# Décoder les en-têtes du courrier électronique

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 10 avril 2006

https://www.bortzmeyer.org/decoder-en-tetes-courrier.html

---

Ma langue maternelle étant le français, je reçois une bonne proportion de messages électroniques dans cette langue, avec des caractères composés comme ç ou ê. La façon dont ils sont encodés pour le voyage dans l'Internet ne me convient pas et je les décode à l'arrivée. Voici comment.

Le RFC 2822 [1], qui normalise la représentation des messages « sur le câble » précise que les en-têtes des messages (donc le sujet, l'expéditeur, etc) doivent être en US-ASCII uniquement. Pour pouvoir représenter des prénoms comme le lien ou bien des sujets avec caractères composés, le RFC 2047, qui fait partie de la série sur MIME, précise qu'ils doivent être encodés. Ainsi, `From: Stéphane Bortzmeyer` devient typiquement `From: =?ISO-8859-1?Q?St=E9phane?= Bortzmeyer`.

Lire ces messages n'est pas un problème, tous les MUA modernes savent le lire et l'afficher proprement (j'utilise mutt).

Mais j'utilise mon courrier électronique pour bien d'avantage qu'une lecture rapide, suivie d'une poubellisation : c'est ma bibliothèque, mon outil de travail, une source d'information et une référence. Notamment, il faut absolument que je puisse chercher dans ce corpus de messages. Et, comme je suis un grand partisan de la séparation des outils (un MUA gère le courrier, il n'a pas besoin de concurrencer les outils de recherche), j'utilise l'excellent grepmail <http://grepmail.sourceforge.net/>.

Je dois donc décoder ces en-têtes avant de les stocker sur ma machine, dans le jeu de caractères que j'utilise (Latin-1 puisque je ne suis pas encore passé à UTF-8 <https://www.bortzmeyer.org/pas-encore-utf8.html>).

Le programme pour décoder est un script Python trivial, qui n'utilise que la bibliothèque standard du langage :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2822.txt

```
#!/usr/bin/python

import email.Header
import sys

header_and_encoding = email.Header.decode_header(sys.stdin.readline())
for part in header_and_encoding:
    if part[1] is None:
        print part[0],
    else:
        upart = (part[0]).decode(part[1])
        print upart.encode('latin-1'),
print
```

Et, pour extraire du message ce que je veux décoder, j'utilise l'outil formail, qui fait partie de procmail. formail extrait le champ indiqué :

```
SUBPYTHON=`formail -czxSubject: | bin/rewrite-email.py`

:0fw
| formail -i"Subject: $SUBPYTHON"
```

Et voilà, le courrier est stocké dans ma boîte aux lettres comme si c'était un fichier texte que j'ai édité, dans l'encodage qui convient à mes outils.