w.c.s. - Development #56537

prendre en charge les fichiers de type "texte" (txt, xml, svg,...) dans l'action document

01 septembre 2021 01:09 - Thomas Noël

Statut:	Fermé	**Début:**	01 septembre 2021
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Parce que ce n'est pas bien dur à faire (juste appliquer le rendu de gabarit au fichier), et que ça permettrait de générer des documents parfois utiles.

Demandes liées:

Lié à Passerelle - Development #56526: connection json → xml	**Rejeté**	**31 août 2021**

Révisions associées

Révision f2eaa892 - 10 septembre 2021 14:57 - Benjamin Dauvergne

wf/export_to_model: support XML templates (#56537)

Historique

#2 - 01 septembre 2021 01:15 - Thomas Noël

- Lié à Development #56526: connection json → xml ajouté

#3 - 01 septembre 2021 08:33 - Frédéric Péters

(il y a quand même toute une série d'abus étranges que ça va amener, mais ok).

> juste appliquer le rendu de gabarit au fichier

Parce que produire de l'XML à base de gabarit, ça n'a jamais vraiment été recommandé; pour s'éviter des galères j'ajouterais aussi une étape "vérifier un minimum ce qui a été produit", i.e. si on est sur de l'xml vérifier qu'il est ~~valide~~ bien formé, pareil pour du json, etc.

#4 - 01 septembre 2021 10:58 - Benjamin Dauvergne

L'implémentation minimale pour moi c'est de dire que si on ne détecte pas un fichier libre-office via libmagic, on est en mode texte. La validation xml si c'est du xml ça peut faire l'objet d'un autre ticket, si on s'aperçoit que c'est fréquent.

#5 - 01 septembre 2021 11:08 - Frédéric Péters

Mon souhait ici est qu'il soit intégré dès le début cette validation.

#6 - 01 septembre 2021 15:31 - Thomas Noël

Frédéric Péters a écrit :

> Mon souhait ici est qu'il soit intégré dès le début cette validation.

Je n'ai pas compris "dès le début". On ne pourra pas valider qu'un template Django destiné à produire du XML va bien le faire, le moindre "if" ou "for" va vite casser la logique du XML. On peut en revanche détecter si c'est bien un template Django à la syntaxe a priori valide, c'est de ça dont tu parlais ?

#7 - 01 septembre 2021 15:32 - Frédéric Péters

"dès le début" c'était pour dire dès ce ticket, c'est mon objection à "ça peut faire l'objet d'un autre ticket, si on s'aperçoit que c'est fréquent.".

#8 - 02 septembre 2021 11:28 - Benjamin Dauvergne

Ok la vue de JSON me fait un peu peur, donc je pense qu'on va limiter ça à ".xml" dès le départ comme on a déjà une détection un peu bâtarde de RTF et OpenDocument à base de content_type/extension. Je n'ai pas du tout envie de débugger du JSON généré par un template. Je préférerai qu'on ouvre les vannes petit à petit suite à cette première expérience (genre si on voit un besoin de CSV, on l'envisagera à ce moment là).

#9 - 02 septembre 2021 12:15 - Thomas Noël

À noter quand même que sur du libreoffice on ne teste rien pas grand chose : upload.content_type.startswith('application/vnd.oasis.opendocument.')

Quelle serait l'idée de valider quelque chose ici ? Si le XML renvoyé est foireux, quelle importance, et autant voir le résultat planté plutôt que rien.

#10 - 02 septembre 2021 12:20 - Frédéric Péters

C'est n'importe quoi de générer du XML à coup de print() et si on veut le faire malgré tout je pense que vérifier que ce qui est généré est bien formé est un minimum, que ça permet d'être averti dès w.c.s., et pas trois jours plus tard par un commentaire comme quoi l'application métier supposée absorber le fichier généré a fait une trace java de 2 kilomètres.

#11 - 02 septembre 2021 13:23 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#12 - 02 septembre 2021 13:23 - Benjamin Dauvergne

- Fichier 0001-wf-export_to_model-support-XML-templates-56537.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#13 - 02 septembre 2021 13:24 - Benjamin Dauvergne

Voilà, je suis allé au plus simple, ça n'accepte que les fichiers .xml et ça vérifie le format du fichier en sortie. Et en vrai pour tricher on peut toujours mettre un fichier qui s'appelle .xml mais qui contient du JSON ou du CSV et le renommer ensuite.

