w.c.s. - Bug #8402

Ajouter la possibilité de valider les fichiers attachés en utilisant fargo pour le stockage

28 septembre 2015 17:18 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	28 septembre 2015
Priorité:	Haut	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:	v1.22		
Patch proposed:	Oui	**Planning:**	

Description

Il doit être possible de voir l'état de validation d'un fichier et de valider celui-ci.

Demandes liées:

Lié à Fargo - Development #8169: Interface de validation des fichiers	**Fermé**	**03 septembre 2015**

Révisions associées

Révision c2b87bbe - 10 novembre 2015 18:39 - Benjamin Dauvergne

formdef: allow field classes to have migrate methods (#8402)

Allow modifying the storage of fields.

Révision 0a6d2488 - 10 novembre 2015 18:39 - Benjamin Dauvergne

fields: handle non string value in export_to_xml() (#8402)

Fields using this facility must override import_from_xml() to convert the imported string value to the real type, for example a list of strings converted to CSV, must be splitted.

Révision 9f9e9685 - 10 novembre 2015 18:40 - Benjamin Dauvergne

add support for file validation (#8402)

Révision 9c4be6b9 - 10 novembre 2015 18:40 - Benjamin Dauvergne

formdef: rewrite update_mimetypes as update_filetype (#8402)

As filetype/mimetypes handling has been changed, handling of settings' filetypes changes must be updated.

Révision caf373d5 - 11 novembre 2015 17:10 - Benjamin Dauvergne

root: add reload_top to _q_exports to mimic w.c.s. (#8402)

Historique

#1 - 28 septembre 2015 17:25 - Benjamin Dauvergne

- Version cible mis à v1.14

- Patch proposed changé de Non à Oui

Je propose un premier jet qui n'est pas encore ce qui sera commité:

- tout ce qui est spécifique à fargo est ajouté dans wcs/file_validation.py, appels au différents web services particulièrement, génération du contenu du bloc "validation" qui accompagnera le résumé d'une demande pour un champ fichier, détection de la possibilité ou non de valider des fichiers
- des modifications sont apportées à wcs.fields.FileField pour permettre de choisir un type de document à associer au champ, si c'est possible,
- des modifications sont apportées à wcs/forms/common.py pour ajouter dans le résumé d'une demande au niveau des champs fichiers, un bloc donnant l'état de la validation de ce fichier
- le fichier fargo.js a été repris de auquotidien

Ce qui reste à faire:

- cacher le bloc en front-office, je ne savais pas la meilleure manière de faire cela (j'ai vu du 'frontoffice' in get_response().breadcrumb(), ça m'a fait un peu peur) et ne l'afficher en back-office qu'aux agents autorisés
- créer des variables de substitution pour pouvoir coder dans les workflows la condition que le fichier a été validé, interdisant par exemple

d'avancer si ce n'est pas fait, ou encore de récupérer les métadonnées
- transférer complètement le code du support fargo d'auquotidien dans w.c.s. puisque déjà le fichier .js est là
- implémenter la partie cliente de l'envoie dans fargo d'un fichier qui n'y est pas en même temps que sa validation

#2 - 28 septembre 2015 17:25 - Benjamin Dauvergne

- Lié à Development #8169: Interface de validation des fichiers ajouté

#3 - 28 septembre 2015 17:25 - Benjamin Dauvergne

- Fichier 0001-add-support-for-file-validation.patch ajouté

#4 - 28 septembre 2015 18:05 - Frédéric Péters

On a déjà un paramètre "Types de fichier" dans les champs de type fichier, là on ajoute un "Types de documents"; je verrais plutôt l'extension des "Types de fichier" existants avec les infos de Fargo. (genre ajouter dans le paramétrage de "types de fichier" une sélection parmi les types proposés par Fargo).

Pour savoir si on est dans le backoffice il y a désormais un get_request().is_in_backoffice().

#5 - 28 septembre 2015 18:27 - Frédéric Péters

> Pour savoir si on est dans le backoffice il y a désormais un get_request().is_in_backoffice().

Qui arrive dans #8306.

#6 - 28 septembre 2015 18:37 - Benjamin Dauvergne

Frédéric Péters a écrit :

> On a déjà un paramètre "Types de fichier" dans les champs de type fichier, là on ajoute un "Types de documents"; je verrais plutôt l'extension des "Types de fichier" existants avec les infos de Fargo. (genre ajouter dans le paramétrage de "types de fichier" une sélection parmi les types proposés par Fargo).

