

w.c.s. - Development #10444

Gérer les justificatifs (champ fichier) et leurs métadonnées comme un tout

25 mars 2016 15:58 - Benjamin Dauvergne

Statut:	Fermé	Début:	04 avril 2016
Priorité:	Normal	Echéance:	04 avril 2016
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:	v1.42	Planning:	
Patch proposed:	Oui		
Description			
<p>Un champ fichier est un justificatif si le document_type a une clé metadata non vide.</p> <p>Pour aller avec les évolutions coté fargo il faut que le champ fichier sache présenter lui même les champs de métadonnée associés à un justificatif ainsi que la liste des justificatifs déjà validés (en faisant un appel GET au web-service de gestion des validations sur fargo en lui passant l'uuid de l'utilisateur en cours); et s'il n'y en a qu'un le pré-sélectionner.</p> <p>Sur upload d'un fichier soit directement soit en le prenant dans fargo, les métadonnées sont remises à zéro.</p> <p>Si le fichier uploadé correspond à un justificatif existant (appel GET sur le web-service de gestion des validation avec un filtre sur content_hash), les champs sont pré-remplis.</p> <p>Un champ pré-rempli n'est pas modifiable.</p> <p>Les métadonnées seront conservées directement sur l'objet PicklableUpload dans un attribut metadata. Les métadonnées sont considérées comme validées si elles contiennent les clés created, creator et que l'heure courante est dans l'intervalle [start, end] donnée par les métadonnées.</p> <p>Des métadonnées non validées font qu'un bouton « Valider » s'affiche en backoffice, sur appui de celui-ci un POST est fait sur le web-service de création de validation dans fargo, les nouvelles métadonnées retournées (avec les champs created, creator, etc.) remplace les métadonnées du formulaire qui est mis à jour. L'intervalle de validité est déduit automatiquement par fargo, donc un simple clic suffit, il n'y aura plus de boîte de dialogue.</p> <p>La possibilité de choisir un justificatif en back-office est exclu de ce ticket (c'est un peu compliqué pour l'instant car cela demande au widget FileWithPreviewWidget de connaître l'utilisateur pour lequel la "saisie pour le compte de" est en cours).</p>			
Demandes liées:			
Suit w.c.s. - Bug #10515: ne pas afficher le lien de sélection depuis le port...		Fermé	01 avril 2016

Révisions associées

Révision a058a015 - 18 avril 2016 10:57 - Benjamin Dauvergne

add support for Basic HTTP authentication to _http_request (#10444)

Révision b70212a9 - 18 avril 2016 10:57 - Benjamin Dauvergne

rewrite file validation (#10444)

- metadata fields are now directly added to the form, as part of the FileWithPreviewWidget sub-widgets (it's a CompositeWidget)
- existing validated metadatas are proposed for prefilling
- if prefilled no file is uploadée, and a special upload class NoUpload is used instead of a PicklableUpload, it only contains the metadatas.
- prefilled fields are disabled,
- on upload of a new file all prefilled fields are emptied and re-enabled.
- modify XML import/export to encode metadata sub-field as JSON in exports
- add test on import/export of FileField with metadata
- rewritten tests around file validation

Révision 13be7a0e - 18 avril 2016 10:57 - Benjamin Dauvergne

move API signing functions in their own module (#10444)

Having them in the api module leads to circular imports.

Also `get_secret()` has been rewritten as `get_secret_and_orig()`.

Révision 8fa39285 - 18 avril 2016 10:57 - Benjamin Dauvergne

tests: generalize site-options.cfg fixtures (#10444)

Révision 5dc0400f - 18 avril 2016 10:57 - Benjamin Dauvergne

implement URL signatures in the file validation web-service calls (#10444)

Historique

#1 - 25 mars 2016 16:01 - Benjamin Dauvergne

- Description mis à jour

#2 - 28 mars 2016 03:38 - Benjamin Dauvergne

Autres règles observées en cours de route: si le champ fichier n'est pas requis, mais qu'un fichier est uploadé et un champ de métadonnée requis par le type de justificatif est vide, une erreur de validation est levée et le champ devient requis, idem si un champ de métadonnées est rempli mais qu'aucun fichier n'a été uploadé, celui-ci devient requis; l'idée est qu'on doit garantir qu'une validation est possible.

