w.c.s. - Development #8274

Permettre l'envoi de fichiers joints lors de l'action "envoyer un mail"

16 septembre 2015 13:32 - Thomas Noël

Statut:	Fermé	**Début:**	02 octobre 2017
Priorité:	Normal	**Echéance:**	17 octobre 2019
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Idée de présentation dans l'action: en dessous du template de mail, une liste d'items "Documents à joindre", où on place des noms de variable form_var_xxxx, qui sont des champs fichiers.

Il faudra aussi prévoir l'envoi de documents attachés dans le journal (posés lors du traitement de la demande), et dans les champs backoffice (cf #8273)

Demandes liées:

Suit w.c.s. - Development #19005: partager les get_date_value et get_file_val...	**Fermé**	**27 septembre 2017**
Suit w.c.s. - Bug #19115: qommon.emails : gestion complète des attachements	**Fermé**	**30 septembre 2017**

Révisions associées

Révision 786343b2 - 02 octobre 2017 15:38 - Thomas Noël

workflows: allow attachments in emails (#8274)

Historique

#1 - 16 septembre 2015 13:33 - Thomas Noël

- Sujet changé de Permettre l'envoie de fichiers joints lors de l'action "envoyer un mail" à Permettre l'envoi de fichiers joints lors de l'action "envoyer un mail"

#2 - 16 septembre 2015 13:40 - Thomas Noël

- Tracker changé de Bug à Development

#4 - 31 octobre 2016 12:16 - Thomas Noël

Dans un premier temps et pour simplifier l'interface, je propose de prévoir uniquement l'envoi de documents contenus dans des champs de traitement de type fichier (la liste de ceux-ci étant connue dans le workflow, contrairement aux variables).

#5 - 23 mai 2017 21:35 - Thomas Noël

- Echéance mis à 12 juin 2017

Essayer d'avoir cela pour le 12 juin, pour la mise en prod du 22.

Quelques notes :

- sans doute un premier patch avec la modification du code d'envoi des mails, pour y ajouter la possibilité de documents attachés (et s'embêter à voir comment tester ça)
- dans l'action Sendmail elle même :
  - donc : permettre l'envoi de n'importe quel champ fichier backoffice
  - quand la valeur du champ n'est pas un fichier ou n'existe pas, si le champ relatif est marqué obligatoire : alerter (bon oui mais comment et qui ? en attendant, lever une exception et loger) ; sinon envoyer le mail sans le fichier concerné
  - quand un champ n'existe plus (supprimé du formdef ou sur workflow) : envoyer le mail sans le fichier

#6 - 23 mai 2017 21:45 - Frédéric Péters

> sans doute un premier patch avec la modification du code d'envoi des mails, pour y ajouter la possibilité de documents attachés (et s'embêter à voir comment tester ça)

À vouloir commenter pour dire qu'il y a un **kwargs qui va permettre de descendre jusque la fonction email elle-même, je lis la signature de la fonction en question et il y a déjà une paramètre attachments. (commit 855a343f de 2010 de Benjamin, je ne sais pas trop à quel projet ça pouvait être lié).

> donc : permettre l'envoi de n'importe quel champ fichier backoffice

Hésitation là-dessus, on veut vraiment un sélecteur de champ fichier ou bien une expression python permettant n'importe quoi. (ou un milieu où on propose un choix parmi une série de ('=form_var_bo_fichier', 'Champ X'), généré à partir des champs, comme ça on se donne ensuite la tranquillité nécessaire pour basculer vers une expression python au moment qui l'appellera, sans se prendre la tête sur une migration.

> quand la valeur du champ n'est pas un fichier ou n'existe pas, si le champ relatif est marqué obligatoire : alerter (bon oui mais comment et qui ? en attendant, lever une exception et logger) ; sinon envoyer le mail sans le fichier concerné

Oui, pour moi on passe par self.compute() et on rapporte les exceptions via notify_of_exception; aussi on autorise le compute() à retourner None et on ignore alors juste cette référence.

#7 - 23 mai 2017 21:48 - Frédéric Péters

> Hésitation là-dessus, on veut vraiment un sélecteur de champ fichier ou bien une expression python permettant n'importe quoi.

(c'est une question mal formulée), et à ce sujet, ça me va de sortir/exploiter le get_file_value() qu'on a dans wcs/wf/backoffice_fields.py.