Il me semble que type de fichier est un champ multiple, j'ai vraiment besoin que le type de document soit unique, la complexité que cela introduirait d'avoir des types de document multiple pour un champ ne me semble pas souhaitable. Ça veut dire refaire le web-service des métadonnées pour accepter une liste de types au lieu d'un seul et de renvoyer éventuellement une liste de structure métadonnées au lieu d'une seule. Coté backoffice w.c.s je vais devoir proposer plusieurs liens ou permettre dans le lien vers le web-service de passer cette liste de types de documents et ainsi restreindre un menu déroulant dans l'écran de validation pour y choisir le bon type, et ensuite modifier mon formulaire en fonction du type choisi. Pour moi le jeu n'en vaut pas la chandelle.

Est-ce qu'on pourrait se permettre de revenir à un champs de fichier unique ? Si on a besoin d'un champ autorisant soit des vidéos soit des images et bien on adaptera la liste des types MIME en conséquence plutôt que de permettre de combiner les types techniques "Fichier Vidéos" et "Fichier Images".

> Pour savoir si on est dans le backoffice il y a désormais un get_request().is_in_backoffice().

Ok.

#7 - 28 septembre 2015 19:22 - Frédéric Péters

Ok, c'est que j'ai lu trop vite et il m'avait semble que ça gérait déjà la sélection de plusieurs types; ça ne m'ennuierait pas trop de limiter à un seul élément la sélection, je viens de faire une inspection rapide et je trouve un seul formulaire en circulation qui utilise plusieurs types (demarches.vincennes.fr/formdefs/163).

Cela étant, la migration pour passer sur un type de fichier unique n'est pas nécessairement aisée, le plus simple ça doit sans doute être de continuer à utiliser une liste mais de forcer celle-ci à contenir un seul élément (le CheckboxesWidget a déjà un paramètre max_choices, qu'on mettrait à 1).

Par la suite, le CheckboxesWidget pourrait gérer le cas particulier du max_choices mis à 1 et basculer sur des boutons radios dans cette circonstance.

#8 - 02 octobre 2015 13:10 - Benjamin Dauvergne

Question vu avec Fred:

> Préfères-tu que je génère directement des types de fichier depuis les types de document fargo, ou bien ajouter un menu déroulant pour laisser la personne qui configure w.c.s. associer un type de document à un type de fichier; en somme est-ce que je laisse un indirection au risque de perturber les gens parce qu'il y a deux concepts différents, ou je mélange les deux choses (et je rajouterai un champ slug au type de fichier pour garder la référence vers la définition dans fargo et dans fargo j'ajoute la possibilité de déclarer des types MIME)

Ok pour générer dixit Fred.

#9 - 02 octobre 2015 17:02 - Benjamin Dauvergne

- Fichier 0001-formdef-allow-field-classes-to-have-migrate-methods-.patch ajouté

- Fichier 0002-add-support-for-file-validation-8402.patch ajouté

- Statut changé de Nouveau à En cours

#10 - 02 octobre 2015 17:03 - Benjamin Dauvergne

J'ai ajouté une migration sur la classe FileField pour renommer file_type en document_type et en changer le format (je stocke un dico décrivant tous les détails du type de fichier, mimetypes, si c'est lié à fargo, etc..).

#11 - 02 octobre 2015 17:04 - Benjamin Dauvergne

Si c'est ok je regarderai pour faire un test de la migration.

#12 - 02 octobre 2015 17:15 - Frédéric Péters

Comme je le notais plus haut je préférerais vraiment conserver une liste, et poser un max_choices=1 sur le CheckboxesWidget.

#13 - 02 octobre 2015 20:43 - Benjamin Dauvergne

Ce n'est juste pas possible, actuellement ce n'est pas la référence au type d'objet qui est stocké dans la liste, c'est directement des types MIME.

#14 - 02 octobre 2015 20:56 - Benjamin Dauvergne

Et utiliser un CheckboxesWidget n'a pas été possible non plus parce que j'ai une donnée complexe maintenant hors les CheckboxesWidget ne gère que (clé, label) et la mécanique des formulaires d'admin n'admet pas de transformation sur un POST de formulaire qui me permettrait de convertir les clés en valeur. Le SingleSelectWidget me permet de passer du (valeur, label, clé) et d'avoir directement la valeur de l'objet 'filetype' au retour de parse(). Je pense vraiment que j'ai limité les modifications en faisant cela. C'est la nécessité du migrate qui te pose souci ?

#15 - 02 octobre 2015 21:25 - Frédéric Péters

Ok pour ça; donc je regarde la fonction migrate().

> if self.file_type == value['mimetypes']:

Mais self.file_type, avant, c'est nécessairement une liste, et value['mimetypes'], maintenant, c'est nécessairement une chaine de caractère (à regarder les exemples dans get_default_document_types()); non ?

Mais self.file_type (bis), c'est de toute façon pas l'ancien attribut mais une propriété qui retourne self.document_type.get('mimetypes').