#14 - 02 septembre 2021 14:45 - Thomas Noël

Benjamin Dauvergne a écrit :

> Voilà, je suis allé au plus simple, ça n'accepte que les fichiers .xml et ça vérifie le format du fichier en sortie. Et en vrai pour tricher on peut toujours mettre un fichier qui s'appelle .xml mais qui contient du JSON ou du CSV et le renommer ensuite.

Sur le test du type : upload.base_filename.endswith('.rtf') → upload.base_filename.endswith('.xml')

raise UploadValidationError(_('Only RTF and OpenDocument files can be used')) : je préférerais ne pas cacher que XML est accepté, et l'ajouter dans le message.

raise TemplatingError(_('Error in template: output is not valid XML (%s)') % str(e)) : j'imagine que ça va planter le workflow qui va s'arrêter sur le statut, ça ne me va pas : les actions suivantes ne sont pas exécutées, risque de se retrouver dans un statut "technique" qui ne devrait pas être possible, mais surtout on ne verra pas le XML généré (pour comprendre le pépin). Perso je trouve qu'on devrait juste avoir le record_error, mais pas ce raise. Oui, il y aura un XML planté dans outstream, mais c'est mieux que de ne rien avoir. Et on comprendra la source de la trace java pourrie.

(Je vois qu'on fait ce raise aussi pour le mode RTF, comme ce n'est plus jamais utilisé on ne voit pas de soucis)

L'alternative serait d'ajouter un "saut en cas d'erreur" dans l'action, et bof.

#15 - 02 septembre 2021 15:36 - Benjamin Dauvergne

- Fichier 0001-wf-export_to_model-support-XML-templates-56537.patch ajouté

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > Voilà, je suis allé au plus simple, ça n'accepte que les fichiers .xml et ça vérifie le format du fichier en sortie. Et en vrai pour tricher on peut toujours mettre un fichier qui s'appelle .xml mais qui contient du JSON ou du CSV et le renommer ensuite.

Je dis une bêtise, ça plantera sur la validation XML, manque de glucose avant le repas.

> Sur le test du type : upload.base_filename.endswith('.rtf') → upload.base_filename.endswith('.xml')

Ok, j'ajoute un test. PS: au passage je vois que la partie avant ça n'est pas bonne non plus, il faut ignorer content_type dans ce cas là et j'ai ajouté un vague test de préfixe sur le contenu en plus du test sur l'extension.

> raise UploadValidationError(_('Only RTF and OpenDocument files can be used')) : je préférerais ne pas cacher que XML est accepté, et l'ajouter dans le message.

Ok, un oubli.

> raise TemplatingError(_('Error in template: output is not valid XML (%s)') % str(e)) : j'imagine que ça va planter le workflow qui va s'arrêter sur le statut, ça ne me va pas : les actions suivantes ne sont pas exécutées, risque de se retrouver dans un statut "technique" qui ne devrait pas être possible, mais surtout on ne verra pas le XML généré (pour comprendre le pépin). Perso je trouve qu'on devrait juste avoir le record_error, mais pas ce raise. Oui, il y aura un XML planté dans outstream, mais c'est mieux que de ne rien avoir. Et on comprendra la source de la trace java pourrie.

Ok, on aura quand même une erreur si le template Django est planté, ça va ça ? (sur une TemplateError on convertit en TemplatingError, je ne sais pas bien c'est utile d'ailleurs).

#16 - 09 septembre 2021 12:46 - Thomas Noël

Jenkins est tout rouge, sans doute y'a des trucs à revoir.

Ok pour la validation du XML qui donne juste un warning, très bien selon moi. Je reste gêné par le "raise" en bas de apply_text_template_to_formdata. Je comprends que c'est repris de apply_rtf_template_to_formdata (que personne plus jamais n'utilise) : pour moi ça va faire le process de génération du document va exploser en milieu de workflow, ça ne va pas. Je préfère qu'on fasse comme pour odt : on laisse filer, au pire y'a pas de doc généré, juste une erreur enregistrée, voilà c'est la vie.

#17 - 09 septembre 2021 16:05 - Benjamin Dauvergne

Thomas Noël a écrit :

> Ok pour la validation du XML qui donne juste un warning, très bien selon moi. Je reste gêné par le "raise" en bas de apply_text_template_to_formdata. Je comprends que c'est repris de apply_rtf_template_to_formdata (que personne plus jamais n'utilise) : pour moi ça va faire le process de génération du document va exploser en milieu de workflow, ça ne va pas. Je préfère qu'on fasse comme pour odt : on laisse filer, au pire y'a pas de doc généré, juste une erreur enregistrée, voilà c'est la vie.

