

## w.c.s. - Bug #27323

### Permettre d'attacher un export JSON d'un formulaire à un mail

15 octobre 2018 09:57 - Benjamin Dauvergne

<b>Statut:</b>	Fermé	<b>Début:</b>	15 octobre 2018
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Dauvergne	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Voir #27302.			

#### Révisions associées

##### Révision 73505821 - 05 novembre 2018 10:24 - Benjamin Dauvergne

evalutils: add attachment(content, filename, content\_type) helper (#27323)

To allow creating mail attachments easily, test contains an example sending a full JSON export of a formdata.

#### Historique

##### #1 - 15 octobre 2018 09:58 - Benjamin Dauvergne

- Fichier 0001-add-json\_attachment-properties-to-LazyFormData-27323.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Assez transparent, avec un test.

##### #3 - 15 octobre 2018 17:02 - Thomas Noël

Ça sera tellement pas utilisé ailleurs que pour #27302, ne peut-on pas construire une variable de traitement de type fichier (via un peu de Python) puis l'envoyer en fichier attaché ?

Et je me réponds à moi-même, que ça pose certainement un soucis de récursivité ./ (la variable faisant partie elle-même du json)

Donc autre idée, parce que le format en dictionnaire de \_\_to\_json\_attachment m'ennuie un peu : avoir plutôt un PicklableUpload ?

##### #4 - 17 octobre 2018 10:13 - Benjamin Dauvergne

Thomas Noël a écrit :

Ça sera tellement pas utilisé ailleurs que pour #27302, ne peut-on pas construire une variable de traitement de type fichier (via un peu de Python) puis l'envoyer en fichier attaché ?

On a base64 dans le global\_eval\_dict ? Je n'en ai pas l'impression, difficile de créer un export fichier dans ce cas (je pense qu'il faudrait plutôt publier un constructeur facile à utiliser genre json\_attachment(filename, content\_type, content).

Et je me réponds à moi-même, que ça pose certainement un soucis de récursivité ./ (la variable faisant partie elle-même du json)

Hein ?

Donc autre idée, parce que le format en dictionnaire de \_\_to\_json\_attachment m'ennuie un peu : avoir plutôt un PicklableUpload ?

Oui aucun souci, c'était plus court de passer par le format JSON qui est de toute façon parfaitement supporté par FileField.convert\_value\_from\_anything() mais bon si tu préfères (le code sera plus long il me semble).

PS: finalement je reste au format JSON, PicklableUpload ne marche pas dans ce cas là (c'est une abstraction pourrie, get\_content() retourne None, parce que tant qu'il a pas été picklé, PicklableUpload n'est pas dans un état fonctionnel ./, un PicklableUpload juste initialisé n'ayant pas d'attribut 'qfilename', PicklableUpload.get\_content() retourne toujours None)

## #5 - 22 octobre 2018 10:52 - Frédéric Péters

On a base64 dans le global\_eval\_dict ?

Si on l'avait, ça permettrait de faire quelque chose sans modification à variables.py ? (parce que ça semble quand même de l'ultra-spécifique, qui ne sera jamais réutilisé).

```
json_attachment(filename, content_type, content)
```

Ou bien ça, ou plutôt juste attachment(content, filename="", content\_type='application/octet-stream'), mais même question qu'au-dessus, on a le nécessaire en variables pour ensuite faire =attachment(form.export\_to\_json(include\_files=True)) ?

## #6 - 22 octobre 2018 11:58 - Benjamin Dauvergne

Frédéric Péters a écrit :

On a base64 dans le global\_eval\_dict ?

Si on l'avait, ça permettrait de faire quelque chose sans modification à variables.py ? (parce que ça semble quand même de l'ultra-spécifique, qui ne sera jamais réutilisé).

Non comme tu le décèles plus loin on n'a pas accès aux méthodes de FormData depuis LazyFormData.

```
json_attachment(filename, content_type, content)
```

Ou bien ça, ou plutôt juste attachment(content, filename="", content\_type='application/octet-stream'), mais même question qu'au-dessus, on a le nécessaire en variables pour ensuite faire =attachment(form.export\_to\_json(include\_files=True)) ?

Non. Je propose de partir sur un as\_json/as\_json\_with\_files qui appelle simplement export\_to\_json et un helper utils.attachment() comme tu le proposes, est-ce que ça irait à tout le monde ?

## #7 - 22 octobre 2018 12:05 - Benjamin Dauvergne

- Fichier 0002-evalutils-add-attachment-content-filename-content\_ty.patch ajouté
- Fichier 0003-rename-as\_json\_attachment-to-as\_json-to-be-squashed.patch ajouté
- Fichier 0001-add-json\_attachment-properties-to-LazyFormData-27323.patch ajouté

## #8 - 22 octobre 2018 13:19 - Frédéric Péters

Tu inclurais dans ton test un content\_type='application/json' pour l'un des deux attachements, pour tester ça ?

Mais aussi, je regarde et form.formdata devrait te donner accès au formdata brut, et du coup l'export\_to\_json devient accessible. (l'accès direct aux objets n'est pas quelque chose que je voudrais encourager mais ça nous donne davantage de temps pour réfléchir à ce qu'on veut comme nouvelles méthodes/propriétés sur LazyFormdata).

## #9 - 22 octobre 2018 15:07 - Benjamin Dauvergne

- Fichier 0001-evalutils-add-attachment-content-filename-content\_ty.patch ajouté

Voilà, voilà.

## #10 - 05 novembre 2018 12:07 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

Ok, go. (pour cette release)

## #11 - 06 novembre 2018 11:31 - Thomas Noël

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

```
commit 735058211a92ae390f649357867fb8f6d1114a3f
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Mon Oct 15 09:57:30 2018 +0200
```

```
evalutils: add attachment(content, filename, content_type) helper (#27323)
```

To allow creating mail attachments easily, test contains an example sending a full JSON export of a formdata.

## #12 - 23 décembre 2018 14:36 - Frédéric Péters

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

### Fichiers

---

0001-add-json_attachment-properties-to-LazyFormData-27323.patch	3,04 ko	15 octobre 2018	Benjamin Dauvergne
0002-evalutils-add-attachment-content-filename-content_ty.patch	1,07 ko	22 octobre 2018	Benjamin Dauvergne
0003-rename-as_json_attachment-to-as_json-to-be-squashed.patch	2,27 ko	22 octobre 2018	Benjamin Dauvergne
0001-add-json_attachment-properties-to-LazyFormData-27323.patch	3,04 ko	22 octobre 2018	Benjamin Dauvergne
0001-evalutils-add-attachment-content-filename-content_ty.patch	2,79 ko	22 octobre 2018	Benjamin Dauvergne