

RFC 6838 : Media Type Specifications and Registration Procedures

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/6838.html>

Tout professionnel de l'Internet voit passer les fameux types de contenu ("*media types*", souvent appelés types MIME) comme `audio/ogg` ou `text/json`. Comment sont enregistrés ces types ? Suivant quelles procédures ? Si je veux un type pour mon super format que j'ai inventé moi, comment je fais ? Ce nouveau RFC, qui remplace l'ancienne norme (RFC 4288¹), donne la réponse à toutes ces questions.

Ces types de contenu sont utilisés dans de nombreux protocoles multimédia. Les deux les plus connus sont HTTP et le format du courrier électronique (RFC 2045 et RFC 5322), le nom de « type MIME » venant d'ailleurs du courrier, même si ces types sont utilisés bien ailleurs. Lorsqu'un client HTTP interroge un serveur, celui-ci se sert ainsi de ces types pour lui indiquer le format des données envoyées :

```
% curl -v http://www.bortzmeyer.org/images/defile-fete-cheval-2008-4.png > /dev/null
...
< Content-Length: 2094295
< Content-Type: image/png
...
```

Ces types sont composés de deux parties, le **type** proprement dit et le **sous-type**, séparés par une barre. Ainsi, dans le cas ci-dessus, `image/png`, le type est `image` (une image) et le sous-type est `PNG` (le format de l'image). Pour assurer l'unicité d'un type, ils sont enregistrés à l'IANA <<https://www.iana.org/assignments/media-types/index.html>> et c'est la gestion de ce registre qui est le cœur de ce RFC.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc4288.txt>

Les noms enregistrés sont regroupés en **arbres**, un arbre étant identifié par une **facette** indiquée dans le sous-type par le texte précédant le point. S'il n'y a pas de point (comme dans tous les exemples donnés jusqu'à présent), il s'agit de l'arbre standard, qui stocke les noms d'intérêt général. Un exemple de nom dans un autre arbre est `image/vnd.microsoft.icon` qui est dans l'arbre « Vendeurs » (`vnd`) et désigne le format ICO.

L'enregistrement dans l'arbre standard est, comme on s'en doute, le plus difficile. Le format en question doit être normalisé par l'IETF ou une autre SDO reconnue par l'IESG (politique « Norme nécessaire » du RFC 5226). Un expert appointé par l'IANA est chargé de réviser ces enregistrements pour s'assurer que tout est complet. Au cas où un changement de l'enregistrement soit nécessaire, c'est la SDO originale qui peut seule décider (section 5.5).

Notez aussi que le sous-type doit normalement désigner un format qui permette l'**interopérabilité**. Si un fichier est étiqueté `image/MACHIN`, un logiciel qui lit ce format MACHIN doit pouvoir lire tous les fichiers de ce format, sans que l'émetteur n'ait à s'angoisser sur les éventuelles variantes acceptées ou pas (cf. section 4.5).

Après l'arbre standard, le plus important est sans doute l'arbre « vendeur » (notez que le terme inclut les organisations non-commerciales) où se trouvent les types liés à un fournisseur de logiciel particulier. Les sous-types dans cet arbre commencent par la facette `vnd` comme par exemple `application/vnd.amiga.ami`. Le contrôle du changement, cette fois, revient au vendeur en question (bien que l'enregistrement lui-même puisse être fait par un tiers). À noter qu'aucune vérification publique n'est nécessaire (ce sont des types pour un produit privé, après tout) mais qu'une telle vérification est quand même recommandée (sur la liste `media-types`, voir plus loin). Dans tous les cas, la demande sera examinée par l'expert nommé par l'IANA.

Comme indiqué plus haut, « vendeur » (et son abréviation `vnd`) n'implique pas forcément une entreprise commerciale. Par exemple, Debian a enregistré `vnd.debian.copyright` ce qui, au dire de Charles Plessy, le développeur Debian qui s'en est chargé, est désormais très simple <<http://lists.debian.org/debian-devel-announce/2012/12/msg00002.html>> avec la nouvelle procédure de notre RFC.

Et s'il n'y a pas du tout de vendeur, même pas une fondation ou association établie ? Si Jean Durand, développeur avec un compte sur GitHub, veut un type de contenu pour le format d'images qu'il a développé ? Pas de panique, Jean peut utiliser l'arbre « personnel ». Son sous-type aura alors la facette `prs` comme dans `application/prs.plucker`. Comme pour l'arbre « vendeur », l'examen public sur la liste `media-types` est recommandé mais pas indispensable.

