

RFC 6721 : The Atom "deleted-entry" Element

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 septembre 2012

Date de publication du RFC : Septembre 2012

<https://www.bortzmeyer.org/6721.html>

Le format Atom, normalisé dans le RFC 4287¹, est surtout utilisé pour la syndication de contenus Web. Dans ce cas, si une entrée d'un flux Atom n'est **plus** publiée (remords de l'auteur ou autre raison), le client Atom n'avait pas de moyen de le savoir. Désormais, avec l'extension de ce RFC, surnommée « pierre tombale », on peut indiquer que l'absence d'une entrée est délibérée.

Rappelez-vous qu'un flux de syndication comporte un certain nombre d'entrées mais rarement la totalité des articles d'un site Web (dans le flux de ce blog, par défaut, les articles des quatre derniers mois). On ne peut donc pas dire, lorsqu'un article n'a pas d'entrée dans le flux Atom, si l'article a été supprimé ou simplement s'il n'a pas été inclus dans le flux. Si, par exemple, l'entrée a été gardée dans la mémoire locale du client de syndication, elle risque d'y rester même si l'article original a disparu du site.

Pour éviter cela, ce RFC ajoute un nouvel élément XML à Atom. Il est défini avec la syntaxe Relax NG (l'espace de noms XML abrégé en `at` est <http://purl.org/atompub/tombstones/1.0>):

```
deletedEntry =
  element at:deleted-entry {
    atomCommonAttributes,
    attribute ref { atomUri },
    attribute when { atomDateConstruct },
    ( element at:by { atomPersonConstruct }?
      & element at:comment { atomTextConstruct }?
      & element atom:link { atomLink }*
      & element atom:source { atomSource }?
      & anyElement* )
  }
```

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc4287.txt>

On note donc que la référence de l'article détruit (`ref`) est obligatoire. Elle correspond à l'attribut `id` d'un flux Atom, qui a la forme d'un URI. Ainsi, cet article que vous êtes en train de lire a l'id `tag:bortzmeyer.org,2006-02:Blog/6721` et une pierre tombale, si je supprimais cet article, comporterait `<at:deleted-entry ref="tag:bortzmeyer.org,2006-02:Blog/6721" ...`

Comme l'attribut `ref`, l'attribut `when` est obligatoire et contient la date et l'heure de destruction (au format du RFC 3339). Il est particulièrement utile si une entrée est détruite puis recréée : en examinant les dates, un client Atom peut savoir quel est l'évènement le plus récent (la suppression ou la création) et ignorer les autres.

Les autres attributs de l'élément `at:deleted-entry` sont facultatifs. C'est le cas par exemple de `comment` qui permet d'ajouter une explication en texte libre. Voici, tiré du RFC, un exemple minimal :

```
<at:deleted-entry
  ref="tag:example.org,2005:/entries/1"
  when="2005-11-29T12:11:12Z"/>
```

et un exemple plus complet :

```
<at:deleted-entry
  ref="tag:example.org,2005:/entries/2"
  when="2005-11-29T12:11:12Z">
  <at:by>
    <name>John Doe</name>
    <email>jdoe@example.org</email>
  </at:by>
  <at:comment xml:lang="la">Lorem ipsum dolor</at:comment>
</at:deleted-entry>
```

Notez bien que l'élément `at:deleted-entry` n'est qu'indicatif : on ne peut évidemment pas garantir qu'un client Atom facétieux ne va pas conserver localement (et même redistribuer) des entrées qu'il avait gardé dans son cache et qui ont été détruites dans le flux original.

Autre piège de sécurité : un client Atom ne doit pas croire aveuglément tous les `at:deleted-entry` qu'il rencontrerait sur le Web ! Car un méchant peut toujours publier des pierres tombales pour un autre site que le sien. Il faut donc vérifier l'origine (la pierre tombale doit être dans le même flux que l'article original, ce qui peut poser des problèmes avec les agrégateurs et autres redistributeurs) et/ou vérifier les signatures cryptographiques attachées (avec XML Signature).

Cette extension est mise en œuvre au moins dans AtomBeat <http://code.google.com/p/atombeat/wiki/TombstonesDesign> (voir leurs intéressants exemples d'usage).