RFC 8187 : Indicating Character Encoding and Language for HTTP Header Field Parameters

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 septembre 2017. Dernière mise à jour le 16 septembre 2017

Date de publication du RFC : Septembre 2017

https://www.bortzmeyer.org/8187.html

Les requêtes et réponses du protocole HTTP incluent des **en-têtes** (comme `User-Agent:` ou `Content-Disposition:` avec des **valeurs**, qui, il y a longtemps, ne pouvaient se représenter directement qu'avec les caractères du jeu ISO 8859-1, voire seulement avec ASCII (c'était compliqué). Comme MIME, dans le RFC 2231 [1], prévoyait un mécanisme très riche pour encoder les en-têtes du courrier électronique, ce RFC 8187 réutilise ce mécanisme pour HTTP (il remplace le RFC 5987, qui avait été le premier à le faire). Pour le corps du message (voir par exemple le RFC 7578), rien ne change.

Cette ancienne restriction à Latin-1 (qui n'est plus d'actualité) vient de la norme HTTP, le RFC 2616, dans sa section 2.2, qui imposait l'usage du RFC 2047 pour les caractères en dehors de ISO 8859-1. Le RFC 7230 a changé cette règle depuis (sa section 3.2) mais pas dans le sens d'une plus grande internationalisation (ISO 8859-1 ne convient qu'aux langues européennes), plutôt en supprimant le privilège d'ISO 8859 et en restreignant à ASCII. Et il ne précise pas vraiment comment faire avec d'autres jeux de caractère comme Unicode. Il ne reste donc que la solution du RFC 2231.

Notre nouveau RFC peut être résumé en disant qu'il spécifie un **profil** du RFC 2231. Ce profil est décrit en section 3, qui liste les points précisés par rapport au RFC 2231. Tout ce RFC n'est pas utilisé, ainsi le mécanisme en section 3 du RFC 2231, qui permettait des en-têtes de plus grande taille, n'est pas importé (section 3.1 de notre RFC).

En revanche, la section 4 du RFC 2231, qui spécifiait comment indiquer la langue dans laquelle était écrite la valeur d'un en-tête est repris pour les paramètres dans les en-têtes. Ainsi, (section 3.2), voici un en-tête (imaginaire : `Information:` n'a pas été enregistré), avec un paramètre `title` traditionnel en pur ASCII :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc2231.txt`

```
Information: news; title=Economy
```

et en voici un avec les possibilités de notre RFC pour permettre les caractères [Caractère Unicode non montré [2]] et € (≪ *"Sterling and euro rates"* ≫) :

```
Information: news; title*=UTF-8''%c2%a3%20and%20%e2%82%ac%20rates
```

Par rapport au RFC 2231 (qui était silencieux sur ce point), un encodage de caractères est décrété obligatoire (c'est bien sûr UTF-8), et il doit donc être géré par tous les logiciels. La mention de l'encodage utilisé est également désormais obligatoire (section 3.2 de notre RFC). La langue elle-même est indiquée par une **étiquette**, selon la syntaxe du RFC 5646. Du fait de ces possibilités plus riches que celles prévues autrefois pour HTTP, les paramètres qui s'en servent doivent se distinguer, ce qui est fait avec un astérisque avant le signe égal (voir l'exemple ci-dessus). Notez que l'usage de l'astérisque n'est qu'une convention : si on trouve un paramètre inconnu dont le nom se termine par un astérisque, on ne peut pas forcément en déduire qu'il est internationalisé.

La valeur du paramètre inclut donc le jeu de caractères et l'encodage (obligatoire), la langue (facultative, elle n'est pas indiquée dans l'exemple ci-dessus) et la valeur proprement dite.