~~

Pour aider les tests, voici une valeur réelle de cfg['filetypes'] (demarches.vincennes.fr)

```
{1: {'mimetypes': ['application/pdf', 'application/vnd.oasis.opendocument.text', 'application/msword', 'applic
ation/vnd.openxmlformats-officedocument.wordprocessingml.document', 'application/vnd.oasis.opendocument.spread
sheet', 'application/vnd.ms-excel', 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet'], 'lab
el': 'Document'}}
```

et le *dict* d'un champ de formulaire :

```
>>> f.fields[10].__dict__
{'extra_css_class': None, 'max_file_size': None, 'prefill': {'type': 'none'}, 'hint': None, 'file_type': ['ima
ge/*', 'application/pdf,application/vnd.oasis.opendocument.text,application/msword,application/vnd.openxmlform
ats-officedocument.wordprocessingml.document,application/vnd.oasis.opendocument.spreadsheet,application/vnd.ms
-excel,application/vnd.openxmlformats-officedocument.spreadsheetml.sheet'], 'varname': None, 'required': False
, 'label': 'Joindre un document', 'in_listing': False, 'type': 'file', 'id': '15'}
```

~~

get_admin_attributes, pas besoin d'une variable intermédiaire.

L'attribut document_type, il faudrait l'initialiser à {} sinon je pense que l'import/export xml peut échouer.

~~

wcs/file_validation.py : manque l'entête gpl. Le json_encode_helper() on l'a maintenant dans misc (il me semblait avoir vu un patch l'utilisant). Détail, entre les fonctions, une seule ligne. La ligne 108 est bien trop longue et les chaines devraient être traduites individuellement, il faut éviter le markup dedans.

Les fonctions qui retournent une valeur, je préfère quand elles sont explicites dans leur "return None" (pas juste "return", pas juste laisser la fonction se terminer).

À ce sujet, validation_link pourrait retourner une chaine vide, ça permettrait un systématique r += self.file_validation_status(f, value) dans common.py

~~

Dans les points "ce qui reste à faire" du premier commentaire, il y a des trucs qui peuvent aller dans d'autres tickets mais il faudrait au moins ne pas afficher la validation hors backoffice.

~~

#16 - 03 octobre 2015 11:06 - Frédéric Péters

Dans les points qui restent à faire, un truc qui n'a pas été identifié et est pourtant important, c'est qu'on permet à l'admin d'aller modifier les types de fichier dans settings/filetypes/, et qu'une fois modifié, il faut aller mettre à jour l'ensemble des formdefs (pour le moment cf FormDef.update_mimetypes).

(il y a déjà un test qui vérifie que ça fonctionne correctement). (test_admin_pages/test_settings_filetypes_update)

#17 - 03 octobre 2015 17:14 - Benjamin Dauvergne

On pourrait aussi ne conserver qu'une référence et ne plus se prendre le choux (je fais déjà un migrate), il restera juste le cas d'un filetype supprimé.

#18 - 03 octobre 2015 17:36 - Frédéric Péters

Ce n'est volontairement pas une référence pour permettre à l'import/export de fonctionner sans avoir à se soucier du paramétrage. C'était plus facile ainsi mais ça pourrait être une référence, l'export devrait alors quand même contenir l'intégralité du paramétrage et à l'import il y aurait à trouver l'entrée de paramétrage qui corresponde, et à la créer si jamais celle-ci n'existait pas.

#19 - 03 octobre 2015 21:04 - Benjamin Dauvergne

Je n'avais pas pensé à l'import/export, en regardant export_to_xml je me dis qu'actuellement document_type n'est pas exportable, car c'est un dico mais une des clés a pour valeur une liste, la clé mimetypes... Il faut soit que je remplace complètement export_to_xml sur FileField (je ne pense pas que je puisse le spécialiser pour un seul champ) soit que je change de manière de stocker pour quelque chose de plus à plat.

#20 - 04 octobre 2015 14:22 - Benjamin Dauvergne

- Fichier 0001-formdef-allow-field-classes-to-have-migrate-methods-.patch ajouté

- Fichier 0002-fields-in-export_to_xml-handle-non-string-value-for-.patch ajouté

- Fichier 0003-add-support-for-file-validation-8402.patch ajouté

Nouveaux patchs, j'ai contourné le problème de mimetypes (et du champ fargo booléen) en surchargeant export/import sur FileField pour transformer le champ mimetypes en chaînes de valeurs séparées par des '|' (chaque valeur est déjà une liste de valeurs séparées par des virgules, pour conserver les groupement histoire que assert_..._import_export_works passe, j'ai du faire ce choix) ensuite pour que l'import de {'fargo': True} fonctionne j'ai ajouté à FileField.export_to_xml() le fait de faire un unicode(v) au lieu de unicode(v, charset) pour des valeurs qui ne sont pas des chaînes simples.