Enfin, il y avait un arbre « privé » avec la facette `x`. Il n'y avait pas d'enregistrement dans cet arbre, chacun pouvait y mettre ce qu'il voulait avec zéro formalité. Ces préfixes `X` ayant créé quelques problèmes (collisions entre deux noms, noms qui deviennent populaires et qui sont alors « enregistrés de fait »), ils sont aujourd'hui abandonnés (RFC 6648) et l'utilisation de l'arbre `x` très découragée.

Bon, maintenant, je peux l'envoyer, ma demande à l'IANA, se demande notre programmeur qui veut enregistrer « son » type ? Pas encore, il faut encore lire toute la section 4 qui précise ce qu'on peut et qu'on ne peut pas enregistrer. Par exemple, le type doit désigner un **format** pas des trucs qui ressemblent à un format mais n'en sont pas comme un encodage de transfert. C'est pour cela qu'on ne trouvera pas Base64 dans le registre des types de contenu.

Les noms doivent évidemment obéir à la syntaxe standard (section 4.2) et `audio/%#&` n'est donc pas acceptable. C'est d'autant plus vrai que certains protocoles qui utilisent les types de contenu peuvent avoir des règles strictes sur les caractères acceptés.

À noter que le sous-type peut avoir une structure : ainsi, les sous-types XML peuvent s'écrire `FORMAT+xml` comme par exemple `application/svg+xml` pour SVG. Cette structure (définie dans le RFC 6839, et comportant un format spécifique, un plus et un suffixe qui indique le format général) permet de garder toute l'information. Un logiciel générique XML peut ainsi faire quelque chose d'un fichier SVG, même s'il ne connaît pas SVG.

Depuis, d'autres formats ont suivi ce système (on trouve des sous-types se terminant par `+json` pour JSON) et un registre de ces formats génériques `<https://www.iana.org/assignments/media-type-structured-media-type-structured-suffix.xml>` a été créé par le RFC 6839.

Il y a ensuite plein de règles spécifiques à certains types. Ainsi, le type `text` est conçu pour du contenu qui est lisible par un humain, sans disposer d'un logiciel particulier. Pour le logiciel qui reçoit un document, c'est une distinction importante : un document `text/UNKNOWN` où `UNKNOWN` est un sous-type inconnu, pourra toujours être montré à l'utilisateur et ce sera mieux que rien. Avec les images (`image/SOMETHING`) ou le son (`audio/SOMETHING`), cela n'aurait pas de sens.

C'est ainsi que `text/troff` (RFC 4263) désigne du texte formaté avec `troff` ; si ce format d'enrichissement prévoit quelques « décorations » au texte, il reste néanmoins lisible sans processeur `troff`.

Le texte brut, sans aucun caractère spécial, est `text/plain` (RFC 2046). Tous les sous-types de `text/` peuvent avoir un paramètre supplémentaire (voir la section 4.3 sur le concept de paramètres), `charset`, qui indique à la fois un jeu de caractères et un encodage (le terme de "*charset*" est donc malheureux mais il est trop tard pour le changer). Les "*charsets*" sont eux-même enregistrés en suivant le RFC 2978. Attention, certains sous-types contiennent l'indication de "*charset*" dans le texte lui-même (c'est le cas de XML) et ne doivent donc pas utiliser le paramètre `charset` (cf. RFC 6657). Dans tous les cas, s'il y a un paramètre `charset`, il doit dans la plupart des cas désormais être obligatoire (pas de valeur implicite). Si, malgré tout, une valeur par défaut est définie, cela doit être UTF-8 (RFC 6657 qui a remplacé l'ancienne règle, qui privilégiait US-ASCII).

`image/` et `audio/` ne posent pas de problèmes particuliers. Par contre, pour `video/`, le RFC rappelle qu'une vidéo inclut souvent du contenu d'autres types (par exemple une piste audio synchronisée) et que cela doit être pris en compte lorsqu'on enregistre un sous-type.

Un type bien plus général, `application/`, est utilisé pour tout le reste. Ce sont les formats spécifiques à une application, par exemple `application/msword` pour le format privé de Microsoft Word, que tant de gens s'obstinent à vous envoyer sans s'être demandé si vous aviez payé votre taxe à Steve Ballmer. Mais `application/` sert aussi pour les cas où le format n'est pas lisible directement (contrairement à `text/`) mais ne rentre pas dans les catégories comme `image/` ou `audio/`. C'est ainsi qu'il existe un `application/pdf` pour les fichiers PDF.

