

w.c.s. - Development #465

Publication d'une demande vers un modèle

01 juin 2011 11:28 - Benjamin Dauvergne

Statut:	Fermé	Début:	17 juin 2011
Priorité:	Haut	Echéance:	30 juin 2011
Assigné à:	Benjamin Dauvergne	% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:			
Description			
Il est demandé à Echirolles (et ailleurs), de pouvoir publier une demande vers un modèle (typiquement pour avoir une sortie papier, genre un courrier).			
Cela concerne un agent, quand une demande est arrivée à un état précis, je pense qu'on peut avoir une action de workflow qui serait "exporter dans modèle", on permet là-dedans de définir un document, et ouste. Pour le format du document, on peut fonctionner avec du RTF seul pour le moment (le modèle produit à partir du logiciel qui devra le lire ne posera pas de soucis de compatibilité).			
La seule difficulté pour le moment, si on veut gérer des modèles différents par formulaire, sans passer par une multiplication des workflows, c'est que les seuls champs déléguables au niveau du formulaire sont les champs de type texte; mais je ne pense même pas que ce soit un obstacle pour Echirolles.			
Sous-tâches:			
Development # 497: Mise à disposition de variables date pour l'export rtf			Rejeté
Demandes liées:			
Dupliqué par Au quotidien - Development #258: Edition d'une demande sur la ba...		Fermé	18 janvier 2011

Historique

#1 - 01 juin 2011 11:46 - Benjamin Dauvergne

Anciennement ticket #455 mort suite à une manipulation malencontreuse de ma part, je reprends ici le dernier commentaire de Fred.

Mis à jour par Benjamin Dauvergne il y a 5 jours

Un truc comme ça ne suffirait-il pas:

<http://perso.entreouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&id=7562b75a9d42e82afd0b8b9c60eb699b3989c076>

(bon là je pond un PDF, remplacer ça pour pondre juste du RTF, c'est kifkif)

#2 Mis à jour par Frédéric Péters il y a 4 jours

Avec un serveur FTP pour permettre à la mairie d'aller mettre ses modèles ? Mmmm, non.

#3 Mis à jour par Benjamin Dauvergne il y a 4 jours

De mon côté je ne vois pas bien comment se servir des workflows, pour déclencher l'action ok, mais comment je renvoie le modèle complété à l'utilisateur qui a déclenché l'action ? Les workflows ne peuvent pas renvoyer un contenu.

#4 Mis à jour par Frédéric Péters il y a 4 jours

cf le paiement, d'un workflow tu peux faire du redirect et servir du contenu. (auquotidien/extra/modules/payments.py)

#5 Mis à jour par Benjamin Dauvergne il y a 4 jours

deuxième essaie, j'ai suivi aussi bien que possible la demande:

<http://perso.entreouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&id=05a3e596906b8d539cb1b27b257a6fe3e04e85a2>

J'ai du ajouter un nouveau Widget fichier pour pouvoir gérer la valeur retournée comme un scalaire.

C'est déployé sur <http://echirolles.dev.au-quotidien.com/> (l'admin est grande ouverte, tout le monde peut s'y créer un compte)

#6 Mis à jour par Frédéric Péters il y a 4 jours

(quand tu fais des tests tu peux vérifier que je ne suis pas configuré pour recevoir toutes les tracebacks)

Un nouveau widget, UploadWidget. L'idée même m'ennuie un tout petit peu, dans la mesure où la chaîne retournée peut être assez importante, et peut se retrouver dans la définition d'un formulaire, alors que ces objets on passe dessus pour la page d'accueil, si à la place de passer sur des fichiers qui font moins de 10ko on passe sur des fichiers qui en font des centaines, je crains pour les performances.