#8 - 24 mai 2017 00:05 - Thomas Noël

J'avais en tête un système qui ne permette pas de Python (parce que oui c'est oversouple et überpuissant, mais ça ne nous apporte aussi des fonctionnalités indocumentables et des dizaines de traces indéchiffrables dans nos mbox ; j'ai parlé).

#9 - 24 mai 2017 00:10 - Frédéric Péters

Mais ça oblige à refaire toute une mécanique qui existe par ailleurs déjà; je comprends l'argument, c'est pour ça que j'avais la suggestion d'un choix dans une liste fermée mais dont les éléments seraient in fine des expressions python menant au fichier choisi (et libre à nous plus tard de transformer le <select> pour avoir un choix "autre" laissant libre entrée d'une expression Python, comme on le fait pour le choix du destinataire).

#10 - 24 mai 2017 04:40 - Pierre Cros

Je continue de trouver la fonctionnalité intéressante (en particulier pour les gens qui n'accèderont pas à la plate-forme), mais faut pas se mettre la rate au court-bouillon sur le délais : si on a ça fin juillet ça va.

#13 - 21 septembre 2017 15:12 - Thomas Noël

- Echéance changé de 12 juin 2017 à 02 octobre 2017

#15 - 26 septembre 2017 17:28 - Thomas Noël

- Fichier 0001-mail-with-attachments-draft-8274.patch ajouté

Une proposition d'UI qui permet de choisir les champs fichiers dispos dans le workflow.

- Si le champs fichier n'a pas de varname, on utilise un __fboN (et donc si le champs est supprimé, ça ne marchera plus, mais on fera en sorte de pas planter).
- Si y'a un varname, on utilise son nom de variable, comme ça on pourra encore se "jouer" à remplacer le champ fichier par un autre, il fonctionnera encore si le varname est stable
- Si c'est du Python, j'imagine proposer un truc permettant d'utiliser soit un dico content/filename/content_type, soit un Upload, à voir

Au final on a attachments = ['form_var_truc', 'form_var_machin', '__fbo1', 'dupython()'] qu'il faudra parcourir pour gérer les __fbo1, et le reste avec des compute.

#16 - 28 septembre 2017 00:45 - Benjamin Dauvergne

On peut facilement obtenir la liste des varname associés aux actions de génération à partir d'un modèle ou d'attachement de fichier, ça me dirait bien de les avoir directement dans le sélecteur plutôt que d'avoir à utiliser une expression Python.

J'ai l'impression que ce sera des cas plus fréquent que d'avoir un champ BO de type fichier; pour moi le cas typique ce sera un agent qui génère un accusé soit via w.c.s. soit manuellement, l'accusé est attaché à la demande puis envoyé par mail.

#17 - 28 septembre 2017 09:57 - Thomas Noël

Benjamin Dauvergne a écrit :

> On peut facilement obtenir la liste des varname associés aux actions de génération à partir d'un modèle ou d'attachement de fichier, ça me dirait

> bien de les avoir directement dans le sélecteur plutôt que d'avoir à utiliser une expression Python.

Avec ce patch (enfin, ce bout de patch) les champs backoffice fichiers sont bien présentés dans une liste jolie, avec un "autre" final si la personne veut s'obliger à taper du python (pour d'autres champs non backoffice, ou pour inventer des bêtises et créer des bogues)

> J'ai l'impression que ce sera des cas plus fréquent que d'avoir un champ BO de type fichier; pour moi le cas typique ce sera un agent qui génère un accusé soit via w.c.s. soit manuellement, l'accusé est attaché à la demande puis envoyé par mail.

Yep.

#18 - 28 septembre 2017 10:19 - Thomas Noël

- Fichier 0001-mail-with-attachments-draft-8274.patch ajouté

- Patch proposed changé de Non à Oui

Voilà un patch plus amusant, qui "cache" dans les paramètres avancés la possibilité de proposer des fichiers attachés s'il n'y a pas de champ fichier backoffice.

