

RFC 4627 : The application/json Media Type for JavaScript Object Notation (JSON)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 août 2006. Dernière mise à jour le 18 janvier 2012

Date de publication du RFC : Juillet 2006

<https://www.bortzmeyer.org/4627.html>

Il existe une pléthore de langages pour décrire des données structurées <<https://www.bortzmeyer.org/data-formats.html>>. XML est le plus connu et voici un de ses concurrents, JSON, décrit dans ce RFC (depuis remplacé par le RFC 8259¹).

JSON existe depuis longtemps mais n'avait pas de norme formelle. C'est désormais fait dans notre RFC, puis dans son successeur, le RFC 8259.

JSON se veut plus léger que XML. Comme son concurrent XML, il permet de représenter des structures de données hiérarchiques.

À noter que JSON doit son origine, et son nom complet ("*JavaScript Object Notation*") au langage de programmation Javascript, dont il est un sous-ensemble. Mais JSON n'est pas un langage de programmation, seulement un langage de description de données, et il ne peut donc pas servir de véhicule pour du code méchant.

Voici un exemple, tiré du RFC, d'un objet exprimé en JSON :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc8259.txt>

```
{
  "Image": {
    "Width": 800,
    "Height": 600,
    "Title": "View from 15th Floor",
    "Thumbnail": {
      "Url": "http://www.example.com/image/481989943",
      "Height": 125,
      "Width": "100"
    },
    "IDs": [116, 943, 234, 38793]
  }
}
```

Les détails sont dans les sections 1 et 2 du RFC. Cet objet d'exemple a un seul champ, `Image`, qui est un autre objet (entre `{` et `}`) et qui a plusieurs champs. Un de ces champs, `IDs`, a pour valeur un tableau.

JSON est donc un format simple, il n'a même pas la possibilité de commentaires <http://stackoverflow.com/questions/244777/can-i-comment-a-json-file> dans le fichier... Voir sur ce sujet une intéressante compilation <http://blog.getify.com/2010/06/json-comments/>.

Voici un exemple d'un programme Python pour écrire un objet Python en JSON (on notera que la syntaxe de Python et celle de JavaScript sont très proches) :

```
import json

objekt = {'Image': {'Width': 800,
                   u'Title': u'View from Smith\'s, 15th Floor, "Nice"',
                   u'Thumbnail': {'Url':
                                   u'http://www.example.com/image/481989943',
                                   u'Width': u'100', u'Height': 125},
                   u'IDs': [116, 943, 234, 38793],
                   u'Height': 600}} # Example from RFC 4627, lightly modified

print json.dumps(objekt)
```

Et un programme pour lire du JSON et le charger dans un objet Python :

```
import json

# One backslash for Python, one for JSON
objekt = json.loads("""
{
  "Image": {
    "Width": 800,
    "Height": 600,
    "Title": "View from Smith's, 15th Floor, \\\"Nice\\\"",
    "Thumbnail": {
      "Url": "http://www.example.com/image/481989943",
      "Height": 125,
      "Width": "100"
    },
    "IDs": [116, 943, 234, 38793]
  }
}
""") # Example from RFC 4267, lightly modified

print objekt
print ""
print objekt["Image"]["Title"]
```

Si vous voulez le faire en Go, il existe un bon article d'introduction <<http://blog.golang.org/2011/01/json-and-go.html>> au paquetage standard `json`.

JSON dispose d'une page Web officielle <<http://json.org>>, où vous trouverez plein d'informations.