À part ça, petite note stylistique, le retour à la ligne des formdef=None dans les

```
- def add_parameters_widgets(self, form, parameters, prefix="):
+ def add_parameters_widgets(self, form, parameters, prefix=",
+     formdef=None):
m'apparait superflu.
```

Aussi, un code comme celui-ci est peut-être amusant à écrire, mais je me poserais des questions quant à sa maintenance:

```
+     variables = (htmltext('<li><span class="variable">[%s]</span></li>') % var for var in formdef.get_varnames())
+     hint = htmltext('<p class="helper variables">%s: <div class="variables"><ul>%s</ul></div></p>') % \
+         (, reduce(htmltext__add__, variables))
(même remarque, un peu atténuée, pour l'utilisation de yield dans get_varnames)
```

#7 Mis à jour par Frédéric Péters il y a 4 jours

Ah et j'oubliais, pour faciliter la relecture, les modifications à send mail et send sms, pour utiliser le nouveau code, tu peux les mettre dans un commit ultérieur ? Merci.

#8 Mis à jour par Benjamin Dauvergne il y a 4 jours

Frédéric Péters a écrit :

(quand tu fais des tests tu peux vérifier que je ne suis pas configuré pour recevoir toutes les tracebacks)

Yep désolé.

Un nouveau widget, UploadWidget. L'idée même m'ennuie un tout petit peu, dans la mesure où la chaîne retournée peut être assez importante, et peut se retrouver dans la définition d'un formulaire, alors que ces objets on passe dessus pour la page d'accueil, si à la place de passer sur des fichiers qui font moins de 10ko on passe sur des fichiers qui en font des centaines, je crains pour les performances.

Mon message de commit est trop succinct: en fait je ne conserve pas le contenu du fichier dans la valeur retournée, le widget s'occupe de sauver le fichier sur le disque (par défaut dans <app_dir>/uploads) et la valeur retournée contient seulement les metadata et le chemin vers le fichier (relatif à l'<app_dir> pour permettre les migrations). C'est juste que tout est géré dans le Widget est pas dans le gestionnaire de submit().

À part ça, petite note stylistique, le retour à la ligne des formdef=None dans les

[...]

m'apparait superflu.

Ok.

Aussi, un code comme celui-ci est peut-être amusant à écrire, mais je me poserais des questions quant à sa maintenance:

[...]

Ok remplacé par:

```
hint = htmltext('%s: <ul>') % _('Usable variables')
for var in formdef.get_varnames():
hint += htmltext('<span class="variable">[%s]</span></li>') % var
hint += htmltext('</ul>')
(même remarque, un peu atténuée, pour l'utilisation de yield dans get_varnames)
```

Ok, j'ai changé ça pour une liste.

#9 Mis à jour par Benjamin Dauvergne il y a 3 jours

Frédéric Péters a écrit :

Ah et j'oubliais, pour faciliter la relecture, les modifications à send mail et send sms, pour utiliser le nouveau code, tu peux les mettre dans un commit ultérieur ? Merci.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&id=e8cecb6bdddffab7c4cdf316e84e5a6bf7a9d84e9>

#10 Mis à jour par Benjamin Dauvergne il y a 3 jours

J'ai modifié les classes UploadedFile/UploadWidget pour ne permettre de ne pas créer trop de fichiers orphelins (je génèrais un nom unique à chaque upload avant, sans jamais effacer les anciens) en laissant l'appelant définir un nom pour le fichier.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=6e407222eefa9b19e8af93878b34a54fa15f9ab5>

J'ai aussi un peu amélioré les messages dans les interfaces et ajouté les traductions en français.

#11 Mis à jour par Thomas Noël il y a un jour

Echéance mis à 30/06/2011

Assigné à mis à Benjamin Dauvergne

#12 Mis à jour par Frédéric Péters il y a environ 8 heures

Ce n'est pas du tout évident à reviewer, sans liste des commits concernés, avec comme seule référence une adresse vers cgit, avec des commits qui peuvent changer à coups de rebase.

"Ne pas créer trop de fichiers orphelins" ça m'ennuie un petit peu, il n'y aurait pas moyen de plutôt "ne pas créer de fichiers orphelins".

Faire confiance à orig_filename, je ne sais pas si c'est une bonne idée, c'est nettoyé en amont par Quixote, ou c'est une valeur sur laquelle le navigateur a un contrôle total ?

Pour créer un fichier unique, pourquoi pas tempfile.mkstemp ?

