

Passerelle - Development #67779

Création d'un connecteur vers Esabora

28 juillet 2022 14:29 - Thomas Noël

Statut:	Fermé	Début:	28 juillet 2022
Priorité:	Normal	Echéance:	
Assigné à:	A. Berriot	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Avec deux endpoints "modbdd" et "mult" pour commencer.			

Révisions associées

Révision 917e8275 - 08 août 2022 16:07 - A. Berriot

esabora: initial connector implementation (#67779)

Historique

#3 - 03 août 2022 09:32 - A. Berriot

- Fichier 0001-esabora-initial-connector-implementation-67779.patch ajouté
- Patch proposed changé de Non à Oui

#4 - 04 août 2022 11:52 - A. Berriot

- Fichier 0001-esabora-initial-connector-implementation-67779.patch ajouté
- Statut changé de Nouveau à Solution proposée

#5 - 04 août 2022 12:01 - A. Berriot

Normalement ça intgère les points discuté avec Thomas et demandé sur le ticket client, à savoir :

- Création d'un dossier dans Esabora, via le endpoint "do-treatment" du connecteur
- Récupération de dossiers stockées dans esabora, via le endpoint "do-search" du connecteur

Malheureusement, Esabora ne fournit pas de structure unique pour les objets envoyés. Chaque webservice a des champs totalement différents, donc il est impossible de documenter correctement les données à envoyer au connecteur.

Le "do-create" gère des cas avancés comme envoyer plusieurs pièces jointes. Par exemple, dans un workflow:

```
Documents_Identité/0 = {{ form_var_fichier1 }}
Documents_Identité/1 = {{ form_var_fichier2 }}
Justificatif_Domicile = {{ form_var_fichier3 }}
```

Enverra à Esabora deux fichiers pour le champ "Documents_Identité" et un fichier pour "Justificatif_Domicile". J'ai testé chez eux et ça fonctionne.

Pour la recherche multicritère via le endpoint do-search, on utilise également une logique similaire:

```
critereons/ID_Dossier = {{ form_var_id_dossier }}
critereons/Date = {{ form_var_date }}
```

Pour effectuer une recherche sur les deux champs "ID_Dossier" et "Id_Date"

(je pose ça là pour pas oublier, je ferai une doc plus complète quand Marie sera de retour et qu'on aura testé ensemble).

#6 - 04 août 2022 12:04 - A. Berriot

- Statut changé de Solution proposée à En cours

#7 - 04 août 2022 12:16 - A. Berriot

- Fichier 0001-esabora-initial-connector-implementation-67779.patch ajouté
- Statut changé de En cours à Solution proposée

#8 - 04 août 2022 12:17 - A. Berriot

Voilà, j'avais raté de trois trucs (un test incomplet, des commentaires inutiles), mon dernier patch est reviewable ;)

#9 - 04 août 2022 15:02 - Thomas Noël

Faire une seule migration.

`verbose_name = _('API Esabora)` : tu peux juste écrire `_('Esabora)` ici.

Truc que j'aurais dû dire plus tôt : la réponse à `do-search` gagnera à être compatible avec le format des sources de données de Publik, une clé `"data"` contenant une liste de dictionnaires avec chacun un `"id"` et un `"text"` :

```
{
  "err": 0,
  "data": [
    {
      "id": "trucl",
      "text": "Foo 1",
      "column_1": "Foo 1",
      "column_2": "Foo 2",
      "column_3": "Foo 3",
    }
    ...
  ],
  "meta": (voir ci-dessous)
}
```

où :

- Le `"text"` sera juste un copie du premier item de `columnDataList`.
- Les clés `"column_1"` seront des slugify(`"Column 1"`).replace('-', '_') .. parce que sinon Publik ne pourra pas gérer
- Dans un dictionnaire supplémentaire `"meta"` on renverrait d'autres données, telle que la liste des vrais noms des colonnes, et d'autres données renvoyées par Esabora qui pourraient peut-être compter un jour :

```
"meta": {
  "columns_name": {
    "column_1": "Column 1",
    "column_2": "Column 2",
    "column_3": "Column 3",
  }
  "searchId": "23568",
  "nbResults": 2,      # tiens, tu as mis 1 dans ton test alors qu'on voit deux résultats :)
  "keyList" : ["internal.id"]
}
```

Avec ce format "source de donnée" la réponse sera plus "classique Publik" et plus facilement utilisable par les habitués.

