JSON

Was ist JSON?

* steht für JavaScript Object Notation
* leichtgewichtiges Daten-Austausch-Forman
* geschrieben in JavaScript Objekt-Notation
* wird benutzt, um Daten zwischen Computern zu senden
* ist sprachunabhängig

Warum JSON?

Ist syntaktisch ähnlich wie Code für JavaScript-Objekte, daher kann JavaScript JSON-Daten einfach in JavaScript-Objekte konvertieren.

JavaScript hat eine Funktion, um JSON-Strings zu JS-Objekten umzuwandeln:  
JSON.parse()

Ein Objekt kann auch in einen JSON-String umgewandelt werden:  
JSON.stringify()

Man kann puren Text von einem Server in einem JS-Objekt verwenden.

Man kann ein JS-Objekt in purem Text zu einem Server senden.

Man kann mit Daten als ein JS-Objekt arbeiten, ohne kompliziertes parsen und übersetzen.

Daten speichern

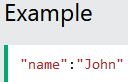
Um Daten zu speichern, müssen sie in einem bestimmten Format sein. Text ist immer ein gültiges Format.

JSON ermöglicht die Speicherung von JS-Objekten als Text.

JSON Syntax

* Daten sind in Name/Wert-Paaren
* Daten werden durch Kommas getrennt
* Geschwungene Klammern beinhalten Objekte
* Eckige Klammern beinhalten Arrays

JSON Daten - Name und Wert



JSON Werte

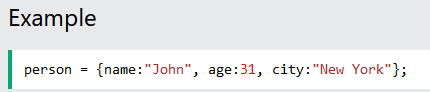
JSON-Werte müssen in folgenden Datentypen sein:

* String
* Nummer
* Objekt
* Array
* Boolean
* null

JavaScript-Werte können aus obigen Datentypen bestehen und zusätzlich:

* Funktion
* Datum
* undefined

JavaScript Objekte



Ein Bild, das Text enthält.

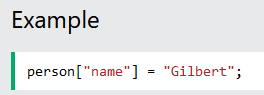
Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



JSON vs. XML

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

JSON ist wie XML, weil

* beide sind selbstbeschreibend (lesbar für Menschen)
* beide sind hierarchisch (Werte innerhalb von Werten)
* beide können geparst werden und von anderen Programmiersprachen benutzt werden
* beide können mit XMLHttpRequest gefangen werden

JSON ist nicht wie XML, weil

* JSON benötigt keinen end-tag
* JSON ist kürzer
* JSON ist schneller zu lesen und zu schreiben
* JSON kann Arrays benutzen

Der größte Unterschied:

XML muss mit einem XML-Parser geparst werden. JSON kann mit einer Standard-JS-Funktion geparst werden.

Warum ist JSON besser als XML?

XML ist schwieriger zu parsen.

JSON kann in ein sofort nutzbares JS-Objekt geparsed werden.

JSON Datentypen

* String
* Nummer
* Objekt (JSON Objekt)
* Array
* Boolean
* null

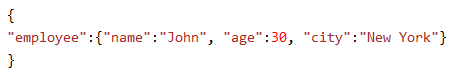
String:



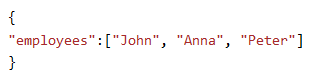
Nummer:



Objekt:



Array:



Boolean:



null:



JSON.parse()

Eine häufige Verwendung für JSON ist der Datenaustausch mit einem Webserver.

Wenn man Daten vom Server bekommt, sind die Daten immer ein String.

Mit JSON.parse() werden Daten in ein JS-Objekt umgewandelt.

Exceptions

Datum parsen

Datums-Objekte sind in JSON nicht erlaubt.

Wenn ein Datum benötigt wird, muss es als String geschrieben werden.

Man kann den String später zurück in ein Datums-Objekt konvertieren.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Funktionen parsen

Funktionen sind in JSON nicht erlaubt, auch sie muss als String geschrieben werden.

Kann später wieder zurück konvertiert werden:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Funktionen sollten in JSON vermieden werden.

JSON.stringify()

Wenn Daten zu einem Server gesendet werden, müssen sie in einem String sein.

Stringify eines JS-Objekts

JavaScript-Objekt:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Stringify eines Arrays



Daten in lokalem Speicher

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ausnahmen (funktioniert nicht…)

* Datum
* Funktionen



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

JSON Objekte



JSON-Objekte stehen innerhalb von {}

JSON-Objekte beinhalten Key/Value-Paare.

Keys und Values werden durch : getrennt.

Keys müssen Strings sein, Values können in diesen Datentypen sein:

* String
* Nummer
* Objekt (JSON Objekt)
* Array
* Boolean
* null

JSON ist kein Objekt, es ist ein String.

Daten sind nur JSON, wenn sie in einem String sind. Wenn sie in eine JS-Variable konvertiert werden, sind sie JS-Objekte.

JavaScript-Objekte



Werte auslesen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

JSON-Arrays

JSON-String:

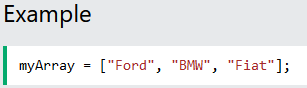


Im JSON-String ist ein JSON-Array literal:



Arrays in JSON sind fast gleich wie Arrays in JavaScript.

