

## Passerelle - Development #38464

### Toulouse Axel - mapper les types OUINONType OUINONREQUIREDType

12 décembre 2019 14:51 - Lauréline Guérin

<b>Statut:</b>	Fermé	<b>Début:</b>	12 décembre 2019
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Lauréline Guérin	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Les types OUINONType OUINONREQUIREDType doivent accepter:			
<ul style="list-style-type: none"><li>• "oui", "OUI", "True", "true", True, "1", 1</li><li>• "non", "NON", "False", "false", False, "0", 0</li></ul>			

#### Révisions associées

##### Révision 134f74a6 - 17 décembre 2019 18:26 - Lauréline Guérin

toulouse\_axel: bool mapping (#38464)

#### Historique

##### #1 - 13 décembre 2019 10:12 - Lauréline Guérin

- Sujet changé de Toulouse Axel - mapper les types null et OuiNonRequiredType à Toulouse Axel - mapper les types OUINONType OUINONREQUIREDType

- Description mis à jour

##### #2 - 13 décembre 2019 10:26 - Lauréline Guérin

- Fichier 0001-toulouse\_axel-bool-mapping-38464.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

##### #3 - 13 décembre 2019 11:19 - Benjamin Dauvergne

Tu n'as normalement pas le droit d'utiliser (?) c'est de la syntaxe regexp python et pas javascript (

[https://json-schema.org/understanding-json-schema/reference/regular\\_expressions.html](https://json-schema.org/understanding-json-schema/reference/regular_expressions.html)): on va préférer [Oo][Uu][Ii], aussi la regexp match n'import quoi qui contient oui, il faudrait ajouter ^...\$.

##### #4 - 13 décembre 2019 11:23 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Tu n'as normalement pas le droit d'utiliser (?) c'est de la syntaxe regexp python et pas javascript (

[https://json-schema.org/understanding-json-schema/reference/regular\\_expressions.html](https://json-schema.org/understanding-json-schema/reference/regular_expressions.html)): on va préférer [Oo][Uu][Ii], aussi la regexp match n'import quoi qui contient oui, il faudrait ajouter ^...\$.

Aussi j'ai l'impression que tu es trop permissive, dans encode\_bool tu devrais finir par retourner obj pas "", et laisser xmlschema finir la validation.

##### #5 - 13 décembre 2019 15:39 - Lauréline Guérin

- Fichier 0001-toulouse\_axel-bool-mapping-38464.patch ajouté

aussi la regexp match n'import quoi qui contient oui, il faudrait ajouter ^...\$.

Visiblement, d'après les tests que j'ai ajoutés, ce n'est pas nécessaire. D'ailleurs le pattern, avant que je le change, ne contenait pas de marqueurs de début et fin de chaîne.

Aussi j'ai l'impression que tu es trop permissive, dans encode\_bool tu devrais finir par retourner obj pas "", et laisser xmlschema finir la validation.

c'est fait

#### #6 - 17 décembre 2019 14:13 - Thomas Noël

Ne manque-t-il pas la définition de schema\_date\_optional ?

#### #7 - 17 décembre 2019 14:15 - Lauréline Guérin

alors c'était fait exprès mais il faut que je retrouve pourquoi ... je finis de me prendre la tête sur le [#38543](#) et je cherche la raison :

#### #8 - 17 décembre 2019 15:56 - Lauréline Guérin

donc c'est fait exprès:

je ne veux pas passer là-dedans: <https://git.entrouvert.org/passerele.git/tree/passerele/utills/xml.py?h=wip/38464-toulouse-axel-null-and-bool#n147> pour avoir de façon automatique la construction DATEType = DATEREQUIREDType + empty-type sans avoir besoin de définir le type json à la main.

Si je définis le schema OUIINONType, c'est parce que OUIINONType n'est PAS la combinaison de OUIINONREQUIREDType + empty-type dans les xsd fournis ...