#3 - 28 mars 2016 03:43 - Benjamin Dauvergne

- Fichier 0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch ajouté

- Fichier 0002-rewrite-file-validation-fixes-10444.patch ajouté

Le premier patch ajoute le support HTTP Basic authentication dans les URLs pour `_http_request`, tout simplement parce que pour mes tests j'ai mis directement `username/mdp` d'un utilisateur administrateur sur `fargo` dans l'URL `fargo_url`.

Le deuxième patch implémente le ticket. Je vais poster une vidéo plus tard.

#4 - 29 mars 2016 12:59 - Benjamin Dauvergne

- Fichier 0002-tests-set-site-charset-to-utf-8.patch ajouté

- Fichier 0003-rewrite-file-validation-fixes-10444.patch ajouté

Voilà avec des tests qui passent et une modification à `tests/utilities.py` pour que le publisher des tests soit bien en utf-8. J'attaque la vidéo.

#5 - 29 mars 2016 13:51 - Frédéric Péters

Dans les tests, je ne retirerais pas les tests actuels, (avec des `document_type` sans metadata), qui doivent continuer à fonctionner.

```
r += htmltext('<p>%s&nbsp;: %s</p>') % (meta_field['label'], metadata_value)
```

La place de l'espace insécable est dans le `.po`; aussi je mettrais ces entrées sous forme de ``, dans un ``, avec un attribut `class`.

```
PARTIAL_FILL_ERROR = _('This field is normally not required, but you must leave it completely '
                        'empty or fill it, You cannot fill it partially')
```

Ça doit être déclaré avec `N_()`, et `_()` appelé quand la chaîne est utilisée.

Aussi, sur le fond, le message est-il correct ? i.e. on doit pouvoir avoir un justific de domicile avec un numéro de boîte optionnel et ne pas dire à l'utilisateur qu'il doit compléter tous les champs.

#6 - 30 mars 2016 01:31 - Benjamin Dauvergne

Frédéric Péters a écrit :

Dans les tests, je ne retirerais pas les tests actuels, (avec des `document_type` sans metadata), qui doivent continuer à fonctionner.

Je ne comprends pas, je n'ai retiré aucun test, j'ai juste modifié tous ceux qui concernaient des types de document `fargo` (et donc avec métadonnées).

[...]

La place de l'espace insécable est dans le `.po`; aussi je mettrais ces entrées sous forme de ``, dans un ``, avec un attribut `class`.

Ok.

[...]

Ça doit être déclaré avec N_(), et _() appelé quand la chaîne est utilisée.

Ok.

Aussi, sur le fond, le message est-il correct ? i.e. on doit pouvoir avoir un justificatif de domicile avec un numéro de boîte optionnel et ne pas dire à l'utilisateur qu'il doit compléter tous les champs.

L'erreur n'est levée que si un champ requis par le type de document n'est pas rempli (bien que le champ fichier dans son ensemble ne soit pas lui requis); donc sur le fond le message est imprécis, on peut laisser vide certains champs, mais c'est difficile à exprimer. Je veux bien une proposition si tu vois mieux. Idéalement ça devrait plutôt être géré côté client sans message pour expliquer ce qui est ou non requis et pourquoi.

#7 - 30 mars 2016 08:01 - Frédéric Péters

Benjamin Dauvergne a écrit :

Frédéric Péters a écrit :

Dans les tests, je ne retirerais pas les tests actuels, (avec des document_type sans metadata), qui doivent continuer à fonctionner.

Je ne comprends pas, je n'ai retiré aucun test, j'ai juste modifié tous ceux qui concernaient des types de document fargo (et donc avec métadonnées).

En cela ça retire les tests sur des documents types qui n'ont pas de métadonnées.

Aussi, sur le fond, le message est-il correct ? i.e. on doit pouvoir avoir un justificatif de domicile avec un numéro de boîte optionnel et ne pas dire à l'utilisateur qu'il doit compléter tous les champs.