Aussi, magie non désirable:

```
+ if callable(filename):  
+     self.filename = filename(upload)
```

#13 Mis à jour par Benjamin Dauvergne il y a environ 7 heures

redmine@entrouvert.com écrivait:

Ce n'est pas du tout évident à reviewer, sans liste des commits concernés, avec comme seule référence une adresse vers cgit, avec des commits qui peuvent changer à coups de rebase.

Désolé j'aurai du donner le nouveau nom de la branche.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/log/&h=printing>

"Ne pas créer trop de fichiers orphelins" ça m'ennuie un petit peu, il n'y aurait pas moyen de plutôt "ne pas créer de fichiers orphelins".

Là je n'en crée plus justement, sauf à détruire le formulaire ou le workflow. Sinon il faudrait matérialiser le modèle comme un véritable objet dans w.c.s avec un panneau pour les lister, les supprimer, les ajouter. C'est par là qu'il faut aller à ton avis ?

Faire confiance à orig_filename, je ne sais pas si c'est une bonne idée, c'est nettoyé en amont par Quixote, ou c'est une valeur sur laquelle le navigateur a un contrôle total ?

Je ne me sers pas d'orig_filename mais de base_filename, et pas dans ce cas là.

Pour créer un fichier unique, pourquoi pas tempfile.mkstemp ?
Pas pensé. Je change ça.

Aussi, magie non désirable:

```
+ if callable(filename):  
+     self.filename = filename(upload)
```

C'est inspiré de Django mais ça n'est pas important. C'est parti.

J'ai aussi modifié le listing des variables pour afficher le label des champs et enlever tous les synonymes inutile (suite à discussion avec Victor).

#14 Mis à jour par Frédéric Péters il y a environ 7 heures

S'il n'y a plus de fichiers orphelins lors d'opérations normales, c'est parfait, c'est la formulation "pas trop de" qui m'inquiétait.

Le commit avec les traductions, il ne devrait contenir que fr.po.

orig_filename, s'il n'est pas utilisé, il n'y a pas de sens à le conserver.

Pour le nom du paramètre "model", je le remplacerais en "model_file", dans l'idée qu'on pourra par la suite définir que les champs se terminant par "_file" sont traités spécialement (dans FormDef::get_workflow_with_options par exemple).

C'est développé sur la branche printing de mon dépôt personnel:

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/log/&h=printing>

S'il n'y a plus de fichiers orphelins lors d'opérations normales, c'est parfait, c'est la formulation "pas trop de" qui m'inquiétait.

Le commit avec les traductions, il ne devrait contenir que fr.po.

Fait avec un rebase

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=4a7c868e6c7fa36f164ae1386b70d7c42c37f156>

orig_filename, s'il n'est pas utilisé, il n'y a pas de sens à le conserver.

Fait.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=84104473fcc9210d5cab0191d98ac2a6ee29aaa3>

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=3bad1b10c447c02fee3bc1137022a7f0d9289dc6>

Pour le nom du paramètre "model", je le remplacerais en "model_file", dans l'idée qu'on pourra par la suite définir que les champs se terminant par "_file" sont traités spécialement (dans FormDef::get_workflow_with_options par exemple).

Fait.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=9097f1972ffa3b6b0f728b699767c60c0f7dffe1>

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=bd4f806f1eaa400f35ec2708e360ddeff1bec9ef>

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=f2d6286c729c7874e9349ab88589f5269d7c14e5>

J'ai aussi ajouté les variables créé par template_on_formdata() à la liste des variables affichées:

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=426b164f75f199453cd55906354aac591b70a332>

J'ai créé une sous-classe d'exception PublishError pour renvoyer les erreurs relatives aux templates:

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=12363a6adb3c1e265b34e5506d191f360845b9fa>

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=a81269a6a4976661cebe9dd0e283c87a509970b6>

J'ai ajouté une validation de l'extension du fichier (demande Victor ce jour):

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=5da9d2d3939ecccc54e5624dad48a12488772f1b>

#2 - 01 juin 2011 12:36 - Frédéric Péters

Il y avait aussi une question de ta part :

Pour le nom du paramètre "model", je le remplacerais en "model_file", dans l'idée qu'on pourra par la suite définir que les champs se terminant par "_file" sont traités spécialement (dans FormDef::get_workflow_with_options par exemple).

