RFC 6839 : Additional Media Type Structured Syntax Suffixes

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 janvier 2013

Date de publication du RFC : Janvier 2013

https://www.bortzmeyer.org/6839.html

Tout le monde connait les « types de contenu » (*"media types"*, aussi appelés types MIME pour des raisons historiques), comme `application/pdf` ou `text/plain`. On sait qu'ils ont un **type** (comme `application` ou `text`, dans les exemples ci-dessus) et un **sous-type** (`pdf` ou `plain` ci-dessus) qui précise le format, à l'intérieur de la catégorie définie par le type. On sait moins que le sous-type peut avoir une structure : on peut voir un format générique et un format plus spécifique, séparés par un +, comme dans `application/xhtml+xml` ou `image/svg+xml`. Cette structuration, introduite à l'origine par le RFC 3023 [1], pour XML, est ici généralisée à d'autres formats et normalisée en détail. (Pour `+xml`, une mise à jour a été faite ensuite dans le RFC 7303.)

Cette technique vient du fait que certains formats ont deux couches : un format décrivant la syntaxe de base et plusieurs formats (plutôt appelés vocabulaires, ou même langages, dans le monde XML) qui définissent des règles sémantiques différentes, mais tous utilisant la même syntaxe de base. L'exemple typique est XML. Ainsi, les fichiers TEI (RFC 6129) sont en XML et un processeur XML généraliste saura en faire quelque chose (par exemple vérifier leur validité syntaxique). Mais TEI n'est pas que XML : c'est aussi la définition d'un certain nombre d'éléments XML légaux et de leurs relations (le **schéma**). Un fichier TEI peut donc être étiqueté `application/xml` mais son type de contenu officiel, `application/tei+xml`, donne davantage d'informations (cf. section 2, sur l'intérêt de ces suffixes). Si on prend un autre vocabulaire XML, le Atom du RFC 4287, c'est aussi du XML mais les éléments acceptés sont complètement différents. Son type de contenu aura donc aussi le suffixe `+xml` mais le type sera distinct, ce sera `application/atom+xml`.

La section 3 liste les sous-types à suffixe possibles (mais cette liste va grandir par la suite, cf. RFC 6838, section 6). Pour chaque suffixe, sont notamment indiquées les règles relatives à son encodage (pour savoir si la ressource doit être transportéee comme du texte ou comme du binaire), et les règles de sécurité. Parmi les suffixes possibles, il y a, entre autres :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc3023.txt`

— `+json` pour le format JSON du RFC 7159. comme `vnd.oftn.l10n+json` (format de la bibliothèque Javascript l10n <`https://github.com/eligrey/l10n.js`>) ou comme les réponses de l'API de GitHub <`http://developer.github.com/v3/media/`>. Comme XML, JSON peut être encodé en UTF-8 (il est alors considéré comme texte, cf. RFC 2045) ou UTF-32 (il est alors vu comme du binaire).
— `+ber` pour le format BER et `+der` pour le format DER, tous les deux d'encodage d'ASN.1 (norme UIT ITU.X690.2008). Un exemple est `dssc+der`, du format du RFC 5698 (qui peut aussi se représenter en XML). À noter que BER présente des risques particuliers en terme de sécurité. Encodage TLV, il permet de construire des fichiers avec des longueurs invalides, permettant de dépasser les limites des données et de déclencher un débordement de tampon chez des décodeurs naïfs. On peut aussi faire des dépassements de pile puisque BER permet d'emboîter des données récursivement.
— `+fastinfoset` est la norme Fast Infoset, un encodage binaire de XML, qui ne semble pas avoir été un grand succès (bien que résolvant un reproche parfois fait à XML, celui que son encodage par défaut est trop bavard).
— `+zip` est le fameux format de compression et d'archivage ZIP. (Aucun n'a apparemment encore été enregistré.)

La section 4 reprend le suffixe `+xml`, déjà décrit dans le RFC 3023 et l'adapte aux exigences du RFC 6838.

Le registre IANA de ces suffixes est désormais en ligne <`https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xml`>.