L'erreur n'est levée que si un champ requis par le type de document n'est pas rempli (bien que le champ fichier dans son ensemble ne soit pas lui requis); donc sur le fond le message est imprécis, on peut laisser vide certains champs, mais c'est difficile à exprimer. Je veux bien une proposition si tu vois mieux. Idéalement ça devrait plutôt être géré côté client sans message pour expliquer ce qui est ou non requis et pourquoi.

Donc le message est faux, va bloquer les usagers qui vont penser qu'ils doivent mettre quelque chose dans le champ "numéro de boîte" alors que ce n'est pas vrai. Pourquoi ne pas en rester au message générique actuel (ERROR_NOTICE) et laisser l'utilisateur regarder le champ fichier pour voir le détail du problème ?

#8 - 30 mars 2016 10:09 - Benjamin Dauvergne

Le message n'est pas faux, il est imprécis. Il faut remplir les champs si on attache un fichier (ou attacher un fichier si on remplit un champ, même pas requis) mais pas forcément tous. Pour les tests je vois le problème, mais j'en ai identifié un plus embêtant qui est l'export des champs fichiers, ça ne marche plus avec les types de documents issus de fargo (je crois que ça ne marchait déjà plus avant car il y a file_type dans extra_attributes alors que le champ s'appelle document_type maintenant). Mais là s'ajoute le problème du contenu du champ metadata. Je me demande s'il vaut mieux n'exporter que l'identifiant du type de document et espérer qu'à l'import on aura un fargo configuré dans w.c.s. connaissant ce type de document pour en récupérer le schéma ou continuer à exporter le schéma tel quel avec possibilité qu'à l'exécution ça ne marche pas (pas de fargo ou un fargo qui n'a pas ce type là).

#9 - 30 mars 2016 11:02 - Benjamin Dauvergne

- Fichier 0002-rewrite-file-validation-fixes-10444.patch ajouté

J'ai ajouté les tests suivants, je vais ajouter des tests sur les imports/exports.

```
diff --git a/tests/test_backoffice_pages.py b/tests/test_backoffice_pages.py
index 1dfbc0d..4c00d57 100644
--- a/tests/test_backoffice_pages.py
+++ b/tests/test_backoffice_pages.py
@@ -1578,6 +1578,51 @@ def test_menu_json(pub):
     assert resp.body == 'foo(%s);' % menu_json_str
     assert resp.headers['content-type'] == 'application/javascript'

+def test_backoffice_file_field_fargo_no_metadata(pub, fargo_url):
+    document_type = {
+        'id': 'justificatif-de-domicile',
+        'fargo': True,
+        'mimetypes': ['application/pdf'],
+        'label': 'Justificatif de domicile',
+    }
+    user = create_user(pub, is_admin=True)
+    user.name_identifiers = ['12345']
+    user.store()
```

```

+ FormDef.wipe()
+ formdef = FormDef()
+ formdef.name = 'form title'
+ formdef.fields = [fields.FileField(
+     id='0', label='1st field', type='file',
+     document_type=document_type)]
+ formdef.store()
+ formdef.data_class().wipe()
+ upload = Upload('test.pdf', '%PDF-1.4', 'application/pdf')
+ digest = hashlib.sha256('%PDF-1.4').hexdigest()
+ app = login(get_app(pub))
+ with mock.patch('wcs.file_validation.fargo_get') as fargo_get:
+     resp = app.get('/form-title/')
+     assert fargo_get.call_count == 0
+     resp.forms[0]['f0$file'] = upload
+ with mock.patch('wcs.file_validation.fargo_get') as fargo_get:
+     resp = resp.forms[0].submit('submit')
+     assert fargo_get.call_count == 0
+ assert 'Check values then click submit.' in resp.body
+ resp = resp.forms[0].submit('submit').follow()
+ assert formdef.data_class().count() == 1
+ formdata = formdef.data_class().select()[0]
+ form_id = formdata.id
+ assert not hasattr(formdata.data['0'], 'metadata')
+ assert not '0_structured' in formdata.data
+ resp = app.get('/backoffice/management/form-title/%s/' % form_id)
+ assert not 'Validate' in resp.body
+ with mock.patch('wcs.file_validation.http_post_request') as http_post_request:
+     resp = app.get('/backoffice/management/form-title/%s/validate?field_id=0' % form_id)
+     assert http_post_request.call_count == 0
+ resp = resp.follow()
+ assert not 'Valid ' in resp.body
+ assert not 'Validate' in resp.body
+
+
def test_backoffice_file_field_validation(pub, fargo_url):
    document_type = {
        'id': 'justificatif-de-domicile',
@@ -1642,6 +1687,7 @@ def test_backoffice_file_field_validation(pub, fargo_url):
    assert formdata.data['0'].metadata == metadata
    assert formdata.data['0_structured'] == metadata
    resp = app.get('/backoffice/management/form-title/%s/' % form_id)
+ assert 'Validate' in resp.body
    for metadata_field in document_type['metadata']:

        fragment = '%s ; %s' % (metadata_field['label'], metadata[metadata_field['name']])
@@ -1757,7 +1803,7 @@ def test_backoffice_file_validation_no_upload(pub, fargo_url):
    fragment = '%s ; %s' % (metadata_field['label'], metadata[metadata_field['name']])
    assert fragment in resp.body
    resp = app.get('/backoffice/management/form-title/%s/' % form_id)
- assert 'Code postal ; 75014' in resp.body
+ assert not 'Validate' in resp.body
    for metadata_field in document_type['metadata']:

        fragment = '%s ; %s' % (metadata_field['label'], metadata[metadata_field['name']])
diff --git a/tests/test_form_pages.py b/tests/test_form_pages.py
index cd15b86..20639a4 100644
--- a/tests/test_form_pages.py
+++ b/tests/test_form_pages.py
@@ -2348,6 +2348,44 @@ def test_file_field_validation(pub, fargo_url):
    assert formdata.data['0_structured'] == metadata

+def test_file_field_fargo_no_metadata(pub, fargo_url):
+    document_type = {
+        'id': 'justificatif-de-domicile',
+        'fargo': True,
+        'label': 'Justificatif de domicile',
+    }
+    user = create_user(pub)
+    user.name_identifiers = ['12345']
+    user.store()
+    FormDef.wipe()
+    formdef = FormDef()
+    formdef.name = 'form title'
+    formdef.fields = [fields.FileField(