Ok a priori pour le changement de nom. Mais de quel traitement particulier parles-tu ?

J'étais resté sur une idée que pour le moment il était considéré que les attributs paramétrables d'un workflow soient des chaînes de caractères, mais à relier le code, ce n'est pas forcément le cas, pas de réponse précise ici, du coup.

#3 - 01 juin 2011 12:36 - Frédéric Péters

- *Projet changé de SFD à w.c.s.*

#4 - 01 juin 2011 12:45 - Frédéric Péters

J'ai aussi ajouté les variables créé par template_on_formdata() à la liste des variables affichées:

Ça m'ennuie un peu d'avoir cette liste ainsi, il y a quand même un gros risque qu'elle ne soit pas synchronisée avec des changements ailleurs; donc pour le temps présent j'évitais ça, en attendant une factorisation de la récupération de variables, qui pourrait alors retourner {key: description} ou {key: value}.

J'ai créé une sous-classe d'exception PublishError pour renvoyer les erreurs relatives aux templates:

Tu peux la faire hériter de qommon.errors.PublishError ?

J'ai ajouté une validation de l'extension du fichier (demande Victor ce jour):

Je préférerais de loin une vérification sur le type MIME, avec un fallback sur l'extension en cas de application/octet-stream, et en notant aussi qu'il pourrait arriver que base_filename soit None.

#5 - 02 juin 2011 21:16 - Benjamin Dauvergne

Frédéric Péters a écrit :

J'ai aussi ajouté les variables créé par `template_on_formdata()` à la liste des variables affichées:

Ça m'ennuie un peu d'avoir cette liste ainsi, il y a quand même un gros risque qu'elle ne soit pas synchronisée avec des changements ailleurs; donc pour le temps présent j'évitais ça, en attendant une factorisation de la récupération de variables, qui pourrait alors retourner `{key: description}` ou `{key: value}`.

Ces variables sont quand même utiles, alors je les ai déplacé près de `template_on_formdata()` où ces variables sont créés, ça devrait prévenir ce problème de maintenance jusqu'à une future factorisation; le mieux ce serait effectivement que ces variables soient ajouté par `FormData.get_as_dict()` et pas dans `template_on_formdata()`.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=c197efbfa3c0fb4c127f00872d5b01ce5ed7dbc0>

J'ai créé une sous-classe d'exception `PublishError` pour renvoyer les erreurs relatives aux templates:

Tu peux la faire hériter de `qommon.errors.PublishError` ?

Ok.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=6950454bec0cde0c19b07546a3ca1ceea387e16d>

J'ai ajouté une validation de l'extension du fichier (demande Victor ce jour):

Je préférerais de loin une vérification sur le type MIME, avec un fallback sur l'extension en cas de `application/octet-stream`, et en notant aussi qu'il pourrait arriver que `base_filename` soit `None`.

Ok.

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=5f66e166cb72383e6b70fab87b310157feed034d>

#6 - 03 juin 2011 15:24 - Benjamin Dauvergne

Frédéric Péters a écrit :

J'étais resté sur une idée que pour le moment il était considéré que les attributs paramétrables d'un workflow soient des chaînes de caractères, mais à relire le code, ce n'est pas forcément le cas, pas de réponse précise ici, du coup.

Moi aussi au début mais finalement je n'ai pas vu la raison, d'ailleurs avec le recul je ne vois pas bien la raison qui t'a poussé à faire le commit r2067.

#7 - 04 juin 2011 23:35 - Benjamin Dauvergne

J'ai ajouté la possibilité de garder la valeur actuelle du modèle; `UploadWidget` ressemble de plus en plus à `FileWithPreviewWidget`, il y aura de la factorisation pour plus tard. J'ai pas voulu l'utiliser au début car le code me semblait un peu compliqué et je préfère ne pas y toucher pour l'instant de peur de casser.