Par contre je dois définir les encode/decode des 4 types OUIINON et DATE pour bien passer là-dedans:

<https://git.entrouvert.org/passerele.git/tree/passerele/utills/xml.py?h=wip/38464-toulouse-axel-null-and-bool#n101>

<https://git.entrouvert.org/passerele.git/tree/passerele/utills/xml.py?h=wip/38464-toulouse-axel-null-and-bool#n109>

#### #9 - 17 décembre 2019 17:08 - Thomas Noël

Ok pour schema, effectivement. Cependant si j'ajoute ce test :

```
--- a/tests/test_toulouse_axel_schema.py
+++ b/tests/test_toulouse_axel_schema.py
@@ -46,6 +46,8 @@ def test_date_mapping(date_type):
     json_data = schema.decode(xml_data)
     assert json_data['DATE'] == '2019-12-12'

+     if date_type == 'DATEType':
+         xml_data = schema.encode({'PORTAIL': {'DATE': ''}})
+         assert xml_data.find('DATE').text == ''
+         json_data = schema.decode(xml_data)
+         assert json_data['DATE'] == None
```

Ca crashe dès l'encode parce que " n'est pas une date. Il faut que j'ajoute ceci pour que ça passe :

```
--- a/passerele/contrib/toulouse_axel/models.py
+++ b/passerele/contrib/toulouse_axel/models.py
@@ -55,10 +55,19 @@ class AxelSchema(JSONSchemaFromXMLSchema):
     'pattern': '[0-9]{4}-[0-9]{2}-[0-9]{2}',
     }

+     def encode_date_optional(self, obj):
+         if not obj:
+             return ''
+         return self.encode_date(obj)
```

Je me dis que mon test est faux, que ce genre de XML n'est pas attendu ...? (Oui, je suis dans la vase, sans doute)

#### #10 - 17 décembre 2019 18:26 - Lauréline Guérin

- Fichier 0001-toulouse\_axel-bool-mapping-38464.patch ajouté

Exact, merci :)

J'ai étoffé un peu les tests de date

#### #11 - 18 décembre 2019 00:21 - Thomas Noël

Bon c'est sans doute encore un peu mon esprit tordu mais j'avoue ne pas comprendre que :

```
xml_data = schema.encode({'PORTAIL': {'DATE': 'foobar'}})
assert xml_data.find('DATE').text == 'foobar'
```

ne plante pas. Ca veut dire qu'on peut construire un XML foireux, qui ne suit pas le format de date imposé par le schéma ?

Finalement j'ai un peu le même soucis avec les tests sur bool: FOOBAR, 42, OUIFOOBAR ou FOONONBAR, ça devrait pas passer ?...

Bon, sans doute que c'est que dans ma tête embrumée et que tout ça est sans aucune importance dans l'usage exact qui sera fait de ce principe dans le connecteur... j'arrive pas encore bien à le percevoir. L'impression que je pinaille pour rien du tout.

**#12 - 18 décembre 2019 09:06 - Lauréline Guérin**

Benjamin Dauvergne a écrit :

Aussi j'ai l'impression que tu es trop permissive, dans encode\_bool tu devrais finir par retourner obj pas "", et laisser xmlschema finir la validation.

voila pourquoi je renvoie un truc foireux :)

**#13 - 18 décembre 2019 10:50 - Thomas Noël**

- Statut changé de Solution proposée à Solution validée

Ok bon ben j'ai trop rien compris et go (dans notre usage je pense qu'on s'en fiche, on va d'abord valider le json reçu avant de construire un xml, bref)

**#14 - 18 décembre 2019 15:34 - Lauréline Guérin**

- Statut changé de Solution validée à Résolu (à déployer)

```
commit 134f74a625b7ccd67c40f17fe9fba2d58843c21
Author: Lauréline Guérin <zebuline@entrouvert.com>
Date: Thu Dec 12 16:20:41 2019 +0100
```

```
toulouse_axel: bool mapping (#38464)
```

allez hop :)

**#15 - 23 décembre 2019 18:15 - Frédéric Péters**

- Statut changé de Résolu (à déployer) à Solution déployée

**Fichiers**

---

0001-toulouse_axel-bool-mapping-38464.patch	6,68 ko	13 décembre 2019	Lauréline Guérin
0001-toulouse_axel-bool-mapping-38464.patch	6,98 ko	13 décembre 2019	Lauréline Guérin
0001-toulouse_axel-bool-mapping-38464.patch	7,61 ko	17 décembre 2019	Lauréline Guérin