```

```

+         id='0', label='1st field', type='file',
+         document_type=document_type)
+     ]
+     formdef.store()
+     formdef.data_class().wipe()
+     upload = Upload('test.pdf', '%PDF-1.4', 'application/pdf')
+     app = login(get_app(pub), username='foo', password='foo')
+     with mock.patch('wcs.file_validation.fargo_get') as fargo_get:
+         resp = app.get('/form-title/')
+         assert fargo_get.call_count == 0
+         resp.forms[0]['f0$file'] = upload
+     with mock.patch('wcs.file_validation.fargo_get') as fargo_get:
+         resp = resp.forms[0].submit('submit')
+         assert fargo_get.call_count == 0
+     assert 'Check values then click submit.' in resp.body
+     resp = resp.forms[0].submit('submit')
+     assert resp.status_int == 302
+     resp = resp.follow()
+     assert 'The form has been recorded' in resp.body
+     assert formdef.data_class().count() == 1
+     formdata = formdef.data_class().select()[0]
+     assert not hasattr(formdata.data['0'], 'metadata')
+     assert not '0_structured' in formdata.data
+
+
+ def test_form_string_field_autocomplete(pub):
+     formdef = create_formdef()
+     formdef.fields = [fields.StringField(id='0', label='string', type='string', required=False)]
diff --git a/tests/test_formdata.py b/tests/test_formdata.py
index c09ee2c..74614bd 100644
--- a/tests/test_formdata.py
+++ b/tests/test_formdata.py
@@ -152,6 +152,29 @@ def test_file_field(pub):
     assert isinstance(substvars.get('form_var_foo_raw'), Upload)

+def test_file_field_fargo_no_metadata(pub):
+    document_types = {
+        'justificatif-de-domicile': {
+            'id': 'justificatif-de-domicile',
+            'fargo': True,
+            'label': 'Justificatif de domicile',
+        }
+    }
+    formdef.data_class().wipe()
+    formdef.fields = [fields.FileField(id='0', label='file', varname='foo',
+                                     document_type=document_types['justificatif-de-domicile'])]
+    formdef.store()
+    formdata = formdef.data_class()()
+    upload = Upload('test.txt', 'text/plain', 'ascii')
+    upload.receive(['first line', 'second line'])
+    formdata.data = {'0': upload}
+    formdata.id = 1
+    substvars = formdata.get_substitution_variables()
+    assert substvars.get('form_var_foo') == 'test.txt'
+    assert substvars.get('form_var_foo_url').endswith('/foobar/1/download?f=0')
+    assert isinstance(substvars.get('form_var_foo_raw'), Upload)
+
+
+ def test_file_field_with_metadata(pub):
+     document_types = {
+         'justificatif-de-domicile': {

