write	

REST and RESTful web services

- ▶ REST is about resources and how to represent resources in different ways.
- ▶ REST is about client-server communication.
- ▶ REST is about how to manipulate resources.
- ▶ REST offers a simple, interoperable and flexible way of writing web services that can be very different from other techniques.

What is REST?

- ▶ **R**epresentational **S**tate **T**ransfer
- ▶ Architectural style (technically not a standard)
- ▶ REST is an architecture all about the Client-Server communication.
- ▶ Uses existing standards, e.g., HTTP

What is REST?

- ▶ Client **requests** a specific **resource** from the server.
- ▶ The server **responds** to that request by delivering the requested resource.
- ▶ Server does not have any information about any client.
- ▶ So, there is no difference between the two requests of the same client.
- ▶ A model which the representations of the resources are transferred between the client and the server.

An Architectural Style

- ▶ REST is the architecture of the Web as it works today and, so it is already used in the **web**!
- ▶ It is an software architectural model which is used to describe distributed systems like **WWW** (World Wide Web).
- ▶ It has been developed in parallel with **HTTP** protocol.