

LE FORMAT « JSON »

Contenu

Un peu d'histoire.....	1
Les enjeux.....	1
Avantages de JSON.....	1
JSON : Le format du « Web ».....	2
Syntaxe de JSON.....	2
Types et structures de données	3
Exemple d'utilisation de JSON.....	3
Limitations du format JSON.....	4
Pour résumer	4
Historique du document	5
Crédits	5

Un peu d'histoire

JSON est né de la RFC 4627 de l'IETF (Internet Engineering Task Force).

Il a été créé par Douglas Crockford et a été cité dans la RFC 7159 pour une mise à jour.

Comme JavaScript ou C++, JSON est dérivé de l'ensemble de normes ECMAScript créées par l'organisme ECMA (European Computer Manufacturers Association) pour la spécification ECMA-262.

Les enjeux

De nos jours, de nombreux formats voient le jour, mais l'important est qu'ils existent suffisamment longtemps pour devenir un standard ou au moins bénéficier d'une norme. Plus un format est accessible et compréhensible par le plus grand nombre, plus il est utilisé, plus il a des chances de devenir un standard.

S'ajoutent à ce premier critère, un critère de performance, pouvant influencer sur la pérennité du format. Lorsque le format JSON a été créé, il devait être rapide à traiter, il fallait qu'un fichier prenne peu de place sur disque, ne consomme pas trop de ressources, comme la mémoire vive, ni de temps de transit sur un réseau.

Également, cela devait être un code facile à manipuler et à traduire pour la machine, permettant d'encoder de grandes quantités de données rapidement.

JSON est né dans l'ère de l'objet et, en tant que format orienté objet, il permet donc d'échanger avec des API manipulant des objets et, moyennant des méthodes pour transcrire les objets, il permet d'échanger avec des API non objet.

Avantages de JSON

JSON (JavaScript Object Notation) est un format léger d'échange de données basé sur un sous-ensemble du langage JavaScript. Il a l'avantage d'être facile à lire ou à écrire pour des humains et d'être aisément exploitable par les machines telles que nos ordinateurs.

JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendant du C, comme par exemple :

- C (lui-même), C++, C#, Java, JavaScript, Perl, Python, PHP, et bien d'autres...

JSON est un format à faible encombrement de stockage ce qui permet une grande vitesse au niveau des échanges ainsi que des économies au niveau de l'espace disque et de la bande passante.

Ces propriétés font de JSON un format d'échange de données idéal.

JSON : Le format du « Web »

En effet, la légèreté du format JSON fait qu'il est de plus en plus utilisé pour l'échange de données entre applications au détriment du format XML, plus verbeux et plus complexe.

Sa lisibilité par l'humain permet une vérification et un accès aux contenus rapide. Ce format permet donc de faire face à des besoins d'échanges de plus en plus importants.

En tant que format de texte capable de s'adapter à de multiples langages, JSON permet de sérialiser, représenter les données, les décrire, les échanger de manière simple et standard.

La popularité de ce format tient aussi au fait que des données JSON peuvent être facilement converties en objet dans de nombreux langages (JavaScript via `JSON.parse()`, PHP via `json_decode()` etc...).

Syntaxe de JSON

En JSON, on structure chaque élément avec un couple « nom=valeur ».

Un exemple simple de code JSON contenant une variable nommée "Acronyme" et ayant pour valeur "CDA":

```
{ "Acronyme": "CDA" }
```

Tout objet est déclaré entre accolades. On utilise des guillemets pour entourer chaque valeur, les crochets pour déclarer un tableau. Les virgules séparent chaque valeur. L'objet JSON est lui-même encapsulé dans des accolades.

Il est important de comprendre qu'une variable peut prendre plusieurs types de valeur. Chaque variable peut contenir une valeur « null », un nombre, une chaîne de caractère, un tableau ou un autre objet JSON.

Un exemple de code :

```
var monObjet = {  
  "fruits": [  
    "articles": {  
      "fraises": 3,  
      "poires": 4,  
      "pommes": null  
    },  
    { "panier": true }  
  ],  
  "legumes": {  
    "carottes": "oui",  
    "poireaux": false  
  },  
  "boissons": ["eau", "eau gazeuse", "soda"]  
}
```

Dans l'exemple ci-dessus, la variable « monObjet » contient :

« fruits » : Un tableau contenant 2 objets. L'objet « articles » possédant 3 éléments et « panier » ayant pour valeur « true ».