```

#10 - 30 mars 2016 18:41 - Benjamin Dauvergne

- Fichier 0002-rewrite-file-validation-fixes-10444.patch ajouté

J'ai viré PARTIAL_FILL_ERROR, à la place il y a simplement des self.REQUIRED_ERROR sur les champs nécessaires. J'ai aussi ajouté des tests d'import/export XML.

#11 - 31 mars 2016 15:20 - Benjamin Dauvergne

- Fichier 0004-implement-URL-signatures-in-the-file-validation-web-.patch ajouté

- Fichier 0003-move-API-signing-functions-in-their-own-module-10444.patch ajouté

- Fichier 0002-rewrite-file-validation-10444.patch ajouté

- Fichier 0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch ajouté

Nouvelle série complète avec l'ajout des signatures d'URL vers fargo, j'ai du bouger une partie des fonctions dans wcs.api vers wcs.api_utils pour éviter les imports circulaires qui se multipliait (du à des appels à wcs.file_validation dans wcs.fields et wcs.qommon.form).

#12 - 31 mars 2016 16:37 - Benjamin Dauvergne

Patch 0004 avec des tests qui passent (certainement que je vais le merger avec le 0002 et mettre le 0003 en premier).

#13 - 31 mars 2016 16:37 - Benjamin Dauvergne

- Fichier 0004-implement-URL-signatures-in-the-file-validation-web-.patch ajouté

#14 - 31 mars 2016 18:59 - Benjamin Dauvergne

- Fichier 0003-move-API-signing-functions-in-their-own-module-10444.patch ajouté

- Patch proposed changé de Non à Oui

Correction à get_secret_and_orig().

#15 - 05 avril 2016 12:38 - Benjamin Dauvergne

- Fichier validation.ogv ajouté

La vidéo.

#16 - 05 avril 2016 15:43 - Brice Mallet

Correspond bien à ce que l'on a discuté, OK pour moi, merci.

Un point, pè que je ne comprends pas trop la cinématique décrite dans la vidéo : à 0:46 on voit que c'est validé or à ma compréhension, l'agent n'a pas encore validé l'adresse à cette étape, non ?

#17 - 05 avril 2016 15:46 - Frédéric Péters

Il y a une validation à 0:42.

#18 - 05 avril 2016 17:47 - Benjamin Dauvergne

Si on utilise un justificatif déjà validé de la liste déroulante (dont il faut améliorer le placeholder je pense) alors on n'a pas besoin de valider, par contre il n'y a aucun fichier qui y sera associé. De fait le justificatif est associé au hash d'un fichier, donc ce serait possible sur présentation d'un fichier déjà validé de retrouver la validation qui va avec, mais je ne l'ai pas fait trouvant le cas d'usage complètement limite. Je me dis qu'il faut forcer de vrai dématérialisation, la donnée informatique est plus importante que le document. Ce qui reste à faire pour la phase 2 c'est de pouvoir dans le remplissage d'un formulaire "pour le compte de", remplir les champs sans télécharger de fichier, en demandant une validation immédiate et aussi voir les validations associées au compte du formulaire et pas celui de l'agent actuellement connecté.

Il me semble que pour le justificatif de domicile (et pour d'autres certainement) il faudrait que j'ajoute un champ "Type du document" pour y marquer, facture EDF, facture d'eau, facture France Telecom, etc.

