

# RFC 7351 : A Media Type for XML Patch Operations

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 31 août 2014

Date de publication du RFC : Août 2014

<https://www.bortzmeyer.org/7351.html>

---

Vous voulez "*patcher*" des documents XML, c'est-à-dire envoyer juste la différence entre deux documents à un serveur pour qu'il applique les opérations de transformation qui vont rendre les deux documents identiques? Le RFC 5261<sup>1</sup> fournissait un format de description de ces "*patches*". Il manquait toutefois un type MIME décrivant ce format. C'est désormais fait, avec la normalisation dans ce RFC du type de média `application/xml-patch+xml`.

Outre le RFC 5261, qui décrivait un format pour les "*patches*" XML, l'un des protocoles qui utilisent le plus XML, HTTP, a une opération `PATCH` qui permet de mettre à jour une ressource Web distante (RFC 5789). HTTP, comme d'autres protocoles de l'Internet, étiquette les ressources envoyées ou récupérées avec un **type de média** (dit aussi « type MIME », cf. RFC 6838), et pour utiliser le format du RFC 5261 avec l'opération `PATCH` du RFC 5789, il fallait un tel type, désormais `application/xml-patch+xml`. (Pour l'opération `PATCH`, les serveurs HTTP choisissent librement les formats de "*patch*" qu'ils gèrent, il n'existe pas de format obligatoire. Ils peuvent accepter `diff`, ce nouvel `application/xml-patch+xml` et/ou bien d'autres.)

Notez que le RFC 5261 définissait un type MIME mais uniquement pour les messages d'erreur, `application/patch-ops-error+xml`.

Outre l'enregistrement du type MIME `application/xml-patch+xml`, ce nouveau RFC corrige également les erreurs du RFC 5261 <[http://www.rfc-editor.org/errata\\_search.php?rfc=5261&rec\\_status=15&presentation=table](http://www.rfc-editor.org/errata_search.php?rfc=5261&rec_status=15&presentation=table)>, liées à la complexité de la combinaison entre XPath et les "*namespaces XML*".

Une des raisons pour lesquelles le RFC 5261 ne définissait pas de type MIME est qu'il ne définissait pas un format complet mais juste les éléments XML indiquant l'ajout, la suppression ou la modification d'éléments XML dans le document cible. Notre nouveau RFC, lui, définit un format complet, où les éléments de "*patch*" sont inclus dans un document dont l'élément racine est `<patch>` (et qui sera donc servi avec le type MIME `application/xml-patch+xml`). Voici un exemple d'un tel document :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5261.txt>

```
<p:patch xmlns:p="urn:ietf:rfc:7351">
  <p:add sel="/section"><para>Nouveau paragraphe, lorem, ipsum, dolor, etc</p:add>
</p:patch>
```

Notez que l'élément à la racine, `<patch>` et les éléments du RFC 5261 comme `<add>` sont dans le même espace de noms `urn:ietf:rfc:7351`. À part ça, on peut reprendre les exemples du RFC 5261 tels quels (voir la section 2.2 de notre nouveau RFC).

L'annexe A contient un certain nombre de détails techniques. D'abord, une discussion détaillée du problème de la correspondance de deux espaces de noms en XML, problème difficile et où le RFC 5261 s'était planté [http://www.rfc-editor.org/errata\\_search.php?eid=3477](http://www.rfc-editor.org/errata_search.php?eid=3477). Ensuite, le problème des espaces de noms dans le document résultat, où les règles ne sont pas les mêmes [http://www.rfc-editor.org/errata\\_search.php?eid=3478](http://www.rfc-editor.org/errata_search.php?eid=3478) en XML et en XPath.

Pour comprendre ce dernier problème, regardez ce document XML :

```
<x xmlns:a="tag:42">
  <y/>
</x>
```

L'espace de noms est le même pour les deux éléments XML (URI `tag:42`, qui utilise de manière inhabituelle le RFC 4151). Mais dans ce document XML quasi-identique :

```
<x xmlns:a="tag:42">
  <y xmlns:a="tag:42"/>
</x>
```

Chaque élément a sa propre déclaration d'espace et de noms et, si le *"patch"* veut modifier `/x/namespace::a`, ce sélecteur XPath ne désigne que le nœud `<x>`, et `<y>` ne doit pas, lui, être modifié. Or, XPath ne fournit pas assez d'informations pour cela, un logiciel de *"patch"* XML doit donc trouver une autre façon de déterminer où stopper sa récursion (par exemple en utilisant DOM). Une mise en œuvre naïve du RFC 5261 qui ne s'appuyerait que sur XPath échouerait dans des cas comme celui indiqué dans ce paragraphe.

Enfin, l'annexe B contient une grammaire ABNF des expressions autorisées par le sous-ensemble de XPath du RFC 5261, pour ceux qui n'aiment pas le W3C Schema qu'utilisait le premier RFC.

Des mises en œuvre complètes de *"XML Patch"* + ce nouveau RFC ? Il y en a apparemment une dans le xDB d'EMC <http://www.emc.com/products/detail/software2/documentum-xdb.htm>.