

Suivi - Tribune : Backend TSV pour la tribune

#1634 Posté par [devnewton \(page perso\)](#) le 19/07/16 à 15:22. État de l'entrée : ouverte [Licence CC by-sa](#)
Tags : [letsmakelinuxfrgreatagain](#), [tribune](#), [bouchot](#), [tsv](#), [nonojeteveux](#)

Objectif

Proposer une alternative simple et légère au backend xml.

Spécification

Le backend TSV:

1. est encodé en est UTF-8
2. est servi en text/tab-separated-values
3. les tags du message ne sont pas XML-encodés
4. n'a pas d'entête : son format est fixe
5. a pour url: <https://linuxfr.org/board/index.tsv>
6. a pour format:

```
 ${id}\t${time}\t${info}\t${login}\t${message}\n
```

Note: les caractères de contrôle sont à bannir du contenu de chaque champ. Note 2: les id doivent être croissant, du plus ancien au plus récent.

question

Posté par [killer](#) le 19/07/16 à 15:32. Évalué à 2 (+0/-0).

Ca manque de finesse dans les spécifications.

- o Comment seraient encodés les retours à la ligne ou les tabulations dans les champs message ?
- o C'est censé répondre à quel problème ?
- o C'est quoi le problème avec le XML ?
- o Pourquoi TSV et pas JSON, qui est largement plus supporté ?
- o Quels sont les tags qui doivent être interprétés par les clients ?
- o La balise a est-elle utilisée pour les urls ?
- o **Pourquoi ?**

Re: question

Posté par [Stein Straßenbahn-Hohe \(page perso\)](#) le 19/07/16 à 18:48. Évalué à 3 (+0/-0).

Comment seraient encodés les retours à la ligne ou les tabulations dans les champs message ?

Pas. Comme un message de tribune est rendu comme du html, les \n, \t et espaces, ou toute chaîne constituée de ces caractères en nombre et combinaison quelconques, sont rendus par un espace simple. Par conséquent c'est au moteur de tribune de "sanitizer" le message en ce sens dans le backend tsv, comme il le fait déjà pour les tags.

C'est censé répondre à quel problème ?

C'est quoi le problème avec le XML ?

Pourquoi TSV et pas JSON, qui est largement plus supporté ?

Tu arrives longtemps après le débat, mais c'est vrai que tu ne fréquentes plus les tribunes où ces discussions ont lieu.

La réponse à tout cela est simplement "simplicité, légèreté et robustesse". Pourquoi se lancer dans un parsing complexe quand un split("\n") suffit à récupérer les posts et un split("\t") les champs du post ?

Le XML, outre qu'il est lourd à parser, comporte la problématique du double niveau de namespace entre les tags du backend et ceux du message, ce qui fait qu'il n'y a pas 2 moteurs de tribune le gérant de façon identique (tags encoded, not encoded, raw, encapsulation CDATA...)

Le JSON est particulièrement adapté pour un client web, mais des coincoins sont toujours développés en client lourd.

Et enfin, le sujet n'est pas d'éliminer les backends XML ou JSON qui ont toujours leur légitimité. Il s'agit de proposer une alternative et de laisser le choix au client.

Quels sont les tags qui doivent être interprétés par les clients ?

La balise a est-elle utilisée pour les urls ?

Hors-sujet, il n'y a pas de changement dans le format du contenu du message (si ce n'est la suppression des \n \t, ce que la plupart des moteurs fait déjà). La demande ne concerne que le format du backend.

Pourquoi ?

Je profite de cette dernière question plutôt large pour aborder d'autres points : on voit que le but premier d'un tel backend est l'efficacité, et celle-ci prend tout son sens lorsque le format TSV est couplé aux autres techniques d'optimisation : renvoi uniquement des posts utiles grâce au paramètre last-id, posts ordonnés par id croissant (permettant au client d'insérer tous les nouveaux posts d'un coup sans trier), support du XPOST pour économiser un get après un post, etc.

Bien cordialement,
Rock Hightram
Mouling Technologies Senior Architect

Note : les commentaires appartiennent à ceux qui les ont postés. Nous n'en sommes pas responsables.