Parmi les limitations actuelles:

- ça ne gère que des champs chaînes de caractères, on pourrait éventuellement supporter des listes à choix ou d'autres simples de ce type, je préfère attendre le besoin
- il n'est pas possible d'ajouter du paramétrage à ces champs autre que leur nom: pas de class halfwidth, fullwidth, etc.. pas de texte d'aide, on pourrait assez facilement accepter un "extra_wcs_kwargs" dans les schémas fargo; mais je laisse ça pour plus tard
- le statut validé ou pas du document ne fait pas partie des variables exportées, donc il n'est pas possible de faire une condition de saut n'autorisant le saut que si c'est validé par exemple

#19 - 06 avril 2016 09:05 - Frédéric Péters

Benjamin Dauvergne a écrit :

Parmi les limitations actuelles :

- ça ne gère que des champs chaînes de caractères, on pourrait éventuellement supporter des listes à choix ou d'autres simples de ce type, je préfère attendre le besoin
- il n'est pas possible d'ajouter du paramétrage à ces champs autre que leur nom: pas de class halfwidth, fullwidth, etc.. pas de texte d'aide, on pourrait assez facilement accepter un "extra_wcs_kwargs" dans les schémas fargo; mais je laisse ça pour plus tard

Contre ces points, dès maintenant, je suggérerais que le contenu de "metadata" (dans le json renvoyé par fargo) corresponde au json définissant les champs dans w.c.s.; de {'label': 'Nom', 'name': 'nom'} à {'label': 'Nom', 'varname': 'nom', 'type': 'string'}, ou que ça soit transformé ainsi (name→varname, ajout de type:string quand absence de type) automatiquement par w.c.s. Pour que w.c.s. derrière les affiche non pas en ajoutant

des StringWidget mais en passant par les classes fields.

Dès maintenant parce que les champs adresse ils vont devoir faire appel au SIG (type de voie, nom de la voie, numéro, complément).

#20 - 06 avril 2016 09:46 - Benjamin Dauvergne

Ok.

#21 - 06 avril 2016 14:11 - Benjamin Dauvergne

En fait je ne vois pas comment utiliser les classes Field dans un CompositeWidget...

#22 - 06 avril 2016 14:34 - Frédéric Péters

Je dirais qu'en appelant `field.add_to_form(self)` (self étant le compositewidget), ça pourrait peut-être passer; sinon factoriser un peu le `add_to_form` pour également avoir un `add_to_composite_widget()`.

#23 - 06 avril 2016 18:38 - Benjamin Dauvergne

- Fichier `0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch` ajouté
- Fichier `0002-rewrite-file-validation-10444.patch` ajouté
- Fichier `0003-move-API-signing-functions-in-their-own-module-10444.patch` ajouté
- Fichier `0004-implement-URL-signatures-in-the-file-validation-web-.patch` ajouté

Ça complexifie un peu (j'ai du mettre des `enumerate(metadata)` partout pour pouvoir produire les `f1`, `f2`, `f3`, etc.).

#24 - 07 avril 2016 15:44 - Frédéric Péters

- Suit Bug `#10515`: ne pas afficher le lien de sélection depuis le porte-doc aux utilisateurs non identifiés ajouté

#25 - 07 avril 2016 16:43 - Frédéric Péters

Après une première série de tests :

Au premier clic sur suivant, sur un formulaire pas complété, les différentes métadonnées sont marquées comme étant obligatoires, mais pas le champ fichier en tant que tel. Je les remplis, je valide, et voilà il est marqué obligatoire.

Le nom du fichier apparait après les métadonnées, c'est étrange.

Il manque l'ajout de 'fargo' au `_q_exports` de `AlternateRootDirectory` dans le module auquo.

La modif avec le `enumerate(metadata)`, il me semble que ça casse la sélection d'un document existant. (le `data-validations` du `<select>` contient des références au `varname`, alors que les champs sont nommés façon `f1$f0`)

Après validation, dans le backoffice, "validated by Frédéric Péters on 2016-04-07T14:27:52.768120Z", ce serait bien d'avoir le timestamp en format/heure locale. (même si c'était déjà comme ça avant).