- <http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=3199ebbbd40c69030b41028605fa40741d49e400> modification de `UploadWidget`
- <http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=ec001a0d2fb71925b091406e798927ddecfecea8> utilisation dans `workflows.py`
- <http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=printing&id=ec9b9b241ff2cff38d6cd9f5de3a796433d17ddc> mise à jour des traductions

#8 - 12 juin 2011 17:54 - Frédéric Péters

Benjamin, deux derniers, je pense, commentaires, (je n'ai pas relu tous les patches),

1. vraiment, le dictionnaire `ADDITIONAL_VARIABLES`, je ne l'incluerais pas, le sujet est discuté dans [#471](#)
2. tu peux remplacer les `super()` par des `base.__init__(self...)` là où possible ?

#9 - 12 juin 2011 20:13 - Benjamin Dauvergne

Frédéric Péters a écrit :

Benjamin, deux derniers, je pense, commentaires, (je n'ai pas relu tous les patches),

1. vraiment, le dictionnaire ADDITIONAL_VARIABLES, je ne l'incluais pas, le sujet est discuté dans [#471](#)

ok. <http://perso.entrouvert.org/~bdauvergne/git/cgi.cgi?url=wcs-perso/commit/&h=printing&id=e53186efc806512d4d307f16886e77f9abf3e564>

1. tu peux remplacer les super() par des base.__init__(self...) là où possible ?

ok mais à chaque fois qu'on fait ça un chaton meurt.

<http://perso.entrouvert.org/~bdauvergne/git/cgi.cgi?url=wcs-perso/commit/&h=printing&id=1d3ead79224755533c4900e3b7f01031a843921a>

#10 - 21 juin 2011 17:18 - Frédéric Péters

s/supper/super/

Et tu peux en supprimer le commit ajoutant receipt_time et now ? Ça doit être géré différemment.

Et ensuite merger.

#11 - 22 juin 2011 02:44 - Benjamin Dauvergne

Commit retiré, branche « mergé ».

#12 - 22 juin 2011 08:23 - Frédéric Péters

- Statut changé de Nouveau à Fermé

Sans doute je n'étais pas clair, mais s/supper/super/, ça indiquait un message de commit à corriger. Tant pis.

Et il semble que la fermeture automatique des tickets ne fonctionne pas.

#13 - 22 juin 2011 10:57 - Benjamin Dauvergne

redmine@entrouvert.com écrivait:

La demande [#465](#) a été mise à jour par Frédéric Péters.

Statut changé de Nouveau à Fermé

Sans doute je n'étais pas clair, mais s/supper/super/, ça indiquait un message de commit à corriger. Tant pis.

Et il semble que la fermeture automatique des tickets ne fonctionne pas.

J'ai du me planter dans la syntaxe, je l'ai déjà vu fonctionner.

#14 - 24 juin 2011 12:30 - Thomas Noël

- Statut changé de Fermé à En cours

Attention, il faudra fermer ce ticket seulement quand on aura terminé le système des variables de substitution, et qu'il sera intégré à cet export rtf.

#15 - 24 juin 2011 12:34 - Frédéric Péters

Je laissais ce côté au ticket [#471](#), mais c'est comme tu le sens.

#16 - 24 juin 2011 12:50 - Victor Claudet

Si on renseigne le document d'export au niveau du workflow, il y a une erreur (voir le traceback).

Si on laisse le document vide au niveau du workflow mais qu'on le renseigne au niveau du formulaire ça marche nickel.

erreur, j'ai testé sur la version qui n'inclue pas le [now] et le [receipt_time]

#17 - 27 juin 2011 10:54 - Frédéric Péters

- Statut changé de En cours à Fermé

Le support pour les variables de substitution a été ajouté.

Author: fpeters

Date: 2011-06-27 08:49:29 +0000 (Mon, 27 Jun 2011)

New Revision: 2155

Log:

Add support for substitution variables to workflows.

- This allows variables to be used in exported models (#465)
- This allows access to the previous status in emails (#447)
- This allows a workflow field to reference a form field (#248)
- This allows a workflow field to reference a user field (#249)