

Arrêter d'interdire des adresses de courrier légales

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 4 Octobre 2006. Dernière mise à jour le 13
Décembre 2009

<http://www.bortzmeyer.org/arreter-d-interdire-des-adresses-legales.html>

Combien de fois ai-je vu une adresse de courrier parfaitement légale être refusée par un formulaire d'inscription quelconque, en général avec une remarque désobligeante, du genre "ERREUR : Votre adresse email est invalide". À quoi est-ce dû ? Au fait que le plupart du temps, les programmeurs qui ont écrit cette vérification ne connaissent pas les règles de syntaxe des adresses de courrier et ne se rendent pas compte qu'il en existe plus de variétés qu'ils ne l'imaginent.

Lorsqu'un formulaire sur le Web vous demande d'indiquer une adresse de courrier électronique, le programme situé derrière vérifie souvent la syntaxe de ladite adresse. Bien sûr, le mieux serait de tester l'adresse, en y envoyant un mot de passe, par exemple. C'est ce que font en général les formulaires d'abonnement à une liste de diffusion. Si cette méthode est un très bon test (si le message est reçu, c'est, par définition, que l'adresse est valide), elle est plus complexe à programmer, nécessite de garder un état (ce qui a été envoyé, ce qui a été confirmé), et elle peut ennuyer le destinataire.

Aussi, le test se limite souvent à la syntaxe : on vérifie, par exemple (**attention**, je prends exprès un exemple complètement faux) que l'adresse correspond à l'expression rationnelle `[A-Za-z0-9_-\.\.]+@[A-Za-z0-9_-\.\.]+`, qui autorise `postmaster@example.org` mais pas `stephane+blog@bortzmeyer.org` qui est pourtant parfaitement légale.

J'utilise en général des "adresses plus" comme `stephane+blog@bortzmeyer.org` car elles permettent facilement de trier son courrier (on la retrouve dans l'en-tête `Delivered-To`). Le MTA est configuré pour délivrer à la boîte dont le nom précède le signe plus (en Sieve, cela se fait avec le RFC 5233¹) et on peut donc créer sans formalité une infinité d'adresses. À chaque formulaire que je remplis, je laisse une adresse différente, comportant le nom de l'organisation qui gère le formulaire, ce qui me permet de savoir ensuite d'où vient l'adresse par laquelle on me joint. Ce n'est pas forcément le signe plus

1. Pour voir le RFC de numéro NNN, <http://www.ietf.org/rfc/rfcNNN.txt>, par exemple <http://www.ietf.org/rfc/rfc5233.txt>

qui est utilisé, d'ailleurs (le MTA Postfix permet de configurer ce caractère avec l'option `recipient_delimiter`, MMDFutilisait le signe égal, qui a les mêmes problèmes).

Mais le résultat est que je me fais jeter par de nombreux sites Web. Par exemple, je cherchais à louer un serveur dédié Kimsufi <<http://www.kimsufi.com/>> pour l'hébergement de ce blog mais je n'ai jamais réussi à me créer un compte client chez OVH <<http://www.ovh.net/>>, le fournisseur, car celui-ci s'obstine à refuser mon adresse. J'ai finalement choisi SliceHost <<http://www.bortzmeyer.org/slicehost-debut.html>> qui avait pile la même bogue, mais qui l'a corrigée en quelques heures (alors que, la plupart du temps, les rapports de bogue sur ce sujet sont ignorés).

Le manque de main d'œuvre pour le développement (une technologie comme XForms ou comme les formulaires de HTML 5 <<http://blog.oldworld.fr/index.php?post/2010/11/17/HTML5-Forms-Valid>> simplifiera peut-être les choses dans le futur) fait que les programmes de vérification des données saisies, qu'ils soient écrits en Javascript, PHP ou un autre langage, sont en général assez bâclés. Il est exceptionnel que leur auteur vérifie la norme qui régit la syntaxe des adresses de courrier, le RFC 5322, section 3.4. À leur décharge, il faut préciser que ladite norme est redoutable. Par exemple, toutes les adresses suivantes sont valides :

- {tropdur}@example.org
- c&a@hotmail.com
- directeur@arts-premiers.museum (j'ai vu une fois un code Javascript qui vérifiait que le TLD avait moins de quatre lettres, ce qui est absurde vue l'existence de .museum ou .travel)
- "Stephane[Bortzmeyer]"@laposte.net

Alors, quel conseil donner au programmeur ? D'abord, il faut bien voir qu'il a très peu de chances de vérifier une adresse par une simple expression rationnelle. Certes, Tom Christiansen l'a fait <http://www.cpan.org/authors/Tom_Christiansen/scripts/ckaddr.gz> mais tout le monde n'est pas Tom Christiansen (lisez la très intéressante page de test des expressions <<http://fightingforalostcause.net/misc/2006/compare-email-regex.php>>). La syntaxe des adresses est décrite par une grammaire, pas par une expression rationnelle. Si on veut analyser tous les cas, il faut utiliser un vrai analyseur syntaxique. N'est-ce pas trop lourd pour une simple et vague vérification ? Dans ce cas, une approche raisonnable est de ne pas chercher à vérifier tous les cas mais de se contenter d'une vérification légère. Par exemple, le RFC 4287, qui normalise le format de syndication Atom a choisi une méthode simple et radicale : `atomEmailAddress = xsd:string { pattern = ".+@.+" }`. Cette expression est trop laxiste (elle accepte "foo bar[@truc.machin) mais elle attrape quand même une partie des erreurs, en ne refusant aucune adresse légale (ce que je trouve pire que d'accepter des adresses illégales). (Les pros des expressions rationnelles - comme Pascal Courtois qui m'a signalé celle-ci - auront noté que, si le moteur d'expressions est gourmand - "greedy", il faut réécrire l'expression en `.+?@.+`).

Pour PHP, je recommande très chaudement un excellent article de Douglas Lovell dans le Linux Journal, "Validate an E-Mail Address with PHP, the Right Way" <<http://www.linuxjournal.com/article/9585>>, qui décrit très bien la difficulté du problème et propose des solutions. Je partage notamment son inquiétude que « "There is some danger that common usage and widespread sloppy coding will establish a de facto standard for e-mail addresses that is more restrictive than the recorded formal standard." »

J'ai parlé dans cet article du cas de formulaires Web d'enregistrement, qui vous demandent votre adresse de courrier. Mais Emmanuel Haguet me signale qu'il y a plus fort : des "webmails" comme celui d'Orange qui refusent d'envoyer du courrier aux adresses qu'ils jugent, bêtement, non standards.

Quelques autres références intéressantes :

- Un bon validateur en ligne <<http://simonslick.com/VEAF/>> qui semble complètement conforme au RFC.
- "What is the best regular expression for validating email addresses?" <<http://stackoverflow.com/questions/201323/what-is-the-best-regular-expression-for-validating-email-add>> sur StackOverflow <<http://www.bortzmeyer.org/stack-overflow.html>>.

- (Transmis par Pierre Beyssac) Un bon article de Derrick Pallas <http://worsethanfailure.com/Articles/Validating_Email_Addresses.aspx> sur WTF, avec plein de commentaires intéressants <http://worsethanfailure.com/Comments/Validating_Email_Addresses.aspx>
- Pour PHP, un outil de validation des adresses <<http://www.php.net/manual/fr/filter.filters.validate.php>> qui **semble** correct. C'est certainement mieux que les bricolages maison des neuneus PHP typiques.
- Un autre bon article, par Sinjo <<http://blog.sinjakli.co.uk/2011/02/13/email-address-validation->>, avec d'intéressants commentaires.
- Un exemple **erroné** pour PostgreSQL, par un auteur qui s'y connaît davantage en expressions rationnelles qu'en courrier électronique <<http://binodsblog.blogspot.com/2011/02/regular-expression.html>>.

J'espère que cet article aura convaincu quelques auteurs de formulaires d'accepter toutes les adresses de courrier légal. Espérons. En attendant, je publie régulièrement les noms <<http://www.bortzmeyer.org/publier-ral.html>> de ceux qui sont mal configurés.