

RFC 5733 : Extensible Provisioning Protocol (EPP) Contact Mapping

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 octobre 2009

Date de publication du RFC : Août 2009

<https://www.bortzmeyer.org/5733.html>

Le protocole d'avitaillement EPP ne spécifie pas comment représenter les objets qu'on crée, détruit, modifie, etc. Cette tâche est déléguée à des RFC auxiliaires comme le nôtre, consacré aux **contacts**, c'est-à-dire aux personnes physiques ou morales responsables d'un objet de la base et qui peuvent être **contactées** à son sujet. (L'objet étant typiquement un nom de domaine ou bien un préfixe IP.)

EPP permet à un **client** de créer, détruire et modifier des objets de types différents. En pratique, EPP n'est utilisé que dans l'industrie des noms de domaine mais, en théorie, il pourrait être utilisé pour n'importe quel type d'objets.

Le type n'est donc pas spécifié dans le protocole EPP de base, normalisé dans le RFC 5730¹, mais dans des RFC supplémentaires. Par exemple, celui qui fait l'objet de cet article spécifie le type, la classe (EPP dit le "*mapping*") pour les contacts. Il remplace le RFC 4933, avec très peu de changement, et marque l'arrivée de cette norme au statut de « norme complète », la dernière étape du chemin des normes de l'IETF.

Ce type est spécifié (section 4 du RFC) dans le langage W3C XML Schema.

Un contact est donc composé d'un **identificateur** (type `clIDType` du RFC 5730). Cet identificateur (on l'appelait traditionnellement le "*handle*") est, par exemple, SB68-GANDI (section 2.1).

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5730.txt>

Les contacts ont également un **statut** (section 2.2) qui est toujours mis par le client EPP, typiquement le bureau d'enregistrement. Ce "*mapping*" ne permet pas aux contacts d'être maîtres de leur propre information et de la changer directement (ce qui est cohérent avec l'approche d'EPP où un intermédiaire, le bureau d'enregistrement, a l'exclusivité des changements).

Les contacts ont aussi évidemment des moyens d'être contactés, via numéro de téléphone, adresse postale, etc. Par exemple, l'adresse de courrier du contact est spécifiée en section 2.6. La syntaxe formelle est celle du RFC 5322, par exemple `joe.o'reilly+verisign@example.com`. (Mais le schéma XML a une syntaxe plus bien laxiste, presque tout est accepté.)

Les contacts pouvant être des personnes physiques, pour protéger leur vie privée, la section 2.9 du RFC décrit aussi un format pour spécifier si ces informations doivent être publiées ou pas. Insuffisant pour tous les cas, ce format est en général remplacé, chez les registres européens, par un "*mapping*" spécifique (par exemple, "*EPP parameters for .pl ccTLD*" <<http://www.watersprings.org/pub/id/draft-zygmuntowicz-epp-pltld-03.txt>> pour les polonais qui utilisent un élément <individual> pour indiquer si le contact est une personne physique, et a donc droit à la protection des lois européennes sur les données personnelles).

La section 3 normalise ensuite l'usage des commandes standards d'EPP (comme <create>) pour les objets de notre classe « contact ». À titre d'exemple, voici la réponse d'un serveur EPP à une requête <epp:info> (section 3.1.2) pour le contact SB68-GANDI :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:infData
        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>SB68-GANDI</contact:id>
        <contact:roid>SH8013-REP</contact:roid>
        <contact:status s="clientDeleteProhibited"/>
        <contact:postalInfo type="int">
          <contact:name>John Doe</contact:name>
          <contact:org>Exemple SA</contact:org>
          <contact:addr>
            <contact:street>123 rue de l'Exemple</contact:street>
            <contact:city>Trifouillis-les-Oies</contact:city>
            <contact:cc>FR</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:voice x="1234">+33.7035555555</contact:voice>
        <contact:fax>+33.7035555556</contact:fax>
        <contact:email>jdoe@example.com</contact:email>
        <contact:crDate>1997-04-03T22:00:00.0Z</contact:crDate>
        <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
        <contact:trDate>2000-04-08T09:00:00.0Z</contact:trDate>
        <contact:authInfo>
          <contact:pw>2fooBAR</contact:pw>
        </contact:authInfo>
        <contact:disclose flag="0">
          <contact:voice/>
          <contact:email/>
        </contact:disclose>
      </contact:infData>
    </resData>
  </response>
</epp>
```

L'espace de noms XML pour les contacts est `urn:ietf:params:xml:ns:contact-1.0`.

Comme rappelé par la section 5, EPP utilise XML dont le modèle de caractères est Unicode depuis le début. Logiquement, on devrait donc pouvoir enregistrer des noms et prénoms comportant des accents (comme « Stéphane ») mais je ne suis pas sûr que cela marche avec tous les registres : c'est une chose de transporter la chaîne de caractères Unicode jusqu'au registre et une autre de la stocker dans la base et de la ressortir proprement.

La classe « Contact » permet de représenter certains éléments (comme l'adresse postale) sous deux formes, une en Unicode complet (en indiquant `type="loc"` et l'autre sous une version restreinte à l'ASCII (en indiquant `type="int"`, notez que ces labels doivent être lus à l'envers, la forme restreinte en labelisée `int` pour « internationale » et la forme complète `loc` pour « locale » alors que le contraire aurait été plus logique.)

Ce RFC remplace son prédécesseur, le RFC 4933 mais ce ne sont que des modifications légères, détaillées dans l'annexe A.