Voici un exemple incluant la langue, ici l'allemand (code `de`, la phrase est ≪ *"Mit der Dummheit k[Caractère Unicode non montré]mpfen G[Caractère Unicode non montré]tter selbst vergebens"* ≫, ou ≪ contre la bêtise, les dieux eux-mêmes luttent en vain ≫, tirée de la pièce ≪ La pucelle d'Orléans ≫) :

```
Quote: theater;
    sentence*=UTF-8'de'Mit%20der%20Dummheit%20k%C3%A4mpfen%20G%C3%B6tter%20selbst%20vergebens.
```

La section 4 couvre ensuite les détails pratiques pour les normes qui décrivent un en-tête qui veut utiliser cette possibilité. Par exemple, la section 4.2 traite des erreurs qu'on peut rencontrer en décodant et suggère que, si deux paramètres identiques sont présents, celui dans le nouveau format prenne le dessus. Par exemple, si on a :

```
Information: something; title="EURO exchange rates";
              title*=utf-8''%e2%82%ac%20exchange%20rates
```

le titre est à la fois en ASCII pur et en UTF-8, et c'est cette dernière version qu'il faut utiliser, même si normalement il n'y a qu'un seul paramètre `title`.

Ces paramètres étendus sont mis en œuvre dans Firefox et Opera ainsi que, dans une certaine mesure, dans Internet Explorer.

Plusieurs en-têtes HTTP se réfèrent formellement à cette façon d'encoder les caractères non-ASCII :

— `Authentication-Control:`, dans le RFC 8053 (≪ *"For example, a parameter "username" with the value "Renee of France" SHOULD be sent as username="Renee of France". If the value is "Renée of France", it SHOULD be sent as username*=UTF-8"Ren%C3%89e%20of%20France instead"* ≫),

2. Car trop difficile à faire afficher par LaTeX

— `Authorization:` (pour l'authentification HTTP, RFC 7616, avec également un paramètre `username` pour l'ASCII et `username*` pour l'encodage défini dans ce RFC),
— `Content-Disposition:`, RFC 6266, qui indique sous quel nom enregistrer un fichier et dont le paramètre `filename*` permet tous les caractères Unicode,
— `Link:`, normalisé dans le RFC 5988, où le paramètre `title*` permet des caractères non-ASCII (`title` étant pour l'ASCII pur).

Les changements depuis le RFC 5987, sont expliqués dans l'annexe A. Le plus spectaculaire est le retrait d'ISO 8859-1 (Latin-1) de la liste des encodages qui doivent être gérés obligatoirement par le logiciel. Cela fera plaisir aux utilisateurs d'Internet Explorer 9, qui avait déjà abandonné Latin-1. Autrement, rien de crucial dans ces changements. Le texte d'introduction a été refait pour mieux expliquer la situation très complexe concernant la légalité (ou pas) des caractères non-ASCII dans les valeurs d'en-tête.

Si vous voulez voir un exemple, essayez de télécharger le fichier `http://www.bortzmeyer.org/files/foobar.tx`
Si votre client HTTP gère l'en-tête `Content-Disposition:` **et** le paramètre internationalisé `filename*`, le fichier devrait être enregistré sous le nom `f[Caractère Unicode non montré ]bàr.txt`.La configuration d'Apache pour envoyer le `Content-Disposition:` est :

```
<Files "foobar.txt">
    Header set Content-Disposition "attachment; filename=foobar.txt; filename*=utf-8''f%%C3%%B6b%%C3%%A0r.txt"
</Files>
```

Par exemple, Safari ou Firefox enregistrent bien ce fichier sous son nom international.

Ah, et puisque ce RFC parle d'internationalisation, on notera que c'est le **premier** RFC (à part quelques essais ratés au début) à ne pas comporter que des caractères ASCII. En effet, suivant les principes du RFC 7997, il comporte **cinq** caractères Unicode : dans les exemples (« *"Extended notation, using the Unicode character U+00A3 ("[Caractère Unicode non montré]", POUND SIGN)"* » et « *"Extended notation, using the Unicode characters U+00A3 ("[Caractère Unicode non montré]", POUND SIGN) and U+20AC ("€", EURO SIGN)"* »), dans l'adresse (« *"Münster, NW 48155"* ») et dans les noms des contributeurs (« *"Thanks to Martin Dürst and Frank Ellermann"* »).