Mais vu de loin ça fait beaucoup de code pour un bénéfice pas terrible, si tu vois quelquechose de plus simple (et aussi des améliorations aux tests d'import/export j'ai regardé les tests plus haut et recopié ce qui me paraissait pertinent).

Je n'ai pas implémenté de fait de recréer le filetype dans cfg['filetypes'], pour l'instant je m'en sors en introduisant dans les valeurs du SingleSelectWidget la valeur en cours si elle n'existe pas dans cfg['filetypes']. Dis moi si tu penses vraiment important d'ajouter à FileField.init_with_json/xml de modifier cfg['filetypes'].

#21 - 04 octobre 2015 14:24 - Benjamin Dauvergne

Je n'ai pas traité le update_mimetypes, je m'y attelle.

#22 - 04 octobre 2015 14:41 - Benjamin Dauvergne

- Fichier 0004-formdef-rewrite-update_mimetypes-as-update_filetype-.patch ajouté

Voilà update_mimetypes renommé en update_filetype, ça se base sur l'id du filetype pour relier ce qui est stocké dans field.document_type et ce qui est stocké dans cfg['filetypes'].

#23 - 05 octobre 2015 16:15 - Frédéric Péters

Première visite sur la page d'accueil après les patchs, exécution du code de migration :

```
Exception:
  type = '<type 'exceptions.KeyError'>', value = ''mimetypes''

Stack trace (most recent call first):
```

```
  File "/home/fred/src/eo/wcs/wcs/fields.py", line 777, in migrate
   775                 document_types = self.get_document_types()
   776                 for key, value in document_types.iteritems():
>  777                     if self.file_type == value['mimetypes']:
   778                         self.document_type = value.copy()
   779                         self.document_type['id'] = key

  locals:
     file_type = 'image/*'
     self = <wcs.fields.FileField object at 0x7fee0c608550>
     key = 'justificatif-de-domicile'
     value = {'fargo': True, 'id': 'justificatif-de-domicile', 'label': 'Justificatif de domicile'}
     document_types = {'_image': {'mimetypes': ['image/*'], 'id': '_image', 'label': 'Fichiers image'}, '_vide
o': {'mimetypes': ['video/*'], 'id': '_video', 'label': 'Fichiers vid\xc3\xa9o'}, 'justificatif-de-domicile':
{'fargo': True, 'id': 'justificatif-de-domicile', 'label': 'Justificatif de domicile'}, '_audio': {'mimetypes'
: ['audio/*'], 'id': '_audio', 'label': 'Fichiers son'}}
```

#24 - 05 octobre 2015 17:12 - Benjamin Dauvergne

- Fichier 0003-add-support-for-file-validation-8402.patch ajouté

Utilisation de python-mock pour améliorer mon test voir le bug que tu as trouvé:

```
diff --git a/tests/test_formdef.py b/tests/test_formdef.py
index 6c0c444..7f0e1e9 100644
--- a/tests/test_formdef.py
+++ b/tests/test_formdef.py
@@ -4,6 +4,8 @@ import shutil

 import pytest

+from mock import patch
+
 from quixote import cleanup
 from wcs import formdef
 from wcs.formdef import FormDef
@@ -134,15 +136,23 @@ def test_substitution_variables_object():
 from wcs.formdef import FormDef
@@ -134,15 +136,23 @@ def test_substitution_variables_object():
         assert substs.foobar

 def test_file_field_migration():
-    formdef = FormDef()
-    formdef.name = 'foo'
-    file_type = ['image/*', 'application/pdf,application/vnd.oasis.opendocument.text,application/msword,appli
cation/vnd.openxmlformats-officedocument.wordprocessingml.document,application/vnd.oasis.opendocument.spreadsh
eet,application/vnd.ms-excel,application/vnd.openxmlformats-officedocument.spreadsheetml.sheet']
-    formdef.fields = [FileField(type='file', id='1', label='file')]
-    formdef.fields[0].__dict__['file_type'] = file_type
-    formdef.store()
-    formdef = FormDef.get(1)
-    assert 'file_type' not in formdef.fields[0].__dict__
-    assert formdef.fields[0].document_type
-    assert formdef.fields[0].document_type['id'] == '_legacy'
-    assert formdef.fields[0].document_type['mimetypes'] == ['image/*', 'application/pdf,application/vnd.oasis
.opendocument.text,application/msword,application/vnd.openxmlformats-officedocument.wordprocessingml.document,
application/vnd.oasis.opendocument.spreadsheet,application/vnd.ms-excel,application/vnd.openxmlformats-officed
ocument.spreadsheetml.sheet']
-    assert formdef.fields[0].document_type['label'] == ','.join(file_type)
+    with patch('wcs.file_validation.get_document_types') as get_document_types:
+        get_document_types.return_value = {
+                'justificatif-de-domicile': {
+                    'id': 'justificatif-de-domicile',
+                    'label': 'Justificatif de domicile',
+                    'fargo': True,
+                },
+        }
+        formdef = FormDef()
+        formdef.name = 'foo'
+        file_type = ['image/*', 'application/pdf,application/vnd.oasis.opendocument.text,application/msword,a
pplication/vnd.openxmlformats-officedocument.wordprocessingml.document,application/vnd.oasis.opendocument.spre
adsheet,application/vnd.ms-excel,application/vnd.openxmlformats-officedocument.spreadsheetml.sheet']
+        formdef.fields = [FileField(type='file', id='1', label='file')]
+        formdef.fields[0].__dict__['file_type'] = file_type
+        formdef.store()
```

```
+        formdef = FormDef.get(1)
+        assert 'file_type' not in formdef.fields[0].__dict__
+        assert formdef.fields[0].document_type
+        assert formdef.fields[0].document_type['id'] == '_legacy'
+        assert formdef.fields[0].document_type['mimetypes'] == ['image/*', 'application/pdf,application/vnd.o
asis.opendocument.text,application/msword,application/vnd.openxmlformats-officedocument.wordprocessingml.docum
ent,application/vnd.oasis.opendocument.spreadsheet,application/vnd.ms-excel,application/vnd.openxmlformats-off
icedocument.spreadsheetml.sheet']
+        assert formdef.fields[0].document_type['label'] == ','.join(file_type)
```

La sortie avant fix:

```
(wcs)bdauvergne@dor-lomin:~/Code/wcs$ PYTHONPATH=$(pwd) py.test --without-postgresql-tests --capture=no -k mig
ration tests/test_formdef.py
============================================================================= test session starts ============
==================================================================
platform linux2 -- Python 2.7.9 -- py-1.4.30 -- pytest-2.7.2
rootdir: /home/bdauvergne/Code/wcs, inifile:
collected 9 items

tests/test_formdef.py F

================================================================================== FAILURES ==================
==================================================================
_____________________________________________________________________________ test_file_field_migration __________
__________________________________________________________________

    def test_file_field_migration():
        with patch('wcs.file_validation.get_document_types') as get_document_types:
            get_document_types.return_value = {
                    'justificatif-de-domicile': {
                        'id': 'justificatif-de-domicile',
                        'label': 'Justificatif de domicile',
                        'fargo': True,
                    },
            }
            formdef = FormDef()
            formdef.name = 'foo'
            file_type = ['image/*', 'application/pdf,application/vnd.oasis.opendocument.text,application/mswor
d,application/vnd.openxmlformats-officedocument.wordprocessingml.document,application/vnd.oasis.opendocument.s
preadsheet,application/vnd.ms-excel,application/vnd.openxmlformats-officedocument.spreadsheetml.sheet']
            formdef.fields = [FileField(type='file', id='1', label='file')]
            formdef.fields[0].__dict__['file_type'] = file_type
            formdef.store()
>           formdef = FormDef.get(1)

tests/test_formdef.py:153:
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
wcs/qommon/storage.py:279: in get
    ignore_migration=ignore_migration)
wcs/qommon/storage.py:361: in get_filename
    o.migrate()
wcs/formdef.py:193: in migrate
    changed |= f.migrate()
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

self = <wcs.fields.FileField object at 0x7f1801ee3b10>

    def migrate(self):
        if 'file_type' in self.__dict__:
            self.document_type = {}
            if self.__dict__['file_type']:
                file_type = self.__dict__['file_type']
                document_types = self.get_document_types()
                for key, value in document_types.iteritems():
>                   if self.file_type == value['mimetypes']:
E                   KeyError: 'mimetypes'

wcs/fields.py:777: KeyError
================================================================== 8 tests deselected by '-kmigration' ====
==================================================================
=================================================================== 1 failed, 8 deselected in 1.35 seconds ==
==================================================================
```

Le fix:

```
diff --git a/wcs/fields.py b/wcs/fields.py
index 00ce14f..7aac3f2 100644
--- a/wcs/fields.py
+++ b/wcs/fields.py
@@ -774,7 +774,7 @@ class FileField(WidgetField):
                 file_type = self.__dict__['file_type']
                 document_types = self.get_document_types()
                 for key, value in document_types.iteritems():
-                    if self.file_type == value['mimetypes']:
+                    if self.file_type == value.get('mimetypes'):
                         self.document_type = value.copy()
                         self.document_type['id'] = key
                 else:
```

Le patch mise à jour avec tout ça.