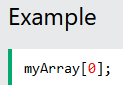
JavaScript-Arrays



Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Zugriff auf Array-Werte

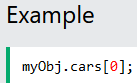


Arrays in Objekten

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Zugriff:



JSON Server

Daten Senden

Wenn Daten in einem JS-Objekt gespeichert sind, können sie in JSON konvertiert und an den Server gesendet werden:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Daten Empfangen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

JSON von einem Server

Man kann JSON-Daten mit einem AJAX-Request anfragen.

So lange die Antwort vom Server im JSON-Format ist, kann es in ein JS-Objekt geparst werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Array als JSON

Wenn auf einem JSON, das aus einem Array abgewandelt wurde, JSON.parse() angewendet wird, wird die Methode ein JS-Array anstatt eines JS-Objektes zurückgeben.

JSON PHP

Das PHP-File

PHP hat einige eingebaute Funktionen um JSON zu handeln.

Objekte in PHP können mit json\_encode() in JSON konvertiert werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Client JavaScript

Hier wird ein AJAX-Aufruf gesendet:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

PHP-Array

Arrays in PHP können mit json\_encode() in JSON konvertiert werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

PHP-Datenbank

PHP ist eine serverseitige Programmiersprache, und kann für den Zugriff auf eine Datenbank verwendet werden.

Wenn auf einem Server eine Datenbank liegt, und man will die ersten 10 Zeilen aus einer Tabelle „customers“ abfragen.

Man erstellt am Client ein JSON-Objekt, das die Anzahl der Zeilen beschreibt.

Bevor der Request an den Server gesendet wird, wird das JSON-Objekt in einen String umgewandelt und als Parameter an die URL der PHP-Seite gesendet:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* Objekt mit „limit“-Key und Value wird erstellt
* Objekt wird in einen JSON-String umgewandelt
* JSON-String wird als Parameter an das PHP-File gesendet
* Warten bis Ergebnis (JSON) zurückkommt
* Erhaltenes Ergebnis anzeigen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* Request wird mit json\_decode() in ein Objekt konvertiert
* Zugriff auf Datenbank erfolgt, und Array mit angefragten Daten wird gefüllt
* Array wird an ein Objekt angehängt und das JSON-Objekt wird mit json\_encode() zurück gegeben

Daten benutzen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

PHP-Methode = POST

Wenn Daten an den Server gesendet werden, ist es oft am besten, die HTTP-POST-Methode zu verwenden.

Um einen AJAX-Request mit der POST-Methode zu senden, muss die Methode spezifiziert werden und der korrekte Header muss benutzt werden.

Die Daten, die zum Server gesendet werden, müssen als Parameter der send()-Methode übergeben werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

JSON HTML

JSON kann leicht in JavaScript übersetzt werden.

JavaScript kann benutzt werden, um HTML in der Webseite zu erzeugen.

HTML-Tabelle

HTML-Tabelle mit Daten aus einem JSON erstellen:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

JSONP

JSONP ist eine Methode, um JSON-Daten zu senden, ohne sich über cross-domain Probleme Sorgen machen zu müssen.

JSONP benutzt keint XMLHttpRequest-Objekt.

JSONP benutzt <script>-tag.

JSONP Intro

JSONP heißt JSON mit Padding.

Eine File-Anfrage von einer anderen Domain kann zu Problemen führen.

Eine Skript-Anfrage von einer anderen Domain hat dieses Problem nicht.

JSONP nutzt diesen Vorteil und benutzt das <script>-tag statt XMLHttpRequest.

Server-File

Das File auf dem Server benutzt das Ergebnis in einem Funktionsaufruf:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Das Ergebnis retourniert einen Aufruf einer Funktion namens „myFunc“ mit den JSON-Daten als Parametern.

Die Funktion muss auf dem Client existieren.

Die JavaScript-Funktion

Die „myFunc“-Funktion ist auf dem Client und bereit, die JSON-Daten zu handeln:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein dynamisches Script-Tag erstellen

Das obere Beispiel führt „myFunc“ aus, wenn die Seite geladen wird. Je nachdem wo das <script> gesetzt wurde.

Das <script> sollte erstellt werden wenn es gebraucht wird.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Dynamisches JSONP-Ergebnis

Das Beispiel wird dynamisch, in dem ein JSON an das PHP-File gesendet wird und das File ein JSON zurückgibt, basierend auf den Informationen, die das File erhält.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* Request in ein Objekt konvertieren mit json\_decode()
* Zugriff auf Datenbank und Array mit Request-Daten füllen
* Array zu einem Objekt hinzufügen
* Array mit json\_encode() in JSON konvertieren
* „myFunc()“ um retourniertes Objekt legen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Callback Funktion

Wenn man über das Server-File keine Kontrolle hat, wie weiß man, ob das Server-File die richtige Funktion ausführt?

Manchmal bietet das Server File eine Callback-Funktion als Parameter an:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung