Format pour transmettre des données structurées sur le réseau

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 juin 2009

https://www.bortzmeyer.org/format-pour-donnees-structurees.html

---

La question du meilleur format pour transmettre des données structurées d'une machine du réseau à une autre a toujours suscité bien des discussions. Ce n'est peut-être pas aussi passionnel que le choix du langage de programmation <https://www.bortzmeyer.org/choix-langage-prog.html> mais cela fait toujours largement débat.

Le débat n'est pas toujours très informé. Sur les forums, on voit souvent, lorsque quelqu'un s'enquiert du « meilleur » format pour transmettre des données structurées, des réponses aussi idiotes que « XML est nul [sans autre explication], utilises JSON » ou bien « Il faut utiliser les Protocol Buffers [là aussi, sans explication, à part que c'est le protocole de Google et que Google a forcément raison] ». (Données « structurées » signifie qu'elles ne sont pas simplement des listes de tuples, donc un format comme CSV (RFC 4180 [1]) ne compte pas.)

Une discussion bien meilleure s'est déroulée récemment sur une liste de diffusion de l'IETF, apps-discuss, la liste du secteur Applications de l'IETF. On trouve son point de départ dans un article de Patrik F[Caractère Unicode non montré [2]]ltstr[Caractère Unicode non montré]m <http://www.ietf.org/mail-archive/web/apps-discuss/current/msg00582.html>. L'IETF n'a pas de format standard pour ses applications, SNMP (RFC 1157) utilise ASN.1, EPP (RFC 4930) et Netconf (RFC 6241) XML, etc.

Pour ceux qui n'ont pas participé aux N discussions précédentes sur le même sujet, voici une liste (que je crois exhaustive) des candidats au rôle de format idéal, ainsi que quelques notes personnelles, et qui assument leur subjectivité, sur chacun. Tous disposent d'une documentation, parfois d'une vraie norme ouverte, et tous ont des mises en œuvre en logiciel libre :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4180.txt
2. Car trop difficile à faire afficher par LaTeX

— XML, normalisé (par le W3C), très répandu, beaucoup de mises en œuvres en logiciel libre dans tous les langages, beaucoup d'experts disponibles, gère proprement Unicode depuis le début.
— JSON, le seul, semble t-il, qui aie fait l'objet d'un RFC, le RFC 7159. Assez répandu dans le monde Web 2.0.
— YAML, plus répandu pour des fichiers (par exemple des fichiers de configuration) locaux, je ne crois pas l'avoir souvent rencontré sur le réseau.
— Les S-expressions, popularisées par le langage Lisp et donc appréciées par les fans de ce langage.
— ASN.1, d'origine ISO et donc, à juste titre, mal aimé dans le monde Internet. ASN.1 permet de décrire les données, l'encodage exact sur le câble étant laissé à des formats compagnons comme BER ou DER.
— Les netstrings, conçu par l'inénarrable djb et qui plait donc à ses fans.
— Les Protocol Buffers, un des plus récents, mais qui a derrière lui tout le poids du géant Google.
— Le système BERT <http://bert-rpc.org/>, introduit par Github <http://github.com/blog/531-introducing-bert-and-bert-rpc> et basé sur Erlang (mais utilisable depuis d'autres langages).
— On peut citer aussi les formats spécifiques à un langage de programmation donné, qui permettent de sérialiser des objets du langage pour les transporter sur le réseau comme `pickle` pour Python ou `java.io.Serializable` pour Java.
— Et, enfin, les informaticiens étant ce qu'ils sont, il y a toute la cohorte des formats conçus et mis en œuvre localement par un programmeur ou bien une petite équipe...