Bon à un moment je vais devoir faire le perform :)

#19 - 28 septembre 2017 10:33 - Benjamin Dauvergne

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > On peut facilement obtenir la liste des varname associés aux actions de génération à partir d'un modèle ou d'attachement de fichier, ça me dirait bien de les avoir directement dans le sélecteur plutôt que d'avoir à utiliser une expression Python.
>
> Avec ce patch (enfin, ce bout de patch) les champs backoffice fichiers sont bien présentés dans une liste jolie, avec un "autre" final si la personne veut s'obliger à taper du python (pour d'autres champs non backoffice, ou pour inventer des bêtises et créer des bogues)

Je ne vois pas en quoi ça répond à ma remarque, je ne parle pas des champs backoffice fichiers, mais des fichiers attachés dans des AttachmentPart (action joindre un fichier ou générer un document office).

#20 - 28 septembre 2017 10:44 - Thomas Noël

Benjamin Dauvergne a écrit :

> Je ne vois pas en quoi ça répond à ma remarque, je ne parle pas des champs backoffice fichiers, mais des fichiers attachés dans des AttachmentPart (action joindre un fichier ou générer un document office).

Ah, mal lu. Pour ça, mon plan ça serait plutôt d'avoir dans l'action de génération de fichier (et dans l'action d'appel webservice quand elle est en mode récup de fichier), la possibilité de pousser le résultat dans un des champs backoffice, à choisir parmi les champs de type fichiers.

#21 - 28 septembre 2017 10:53 - Benjamin Dauvergne

Quel raison t'incite à vouloir tout router dans les champs BO ? Il y a peut-être eu une discussion sur ce sujet que j'ai raté.

#22 - 28 septembre 2017 11:07 - Benjamin Dauvergne

Thomas Noël a écrit :

> Dans un premier temps et pour simplifier l'interface, je propose de prévoir uniquement l'envoi de documents contenus dans des champs de traitement de type fichier (la liste de ceux-ci étant connue dans le workflow, contrairement aux variables).

Je rebondis sur ça, la liste des variables est connu, au moins partiellement (on connait d'avance le nom des variables pour toutes les actions du workflow). Pour les web-service c'est juste compliqué parce qu'on ne connaît pas d'avance le format qui est renvoyé, si on avait des schémas JSON oui dans des sortes de WSDL, oui, mais on en a pas.

L'idée ce serait ça

```
wf = self.parent.parent
for status in wf.possible_status:
    for item in status.items:
        if isinstance(item, (AddAttachmentWorkflowStatusItem, ExportToModel)) and item.varname:
            codename = '=attachments.%s' % item.varname
            attachments_fields.append(
```

```
                # la valeur, le label, la clé
                (codename, item.label, codename))
```

Plutôt que de lister explicitement les classes d'Item qui génèrent des fichiers attachés, on pourrait aussi leur ajouter une méthode get_file_attachment_variables().

#23 - 28 septembre 2017 12:03 - Thomas Noël

Benjamin Dauvergne a écrit :

> Quel raison t'incite à vouloir tout router dans les champs BO ? Il y a peut-être eu une discussion sur ce sujet que j'ai raté.

Parce que c'est plus simple à expliquer à tout le monde, qu'elles ne sont pas cachées au fin fond d'une action, qu'elles sont toujours là... Moi c'est pour ça que je préfère (surtout parce que c'est plus simple à expliquer). Le but c'est quand même d'arriver à permettre des workflows où les gens n'aient pas à configurer des noms de variables et autres paramétrages "techniques".

Ensuite, oui, on peut parfaitement ajouter la liste des champs qu'on sait être normalement présents via des actions qui génèrent des docs, mais ça sera pas simple de les nommer ("fichier récupéré depuis le webservice appelé en 4ème position du statut trucmuche...", "fichier généré par la 3ème action de création d'un fichier dans le statut chose"...). Un simple "item.label" ne suffira pas.

(sinon pour les webservices, ceux qui doivent recevoir des documents sont configurés comme tels à travers response_type='attachment', donc on pourrait les avoir aussi)

Bref, je vais crânement suivre mon plan d'abord avec mes backoffice_fields, on verra comment compléter avec all_other_file_fields ensuite ;)

#24 - 29 septembre 2017 00:52 - Benjamin Dauvergne

Non non, c'est très bien ce que tu dis, mais je voudrais que ce soit public, compris et acceptés par tous que ce sera la règle pour faire des workflows maintenant, moi ça m'irait qu'on aille jusqu'à virer varname des actions générant des fichiers pour le remplacer par le fait de choisir la variable backoffice qui recevra le résultat, ça nous débarrassera aussi de ces histoires de attachements.xxx[y] parce qu'on se retrouve avec plusieurs fichiers pour la même variable :(Explicit is better than implicit, tu as raison.