« legumes » : Un objet contenant 2 variables (carottes, poireaux).

« boissons » : Un tableau contenant 3 valeurs (eau, eau gazeuse, soda).

Types et structures de données

JSON supporte l'objet, mais en JSON tout n'est pas un objet comme en java ou en C#.

Pour rappel,

Chaque élément est un couple avec une clé et une valeur.

On peut tout de même, en JSON, utiliser ces types : booléen, nombre, char, string avec des caractères 'unicode', null.

Comme structure de données on peut utiliser les tableaux, entre crochets.

Chaque valeur est séparée par un caractère virgule.

JSON ne reconnaît que les nombres décimaux négatifs ou positifs et les puissances de nombres. Il n'est pas adapté aux fichiers binaires ni aux nombres fictifs (ex : NaN).

Le format JSON interdit formellement d'insérer des commentaires, et les caractères accentués se codent sous leur forme Unicode (ex : \u00e9). (➤ [Table des caractères Unicode](#)).

Exemple d'utilisation de JSON

JSON est disponible pour plus de 50 langages, dont JavaScript, PHP, Perl, Python, Ruby, Java.

Il est également portable, contrairement à d'autres formats de textes nécessitant l'installation du DOM ou autre.

Côté client, JSON est particulièrement simple à utiliser avec JavaScript. Côté serveur, PHP et Java l'utilisent par le billet de parseurs.

JSON peut être utilisé par invocation directe dans une page HTML comme un fichier JavaScript ; Ou bien il peut être appelé par la méthode XMLHttpRequest. On peut également le charger via une commande JavaScript.

Ci-dessous un exemple de chargement d'un fichier JSON avec la méthode XMLHttpRequest:

Soit le fichier « fichier.JSON » :

```
{
  "menu": { "value": "Accueil" },
  "commandes":
  [
    {
      "titre": "Articles",
      "action": "get_articles",
    }
  ]
}
```

La requête AJAX

```
var req = new XMLHttpRequest();
req.open("GET", "fichier.JSON", true);
req.onreadystatechange = maFonction; // la fonction de prise en charge
req.send(null);
```

La fonction JavaScript associée:

```
function maFonction ()  
{  
  if (req.readyState == 4)  
  {  
    var result = eval('(' + req.responseText + ')');  
  }  
}
```

Un exemple d'utilisation des données :

```
var nomMenu = document.getElementById('jsmenu'); // trouver un champ  
nomMenu.value = result.menu.value; // assigner une valeur au champ
```

... Et la manière d'y accéder :

```
result.commandes[0].titre // lire la valeur de "titre" dans le tableau  
result.commandes[0].action // lire la valeur de "action" dans le tableau
```

Limitations du format JSON

	Valeur MAX recommandée	Valeur MAX théorique
Taille MAX d'un objet JSON	4 Mo	128 Mo
Taille MAX d'une clé JSON	256 octets	8 Ko
Taille MAX d'un string JSON	8 Ko	2 Mo
Taille MAX d'un nombre JSON	Valeur MAX INT32	Valeur MAX INT64
Niveaux d'imbrication MAX	64 niveaux	256 niveaux





Pour résumer

- JSON est un format indépendant léger et portable
- JSON est facilement interprétable par les humains et les machines
- Plus de 50 langages de programmation sont nativement compatibles avec JSON
- JSON remplace peu à peu XML dans les échanges de données via les Web Services
- Un objet JSON est encapsulé dans des accolades
- Une variable d'un objet JSON peut contenir : string Unicode, nombre réel, booléen, valeur « null », tableau

--- FIN DU DOCUMENT ---

La reproduction partielle ou intégrale du présent document sur un support, quel qu'il soit, est formellement interdite sans l'accord écrit et préalable du Centre de Réadaptation de Mulhouse.

Légende des icônes

	Information complémentaire
	Point d'attention particulier
	Intervention du formateur possible
	Lien vers une ressource externe

Historique du document

Auteur	Date	Observations
MD v1.0.0	26/09/2018	Création du document

Crédits

<https://www.xul.fr/ajax-format-json.php>

https://www.ibm.com/support/knowledgecenter/en/SS9H2Y_7.2.0/com.ibm.dp.doc/json_parserlimits.html