Enfin, un type un peu spécial, `multipart/` est là pour le contenu composite (RFC 2046). Si le protocole permet de mettre plusieurs objets de type différents dans un message, celui-ci va être étiqueté `multipart/QUELQUECHOSE`. On verra alors `multipart/signed` (deux parties, le contenu lui-même et une signature, par exemple en PGP), `multipart/mixed` (plusieurs parties distinctes, chacune ayant son type et sous-type), `multipart/alternative` (plusieurs parties de sous-types différents mais normalement équivalentes, le logiciel choisira le format qu'il préfère), etc.

Le RFC prévoit la possibilité d'enregistrer de nouveaux types (et pas seulement des sous-types). Il faudra pour cela un RFC sur le chemin des normes et notre RFC 6838 prévoit qu'un tel enregistrement sera plutôt rare. Un exemple est la création du type `font/` par le RFC 8081.

Est-ce qu'un sous-type peut être enregistré pour un format breveté? Oui (autrement, on n'aurait pas `audio/mpeg`), mais en suivant les règles habituelles de l'IETF, notamment de divulgation des brevets connus par un participant (RFC 8179 et RFC 5378).

Il faut aussi penser à la sécurité (section 4.6). La demande d'enregistrement d'un type de contenu doit comporter une section sur la sécurité de ce format, indiquant quels sont les risques connus. Quels formats peuvent créer des problèmes? Ceux où le format est en fait un langage complet, permettant des actions comme ouvrir un fichier, sont les principaux. C'est ainsi qu'`application/postscript` a fait souvent parler de lui (voir son analyse de sécurité détaillée dans la section 4.5.2 du RFC 2046). Un tel « contenu actif » est toujours risqué.

Une fois qu'on a digéré toutes ces règles, ainsi que celles qui se trouvent dans le RFC et que je n'ai pas reprises ici, on peut passer à la rédaction de sa demande d'enregistrement d'un sous-type. La section 5 expose les procédures. La méthode canonique (mais il existe des variantes) est de rédiger sa demande (le gabarit, qui précise quelles informations doivent être données, figure en section 5.6), puis la soumettre d'abord à la liste `media-types@ietf.org` pour avoir un avis de la communauté. Ensuite, si l'enregistrement est dans l'arbre standard, on envoie la demande à l'IANA qui vérifiera que le format est bien décrit dans une norme stable (un RFC ou son équivalent dans une autre SDO). Pour les autres arbres, on écrit à l'IANA, ou bien on utilise le formulaire Web <https://www.iana.org/cgi-bin/mediatypes.pl>. Les demandes seront ensuite examinées par l'expert désigné, actuellement Ned Freed, avec Mark Baker en secours. Au cas où on n'aimerait pas sa décision, on peut toujours tenter un appel auprès de l'IESG.

Une autre procédure, en section 6, décrit le mécanisme d'enregistrement des suffixes de sous-types (comme `+xml` ou `+json`, voir le registre <https://www.iana.org/assignments/media-type-structured-media-type-structured-suffix.xml>), créé par le RFC 6839).

La nouvelle liste de discussion pour un premier examen des demandes d'enregistrement a été créée le 17 septembre 2012. Elle se nomme `media-types@ietf.org` et remplace l'ancienne `ietf-types`.

Les amateurs d'histoire peuvent lire la section 1.1 de ce RFC : elle rappelle la naissance de ces types de contenu, dans le seul contexte du courrier électronique, lorsque celui-ci est devenu multimédia, en 1992, avec le RFC 1341. Notez que le courrier est **asynchrone**. Cela veut dire qu'il n'est pas possible de négocier un type de données entre les deux machines. L'émetteur doit lancer son message sans connaître le récepteur et étiqueter correctement le contenu est donc crucial. Pour limiter les risques d'interopérabilité (l'émetteur choisit un type très rare et le récepteur ne le connaît pas), le choix avait été fait de limiter sévèrement le nombre de types, par le biais d'une procédure d'enregistrement très restrictive (RFC 2048, puis son successeur le RFC 4288, voir aussi la section 4.9 de notre RFC 6838). Aujourd'hui, les types sont souvent utilisés dans des protocoles synchrones comme HTTP, où la négociation est possible (le serveur HTTP sait ce qu'accepte le client, grâce à l'en-tête `Accept` :), ces anciennes règles n'ont donc plus de sens. D'où la libéralisation que marque notre nouveau RFC.

Outre celle-ci, les grands changements depuis le RFC 4288 sont (cf. annexe B) :

- Spécification formelle des suffixes de sous-types (comme `+xml`) et création d'un registre de ces sous-types,
- Changement des règles pour les noms commençant par `x-`, suite au RFC 6648,
- Mise à jour des règles pour le type `text/` pour tenir compte du RFC 6657.