Je propose de renommer `"do-search"` en juste `"search"` ... et `"do-treatment"` en `"treatment"` mais je suis pas même bien à l'aise avec ce nom, l'usage ça sera surtout pour insérer des machins, on pourrait l'appeler `"insert"` ?

Hey, tu as vu que je ne t'embête pas avec les fonctions `exception_to_text` & co déclarées après leur usage !? Je vieilli.

Personnellement, je retirerais la référence à stackoverflow parce que c'est la honte non ? ;) Je veux dire, la formule est juste la définition de cet encodage.

#10 - 08 août 2022 14:41 - A. Berriot

- Fichier `0001-esabora-initial-connector-implementation-67779.patch` ajouté

#11 - 08 août 2022 14:49 - A. Berriot

Merci pour tes retours

Thomas Noël a écrit :

Faire une seule migration.

`verbose_name = _('API Esabora)` : tu peux juste écrire `_('Esabora)` ici.

fait

Truc que j'aurais dû dire plus tôt : la réponse à do-search gagnera à être compatible avec le format des sources de données de Publik, une clé "data" contenant une liste de dictionnaires avec chacun un "id" et un "text" :

[...]

où :

- Le "text" sera juste un copie du premier item de columnDataList.
- Les clés "column_1" seront des slugify("Column 1").replace('-', '_') .. parce que sinon Publik ne pourra pas gérer
- Dans un dictionnaire supplémentaire "meta" on renverrait d'autres données, telle que la liste des vrais noms des colonnes, et d'autres données renvoyées par Esabora qui pourraient peut-être compter un jour :

[...]

Avec ce format "source de donnée" la réponse sera plus "classique Publik" et plus facilement utilisable par les habitués.

J'ai adapté, je me suis permise d'utiliser un mapping pour keys_name sur le même modèle que ta suggestion pour columns_name, par souci de cohérence.

Je propose de renommer "do-search" en juste "search" ... et "do-treatment" en "treatment" mais je suis pas même bien à l'aise avec ce nom, l'usage ça sera surtout pour insérer des machins, on pourrait l'appeler "insert" ?

J'ai utilisé ces nom par cohérence vu que ça tape derrière sur l'API esabora qui utilise doTreatment et doSearch. Il me semble que ça serait plus simple pour une personne qui se réfèrera à la documentation, mais je peux changer.

Hey, tu as vu que je ne t'embête pas avec les fonctions exception_to_text & co déclarées après leur usage !? Je vieilli.

Je n'ai pas bien compris. exception_to_text est importée, pas déclarée dans le module.

Personnellement, je retirerais la référence à stackoverflow parce que c'est la honte non ? ;) Je veux dire, la formule est juste la définition de cet encodage.

Je ne vois pas du tout le problème à mettre un lien vers une doc sur la question. Est-ce que je devrai avoir honte de mon cursus littéraire et de ne pas être capable d'écrire ou de comprendre les formules pour faire de la base 64 ?

#12 - 08 août 2022 15:29 - Thomas Noël

Agate Berriot a écrit :

Truc que j'aurais dû dire plus tôt : la réponse à do-search gagnera à être compatible avec le format des sources de données de Publik, une clé "data" contenant une liste de dictionnaires avec chacun un "id" et un "text" :

(...)

J'ai adapté, je me suis permise d'utiliser un mapping pour keys_name s

Attention : c'est "id" et non pas "internalid" qu'il faut en sortie.

```
expected = {
  'err': 0,
  'data': [
    {
      'internalid': 'id1', <---- on devrait voir ici « 'id': 'id1', »
      'text': 'Foo 1',
      "column_1": 'Foo 1',
      "column_2": 'Foo 2',
      "column_3": 'Foo 3',
    },
  ],
}
```

Je propose de renommer "do-search" en juste "search" ... et "do-treatment" en "treatment" mais je suis pas même bien à l'aise avec ce nom, l'usage ça sera surtout pour insérer des machins, on pourrait l'appeler "insert" ?

J'ai utilisé ces nom par cohérence vu que ça tape derrière sur l'API esabora qui utilise doTreatment et doSearch. Il me semble que ça serait plus simple pour une personne qui se réfèrera à la documentation, mais je peux changer.

Non, bonne idée, effectivement il faudra lui un peu la doc Esabora pour savoir le payload, donc autant reprendre d'autres termes par ailleurs, bien vu.

Hey, tu as vu que je ne t'embête pas avec les fonctions exception_to_text & co déclarées après leur usage !? Je vieilli.