#26 - 12 avril 2016 11:06 - Benjamin Dauvergne

- Fichier `0005-implement-URL-signatures-in-the-file-validation-web-.patch` ajouté
- Fichier `0004-tests-generalize-site-options.cfg-fixtures-10444.patch` ajouté
- Fichier `0003-move-API-signing-functions-in-their-own-module-10444.patch` ajouté
- Fichier `0002-rewrite-file-validation-10444.patch` ajouté
- Fichier `0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch` ajouté

Nouvelle série tous les points vus plus haut sont corrigés:

- champs fichier marqué obligatoire
- le nom du fichier est maintenant placé avant les champs de métadonnées
- fargo conservé (patch du `#10515` mis à jour)
- nom des attributs `data-` sur les champs de sélection corrigées
- les timestamps sont formatés avec les chaînes locales:

```
...
        created = misc.localstrftime(
            time.localtime(
                misc.parse_isotime(value.metadata['created']))
        )
...
    r += htmltext('<p>%s</p>') % (_('Valid from %(start)s to %(end)s') % {
```

```
'start': strftime(
    misc.date_format(),
    misc.get_as_datetime(value.metadata['start'])),
'end': strftime(
    misc.date_format(),
    misc.get_as_datetime(value.metadata['end'])),
```

#27 - 12 avril 2016 11:07 - Benjamin Dauvergne

Aussi j'ai déplacé un bout du [#10506](#) ce qui est devenu ce patch:

```
commit 30edefdeebccb4e94143bc0899baef2d51af8490
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Fri Apr 8 12:56:49 2016 +0200
```

```
tests: generalize site-options.cfg fixtures (#10444)
```

#28 - 12 avril 2016 15:20 - Frédéric Péters

En testant (la précédente série, mais cette part du code ne me semble pas avoir été modifiée), j'ai eu des crashes sur `self.document_type == None`.

Pour avancer en local j'ai rapidement fait ces modifs :

```
diff --git a/wcs/fields.py b/wcs/fields.py
index e7bf08a..347f15d 100644
--- a/wcs/fields.py
+++ b/wcs/fields.py
@@ -725,7 +725,7 @@ class FileField(WidgetField):
     @property
     def metadata(self):
-        return self.document_type.get('metadata', [])
+        return (self.document_type or {}).get('metadata', [])
```

```
diff --git a/wcs/forms/common.py b/wcs/forms/common.py
index 9a46cbf..87cea9d 100644
--- a/wcs/forms/common.py
+++ b/wcs/forms/common.py
@@ -634,8 +634,8 @@ class FormStatusPage(Directory):
     def display_file_field(self, form_url, field, value):
         r = TemplateIO(html=True)
         validated = None
-        is_fargo_dt = field.document_type.get('fargo', False)
-        has_metadata = bool(field.document_type.get('metadata', []))
+        is_fargo_dt = (field.document_type or {}).get('fargo', False)
+        has_metadata = bool((field.document_type or {}).get('metadata', []))
         if file_validation.has_file_validation() and is_fargo_dt and has_metadata:
             validated = file_validation.is_valid(self.filled, field, value)
             if validated is False:
```

#29 - 12 avril 2016 16:01 - Benjamin Dauvergne

Le problème me semble venir de formulaire datant d'avant l'ajout du champ `document_type`, alors ce patch là pourrait aider:

```
diff --git a/wcs/fields.py b/wcs/fields.py
index c83a813..a1189ed 100644
--- a/wcs/fields.py
+++ b/wcs/fields.py
@@ -890,6 +890,10 @@ class FileField(WidgetField):
     value = data.get(field_id)
     return getattr(value, 'metadata', {})

+    def __setstate__(self, state):
+        self.__dict__ = state
+        self.document_type = self.document_type or {}
+
register_field_class(FileField)
```

#30 - 12 avril 2016 16:18 - Benjamin Dauvergne

- Fichier 0001-initialize-document_type-in-FileField.__setstate__.patch ajouté

En virant les autres self.document_type or {} pour faire propre (je le rebaserai dans le 0002 ensuite).

#31 - 12 avril 2016 16:18 - Frédéric Péters

Oui, ce patch-là fonctionne.