#25 - 05 octobre 2015 17:55 - Frédéric Péters

Je propose ça pour ne pas avoir une vue du frontoffice dans la popup :

```
diff --git a/wcs/root.py b/wcs/root.py
index 8eb6ee3..9382c9d 100644
--- a/wcs/root.py
+++ b/wcs/root.py
@@ -309,9 +309,8 @@ class RootDirectory(Directory):
         return forms.root.RootDirectory()._q_lookup(component)

     def reload_top(self):
-        r = TemplateIO(html=True)
-        r += htmltext('<script>window.top.document.location.reload();</script>')
-        return r.getvalue()
+        get_response().filter = {}
+        return htmltext('<html><body><script>window.top.document.location.reload();</script></body></html>')

     admin = None
     backoffice = None
```

~~

Il faudra penser à ajouter le ('reload-top', 'reload_top') dans auquotidien (j'ai le patch en local).

~~

À part ça, je n'afficherais pas le lien de validation pour les documents qui sont déjà validés; aussi je trouverais super d'avoir l'info sur le statut de validation (quand c'est pertinent) dans une classe du d'au-dessus.

```
            r += htmltext('<div class="value">')  # <--- ce <div> là.
            s = f.get_view_value(value)
            s = s.replace(str('[download]'), str('%sdownload' % form_url))
            r += s
            if isinstance(f, FileField) and get_request().is_in_backoffice():
                r += htmltext(self.file_validation_status(f, value))
```

La classe pourrait être posée qu'on soit ou pas dans le backoffice.

~~

(la description du patch 0002- a une longue trop longue, faudrait rewrapper, mais par contre le commentaire dans wcs/fields.py a lui des lignes trop courtes (ça tient sur deux lignes en respectant la limite à 80 caractères). (et s/overload/overload/)

~~

Pour les tests j'ai donné plus haut un extrait d'une vraie valeur trouvée à Vincennes, il pourrait y avoir un test sur celle-ci ?

#26 - 06 octobre 2015 16:45 - Thomas Noël

- Version cible changé de v1.14 à v1.15

#27 - 07 octobre 2015 11:25 - Thomas Noël

- Version cible changé de v1.15 à v1.16

#28 - 07 octobre 2015 13:13 - Frédéric Péters

- Version cible changé de v1.16 à v1.17

#29 - 11 octobre 2015 17:45 - Frédéric Péters

- Version cible changé de v1.17 à v1.18

#31 - 28 octobre 2015 18:20 - Benjamin Dauvergne

- Priorité changé de Normal à Haut

#32 - 29 octobre 2015 20:02 - Benjamin Dauvergne

- Fichier 0001-formdef-allow-field-classes-to-have-migrate-methods-.patch ajouté

- Fichier 0002-fields-handle-non-string-value-in-export_to_xml-8402.patch ajouté

- Fichier 0003-add-support-for-file-validation-8402.patch ajouté

- Fichier 0004-formdef-rewrite-update_mimetypes-as-update_filetype-.patch ajouté

- Fichier 0001-root-add-reload_top-to-_q_exports-to-mimic-w.c.s.-84.patch ajouté

Frédéric Péters a écrit :

> Je propose ça pour ne pas avoir une vue du frontoffice dans la popup :
>
> [...]

Ok.

> ~~
>
> Il faudra penser à ajouter le ('reload-top', 'reload_top') dans auquotidien (j'ai le patch en local).

Patch attaché en dernier.

> ~~
>
> À part ça, je n'afficherais pas le lien de validation pour les documents qui sont déjà validés; aussi je trouverais super d'avoir l'info sur le statut de validation (quand c'est pertinent) dans une classe du d'au-dessus.

Ok.

> [...]
>
> La classe pourrait être posée qu'on soit ou pas dans le backoffice.

Ok.

> ~~
>
> (la description du patch 0002- a une longue trop longue, faudrait rewrapper, mais par contre le commentaire dans wcs/fields.py a lui des lignes trop courtes (ça tient sur deux lignes en respectant la limite à 80 caractères). (et s/overload/overload/)

Ok.

> ~~
>
> Pour les tests j'ai donné plus haut un extrait d'une vraie valeur trouvée à Vincennes, il pourrait y avoir un test sur celle-ci ?