#25 - 30 septembre 2017 19:48 - Thomas Noël

- Suit Development #19005: partager les get_date_value et get_file_value de wf/backoffice_fields.py ajouté

#26 - 30 septembre 2017 19:49 - Thomas Noël

- Suit Bug #19115: qommon.emails : gestion complète des attachements ajouté

#27 - 30 septembre 2017 19:56 - Thomas Noël

- Fichier 0001-workflows-allow-attachments-in-emails-8274.patch ajouté

Voilà le code, il manque encore les tests. Ça vient après #19005 et #19115 qui peuvent, eux, être relus dès maintenant.

#28 - 01 octobre 2017 01:24 - Thomas Noël

- Fichier 0001-workflows-allow-attachments-in-emails-8274.patch ajouté

- Statut changé de Nouveau à En cours

Voilà.

Donc, ça vient après #19005 et #19115.

#29 - 01 octobre 2017 08:49 - Frédéric Péters

Le calcul de la liste des pièces jointes possibles, il pourrait se faire dans le if 'attachments' in parameters:; et éventuellement déplacé dans une méthode get_attachments_options. Ajouter un notify_of_exception sur le cas du ValueError lors du convert_value_from_anything.

Idéalement il faudrait un test de l'interface d'édition, test_workflows_edit_sendmail_action dans test_admin_pages.py.

#30 - 01 octobre 2017 13:15 - Thomas Noël

- Fichier 0001-workflows-allow-attachments-in-emails-8274.patch ajouté

Yep, get_attachments_options séparée c'est plus joli. Pour le test j'ai gonflé le test_workflows_edit_email_action existant.

(C'est aussi testé "en vrai" avec des mails envoyés à un webmail trèès connu)

#31 - 02 octobre 2017 13:55 - Frédéric Péters

Je pense que je mettrais plutôt :

```
            for attachment in self.attachments:
                try:
                    picklableupload = eval(attachment, global_eval_dict, local_eval_dict)
                except:
                    get_publisher().notify_of_exception(sys.exc_info(),
                                                        context='[Sendmail/attachments]')
                    continue
                if not picklableupload:
                    continue
                try:
                    picklableupload = FileField.convert_value_from_anything(picklableupload)
                except ValueError:
                    get_publisher().notify_of_exception(sys.exc_info(),
                                                        context='[Sendmail/attachments]')
                    continue
                attachments.append(picklableupload)
```

Comme ça il devient légitime pour un eval() de retourner None, ce qui me semble avoir du sens.

#32 - 02 octobre 2017 14:17 - Thomas Noël

- Fichier 0001-workflows-allow-attachments-in-emails-8274.patch ajouté

Frédéric Péters a écrit :

> Je pense que je mettrais plutôt : (...)
> Comme ça il devient légitime pour un eval() de retourner None, ce qui me semble avoir du sens.

J'aime ce sens du détail, tout à fait d'accord.

#33 - 02 octobre 2017 15:15 - Frédéric Péters

Ack.

#34 - 02 octobre 2017 15:40 - Thomas Noël

- Statut changé de En cours à Résolu (à déployer)

```
commit 786343b28b2f33737b331b58c7351ebc0df3ce95
Author: Thomas NOEL <tnoel@entrouvert.com>
Date:   Tue Sep 26 17:13:47 2017 +0200

    workflows: allow attachments in emails (#8274)
```

#35 - 23 décembre 2018 14:53 - Frédéric Péters

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

Fichiers

0001-mail-with-attachments-draft-8274.patch	2,51 ko	26 septembre 2017	Thomas Noël
0001-mail-with-attachments-draft-8274.patch	3,23 ko	28 septembre 2017	Thomas Noël
0001-workflows-allow-attachments-in-emails-8274.patch	4,93 ko	30 septembre 2017	Thomas Noël
0001-workflows-allow-attachments-in-emails-8274.patch	10,5 ko	30 septembre 2017	Thomas Noël
0001-workflows-allow-attachments-in-emails-8274.patch	15,3 ko	01 octobre 2017	Thomas Noël
0001-workflows-allow-attachments-in-emails-8274.patch	15,3 ko	02 octobre 2017	Thomas Noël