La ligne qui suit peut alors également être simplifiée :

```
@property
def file_type(self):
    return (self.document_type or {}).get('mimetypes', [])
```

#32 - 12 avril 2016 16:21 - Benjamin Dauvergne

- Fichier 0002-rewrite-file-validation-10444.patch ajouté

Hop rebasé dans le 0002.

#33 - 13 avril 2016 10:06 - Benjamin Dauvergne

J'ai testé en important un formulaire de bistro (candidature spontané qui contient un champ fichier), aucun souci.

#34 - 13 avril 2016 10:35 - Benjamin Dauvergne

Et j'ai mis en phase le code coté fargo il restait deux bidouilles, renvoyer des dates iso8601 sans microsecondes (le code dans w.c.s. n'aime pas, faudra passer à isodate là aussi) et gérer les cas où les données du justificatif ne sont plus en phases avec les champs et le template de résumé (KeyError dans ce cas, qui est ignoré en faveur d'un template générique qui concatène tout).

#35 - 18 avril 2016 10:11 - Frédéric Péters

ok.

#36 - 18 avril 2016 11:01 - Benjamin Dauvergne

- Statut changé de Nouveau à Résolu (à déployer)

#37 - 01 mai 2016 20:57 - Frédéric Péters

- Version cible mis à v1.42

#38 - 28 juin 2016 19:51 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch	1,07 ko	28 mars 2016	Benjamin Dauvergne
0002-rewrite-file-validation-fixes-10444.patch	21,1 ko	28 mars 2016	Benjamin Dauvergne
0002-tests-set-site-charset-to-utf-8.patch	1,29 ko	29 mars 2016	Benjamin Dauvergne
0003-rewrite-file-validation-fixes-10444.patch	39,6 ko	29 mars 2016	Benjamin Dauvergne
0002-rewrite-file-validation-fixes-10444.patch	46,4 ko	30 mars 2016	Benjamin Dauvergne
0002-rewrite-file-validation-fixes-10444.patch	49,1 ko	30 mars 2016	Benjamin Dauvergne
0004-implement-URL-signatures-in-the-file-validation-web-.patch	3,3 ko	31 mars 2016	Benjamin Dauvergne
0003-move-API-signing-functions-in-their-own-module-10444.patch	12 ko	31 mars 2016	Benjamin Dauvergne
0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch	1,08 ko	31 mars 2016	Benjamin Dauvergne
0002-rewrite-file-validation-10444.patch	49,1 ko	31 mars 2016	Benjamin Dauvergne
0004-implement-URL-signatures-in-the-file-validation-web-.patch	7,54 ko	31 mars 2016	Benjamin Dauvergne
0003-move-API-signing-functions-in-their-own-module-10444.patch	12,1 ko	31 mars 2016	Benjamin Dauvergne
validation.ogv	5,31 Mo	05 avril 2016	Benjamin Dauvergne
0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch	1,08 ko	06 avril 2016	Benjamin Dauvergne
0002-rewrite-file-validation-10444.patch	51,7 ko	06 avril 2016	Benjamin Dauvergne
0003-move-API-signing-functions-in-their-own-module-10444.patch	12,1 ko	06 avril 2016	Benjamin Dauvergne
0004-implement-URL-signatures-in-the-file-validation-web-.patch	8,01 ko	06 avril 2016	Benjamin Dauvergne
0004-tests-generalize-site-options.cfg-fixtures-10444.patch	2,54 ko	12 avril 2016	Benjamin Dauvergne

0005-implement-URL-signatures-in-the-file-validation-web-.patch	8,27 ko	12 avril 2016	Benjamin Dauvergne
0003-move-API-signing-functions-in-their-own-module-10444.patch	12,1 ko	12 avril 2016	Benjamin Dauvergne
0001-add-support-for-Basic-HTTP-authentication-to-_http_r.patch	1,08 ko	12 avril 2016	Benjamin Dauvergne
0002-rewrite-file-validation-10444.patch	53,3 ko	12 avril 2016	Benjamin Dauvergne
0001-initialize-document_type-in-FileField.__setstate__.patch	1,8 ko	12 avril 2016	Benjamin Dauvergne
0002-rewrite-file-validation-10444.patch	54,4 ko	12 avril 2016	Benjamin Dauvergne