Bon en fait de toute façon template_on_formdata() ne peut pas lever TemplateError parce qu'il vérifie que la syntaxe est bonne et que le constructeur de Template ne reçoit pas raises=True concernant les erreurs durant le rendu, en fait c'est tout à inutile (ou alors peut-être en EZT, mais passons), je vire ça et ne met aucune contrôle de plus.

#18 - 09 septembre 2021 16:48 - Thomas Noël

Benjamin Dauvergne a écrit :

> Thomas Noël a écrit :
>
> > Ok pour la validation du XML qui donne juste un warning, très bien selon moi. Je reste gêné par le "raise" en bas de apply_text_template_to_formdata. Je comprends que c'est repris de apply_rtf_template_to_formdata (que personne plus jamais n'utilise) : pour moi ça va faire le process de génération du document va exploser en milieu de workflow, ça ne va pas. Je préfère qu'on fasse comme pour odt : on laisse filer, au pire y'a pas de doc généré, juste une erreur enregistrée, voilà c'est la vie.
>
> Bon en fait de toute façon template_on_formdata() ne peut pas lever TemplateError parce qu'il vérifie que la syntaxe est bonne et que le constructeur de Template ne reçoit pas raises=True concernant les erreurs durant le rendu, en fait c'est tout à inutile (ou alors peut-être en EZT, mais passons), je vire ça et ne met aucune contrôle de plus.

Ca me va d'aller avec ça.

Il me reste le petit « xml_prefix == b'<' » qui impose donc que le document ne commence jamais par du Django... en vrai je pense que ça peut s'entendre comme condition. Je pensais même même être plus méchants encore et imposer « <?xml » mais ça demanderait d'être sûr qu'il y a 5 caractères, etc... (d'ailleurs il se passe quoi si on upload un fichier vide ?).

#19 - 09 septembre 2021 16:49 - Thomas Noël

Au passage, ajouter "XML" dans le

```
raise UploadValidationError(_('Only RTF and OpenDocument files can be used'))
```

#20 - 09 septembre 2021 17:20 - Benjamin Dauvergne

- Fichier 0001-wf-export_to_model-support-XML-templates-56537.patch ajouté

J'ai carrément viré RTF du message, on va dire que c'est la dépréciation qui commence.

#21 - 10 septembre 2021 12:23 - Thomas Noël

Ah, zut, raté ça, lorsque le rendu n'est pas XML, plutôt que "Error in template for export to model", dire plutôt "Le résultat du gabarit n'est pas un document XML valide" (enfin en anglais, mais tu connais mes performances).

Après ça je pourrais très certainement valider.

#22 - 10 septembre 2021 14:57 - Benjamin Dauvergne

- Fichier 0001-wf-export_to_model-support-XML-templates-56537.patch ajouté

Je commence à en avoir un peu ras le cul de vous, allez vous jeter dans les plantes vertes.

#23 - 10 septembre 2021 15:56 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne a écrit :

> Je commence à en avoir un peu ras le cul de vous, allez vous jeter dans les plantes vertes.

Ah, y'en a, super !

En vrai, merci à toi pour ce dev (me retrouver juste relecteur-râleur fût confortable)

#24 - 10 septembre 2021 16:10 - Benjamin Dauvergne

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

```
commit f2eaa89247504a75e848ff32a23dbe2436a21b0a
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Thu Sep 2 11:43:30 2021 +0200

    wf/export_to_model: support XML templates (#56537)
```

#26 - 14 septembre 2021 19:16 - Frédéric Péters

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

Fichiers

0001-wf-export_to_model-support-XML-templates-56537.patch	6,98 ko 02 septembre 2021	Benjamin Dauvergne
0001-wf-export_to_model-support-XML-templates-56537.patch	7,79 ko 02 septembre 2021	Benjamin Dauvergne
0001-wf-export_to_model-support-XML-templates-56537.patch	6,63 ko 09 septembre 2021	Benjamin Dauvergne
0001-wf-export_to_model-support-XML-templates-56537.patch	6,64 ko 10 septembre 2021	Benjamin Dauvergne