Il y est déjà:

```
+        file_type = ['image/*', 'application/pdf,application/vnd.oasis.opendocument.text,application/msword,a
pplication/vnd.openxmlformats-officedocument.wordprocessingml.document,application/vnd.oasis.opendocument.spre
adsheet,application/vnd.ms-excel,application/vnd.openxmlformats-officedocument.spreadsheetml.sheet']
+        formdef.fields = [FileField(type='file', id='1', label='file')]
```

#33 - 05 novembre 2015 20:56 - Frédéric Péters

- Statut changé de En cours à Fermé

#34 - 05 novembre 2015 21:35 - Benjamin Dauvergne

Euh je n'ai toujours rien poussé, j'attends un ack.

#35 - 05 novembre 2015 22:09 - Frédéric Péters

- Statut changé de Fermé à En cours

- Version cible v1.18 supprimé

#36 - 06 novembre 2015 11:17 - Frédéric Péters

Dans 0002, s/overloaf/overload/.

Dans 0003, dans les tests, en ajouter un dans test_backoffice_pages et un autre dans test_form_pages, pour couvrir display_file_field.

#37 - 06 novembre 2015 13:09 - Benjamin Dauvergne

- Fichier 0002-fields-handle-non-string-value-in-export_to_xml-8402.patch ajouté

Frédéric Péters a écrit :

> Dans 0002, s/overloaf/overload/.

Ok.

> Dans 0003, dans les tests, en ajouter un dans test_backoffice_pages et un autre dans test_form_pages, pour couvrir display_file_field.

Je viens de faire un git grep display_file_field dans mon dépôt w.c.s. et je ne trouve rien, alors je veux bien des explications sur ce point.

#38 - 06 novembre 2015 14:47 - Frédéric Péters

Dans 0003 :

```
            if isinstance(f, FileField):
                r += htmltext(self.display_file_field(f, value))
```

#39 - 06 novembre 2015 14:54 - Benjamin Dauvergne

Ah oui dsl je ne devais pas être dans la bonne branche, faudra que je pense à traffiquer un peu mon zsh.

#40 - 10 novembre 2015 08:43 - Benjamin Dauvergne

- Fichier 0003-add-support-for-file-validation-8402.patch ajouté

#41 - 10 novembre 2015 09:02 - Frédéric Péters

```
    with file(os.path.join(pub.app_dir, 'site-options.cfg'), 'w') as site_option:
        site_option.write('''[options]
fargo_url = http://fargo.example.net/
''')
```

Ça va écraser le site-options.cfg qui peut exister quand les tests s'exécutent en mode SQL.

Par ailleurs, dans test_form_pages, je dois relire ce qu'on écrivait mais il me semblait qu'on n'affichait rien concernant la validité des pièces, du coup, pourquoi un appel à fargo ?

#42 - 10 novembre 2015 09:05 - Benjamin Dauvergne

Frédéric Péters a écrit :

> [...]
>
> Ça va écraser le site-options.cfg qui peut exister quand les tests s'exécutent en mode SQL.

Ok donc test de présence/lecture/écriture ?

> Par ailleurs, dans test_form_pages, je dois relire ce qu'on écrivait mais il me semblait qu'on n'affichait rien concernant la validité des pièces, du coup, pourquoi un appel à fargo ?

Tu disais:

> À part ça, je n'afficherais pas le lien de validation pour les documents qui sont déjà validés; aussi je trouverais super d'avoir l'info sur le statut de validation (quand c'est pertinent) dans une classe du d'au-dessus.
> [...]
> La classe pourrait être posée qu'on soit ou pas dans le backoffice.

Pour pouvoir afficher cette classe en front-office, je dois faire un appel à Fargo.

#43 - 10 novembre 2015 09:19 - Benjamin Dauvergne

- Fichier 0003-add-support-for-file-validation-8402.patch ajouté

Initialisation de fargo_url déplacé dans une fixture globale (avec finalizer pour défaire) dans conftest.py (j'ai eu un doute que ça se combine bien avec les fixture "pub" réparties un peu partout ailleurs, mais oui ça marche, pytest c'est bien).

#44 - 10 novembre 2015 13:45 - Frédéric Péters

Ça passe chez toi ?

```
tests/test_backoffice_pages.py::test_backoffice_file_field_validation[pickle] FAILED
tests/test_backoffice_pages.py::test_backoffice_file_field_validation[sql] FAILED
...
tests/test_form_pages.py::test_file_field_validation[pickle] FAILED
tests/test_form_pages.py::test_file_field_validation[sql] FAILED
...
```