Je n'ai pas bien compris. `exception_to_text` est importée, pas déclarée dans le module.

Je parlais de `esabora_row_to_object` ou `get_treatment_payload`. Dans d'autres connecteurs tu verras qu'on fait souvent de ces choses des `classmethod`, mais c'est presque du snobisme ou pire des javateries.

Mais laisse tomber, ton module est très bien comme ça, c'est pas déclaré à 36 kilomètres.

Personnellement, je retirerais la référence à `stackoverflow` parce que c'est la honte non ? ;) Je veux dire, la formule est juste la définition de cet encodage.

Je ne vois pas du tout en quoi le problème à mettre un lien vers une doc sur la question. Est-ce que je devrai avoir honte de mon cursus littéraire et de ne pas être capable d'écrire ou de comprendre les formules pour faire de la base 64 ?

Pourtant c'est marrant à comprendre `base64` (y'a pas beaucoup plus bête comme codage). Mais bon, c'est comme ça, quand on me parle de `stackoverflow` je réponds toujours par une taquinerie. Pour faire croire que j'en ai jamais besoin, sans doute.

#13 - 08 août 2022 15:42 - A. Berriot

- *Fichier 0001-esabora-initial-connector-implementation-67779.patch ajouté*

#14 - 08 août 2022 15:48 - A. Berriot

Thomas Noël a écrit :

Agate Berriot a écrit :

Truc que j'aurais dû dire plus tôt : la réponse à `do-search` gagnera à être compatible avec le format des sources de données de PubliK, une clé "data" contenant une liste de dictionnaires avec chacun un "id" et un "text" :
(...)

J'ai adapté, je me suis permise d'utiliser un mapping pour `keys_name s`

Attention : c'est "id" et non pas "internalid" qu'il faut en sortie.

D'accord. J'ai bricolé ce que j'ai pu, parce que la doc d'Esabora laisse sous entendre qu'il peut y avoir plusieurs items dans `keyList`. Donc j'ai appliqué la même logique que pour `text` : pour chaque objet, je prends comme id la valeur correspondant au premier item de `keyList`.

J'ai laissé les champs de `keyList` dans la payload car leur nom n'est absolument pas spécifié. J'ai utilisé `internalid` dans mes tests mais ça peut aussi bien être `Reference_SAS` ou `Toto`, au moins une personne d'accéder à un champ précis pourra le faire simplement sans se préoccuper de notre logique de mapping d'ID, surtout s'il peut y avoir plusieurs id interne renvoyés par Esabora...

Je parlais de `esabora_row_to_object` ou `get_treatment_payload`. Dans d'autres connecteurs tu verras qu'on fait souvent de ces choses des `classmethod`, mais c'est presque du snobisme ou pire des javateries.

Je préfère ne pas mettre dans une méthode un truc qui n'a pas besoin de `self` ou de `cls`. (Et d'une manière générale de ne pas utiliser les classes, je leur trouve de moins en moins d'intérêt et ça complexifie tout les tests).

#15 - 08 août 2022 16:00 - Thomas Noël

- *Statut changé de Solution proposée à Solution validée*

Ca me semble tout bon.

S'agissant d'un nouveau connecteur sans impact ailleurs, et qui n'a pas besoin de mention dans les notes de mise à jour, tu peux le pousser dès que jenkins est vert (inutile d'attendre une semaine).

Je m'occuperai de la release aussitôt (traduction et tags).

#16 - 08 août 2022 16:07 - A. Berriot

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

```
commit 8171aa44a967d57f6fa169221d0048a33f2ff9c5
Author: Agate <aberriot@entrouvert.com>
Date: Mon Aug 1 16:48:58 2022 +0200
```

```
esabora: initial connector implementation (#67779)
```

#17 - 08 août 2022 17:14 - Transition automatique

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

#18 - 09 octobre 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-esabora-initial-connector-implementation-67779.patch	14,5 ko	03 août 2022	A. Berriot
0001-esabora-initial-connector-implementation-67779.patch	18,4 ko	04 août 2022	A. Berriot
0001-esabora-initial-connector-implementation-67779.patch	18,6 ko	04 août 2022	A. Berriot
0001-esabora-initial-connector-implementation-67779.patch	18,8 ko	08 août 2022	A. Berriot
0001-esabora-initial-connector-implementation-67779.patch	18,9 ko	08 août 2022	A. Berriot