#45 - 10 novembre 2015 15:04 - Benjamin Dauvergne

Ouaip :

```
(wcs)bdauvergne@dor-lomin:~/Code/wcs$ PYTHONPATH=$(pwd) py.test -vv -k file_field --capture=no tests
============================================================================= test session starts =============
==================================================================
platform linux2 -- Python 2.7.9 -- py-1.4.30 -- pytest-2.6.3 -- /home/bdauvergne/.virtualenvs/wcs/bin/python2
plugins: random
collected 729 items

tests/test_backoffice_pages.py::test_backoffice_file_field_validation[pickle] PASSED
tests/test_backoffice_pages.py::test_backoffice_file_field_validation[sql] PASSED
tests/test_form_pages.py::test_form_file_field_submit[pickle] PASSED
tests/test_form_pages.py::test_form_file_field_submit[sql] PASSED
tests/test_form_pages.py::test_file_field_validation[pickle] PASSED
tests/test_form_pages.py::test_file_field_validation[sql] PASSED
tests/test_formdata.py::test_file_field[pickle] PASSED
tests/test_formdata.py::test_file_field[sql] PASSED
tests/test_formdef.py::test_file_field_migration PASSED
tests/test_formdef_import.py::test_file_field PASSED

==================================================================== 719 tests deselected by '-kfile_field' ==
==================================================================
==================================================================== 10 passed, 719 deselected in 8.75 seconds =
==================================================================

(wcs)bdauvergne@dor-lomin:~/Code/wcs$ pip freeze
beautifulsoup4==4.4.1
feedparser==5.2.1
funcsigs==0.4
mechanize==0.2.5
mock==1.3.0
pbr==1.8.1
psycopg2==2.6.1
py==1.4.30
pytest==2.6.3
pytest-random==0.2
python-dateutil==2.4.2
qrcode==5.1
scgi==1.8
six==1.10.0
vobject==0.6.6
waitress==0.8.10
-e git+git@git.entrouvert.org:wcs@cfd721045a207ffadfbb6468d99b73ce3f9e3368#egg=wcs-wip_file-validation
WebOb==1.5.1
WebTest==2.0.18
```

#46 - 10 novembre 2015 16:31 - Benjamin Dauvergne

- Fichier 0003-add-support-for-file-validation-8402.patch ajouté

Test test_backoffice_file_field_validation modifié pour passer en SQL.

#47 - 10 novembre 2015 17:11 - Frédéric Péters

Ok, go.

#48 - 11 novembre 2015 17:32 - Frédéric Péters

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

Ça a donc été poussé puis j'ai fait une correction au test pour aussi fonctionner quand python-magic est présent. Et j'ai poussé le patch côté auquo.

#49 - 03 décembre 2015 15:52 - Frédéric Péters

- Version cible mis à v1.22

#50 - 03 décembre 2015 15:56 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-add-support-for-file-validation.patch	10,6 ko	28 septembre 2015	Benjamin Dauvergne
0001-formdef-allow-field-classes-to-have-migrate-methods-.patch	1,03 ko	02 octobre 2015	Benjamin Dauvergne
0002-add-support-for-file-validation-8402.patch	13,8 ko	02 octobre 2015	Benjamin Dauvergne
0001-formdef-allow-field-classes-to-have-migrate-methods-.patch	1,03 ko	04 octobre 2015	Benjamin Dauvergne
0002-fields-in-export_to_xml-handle-non-string-value-for-.patch	1,5 ko	04 octobre 2015	Benjamin Dauvergne
0003-add-support-for-file-validation-8402.patch	19,1 ko	04 octobre 2015	Benjamin Dauvergne
0004-formdef-rewrite-update_mimetypes-as-update_filetype-.patch	4,83 ko	04 octobre 2015	Benjamin Dauvergne
0003-add-support-for-file-validation-8402.patch	19,6 ko	05 octobre 2015	Benjamin Dauvergne
0001-formdef-allow-field-classes-to-have-migrate-methods-.patch	1,03 ko	29 octobre 2015	Benjamin Dauvergne
0002-fields-handle-non-string-value-in-export_to_xml-8402.patch	1,42 ko	29 octobre 2015	Benjamin Dauvergne
0004-formdef-rewrite-update_mimetypes-as-update_filetype-.patch	4,83 ko	29 octobre 2015	Benjamin Dauvergne
0003-add-support-for-file-validation-8402.patch	20,5 ko	29 octobre 2015	Benjamin Dauvergne
0001-root-add-reload_top-to-_q_exports-to-mimic-w.c.s.-84.patch	1,01 ko	29 octobre 2015	Benjamin Dauvergne
0002-fields-handle-non-string-value-in-export_to_xml-8402.patch	1,42 ko	06 novembre 2015	Benjamin Dauvergne
0003-add-support-for-file-validation-8402.patch	26,7 ko	10 novembre 2015	Benjamin Dauvergne
0003-add-support-for-file-validation-8402.patch	27,6 ko	10 novembre 2015	Benjamin Dauvergne
0003-add-support-for-file-validation-8402.patch	27,7 ko	10 novembre 2015	Benjamin Dauvergne