

Passerelle - Development #24364

Génération de document de PDF à partir de PDF remplissables (PDF Form)

07 juin 2018 17:58 - Benjamin Dauvergne

Statut:	Nouveau	Début:	07 juin 2018
Priorité:	Bas	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Pour une utilisation rapide de CERFA remplissables, quand ça marche.			
Demandes liées:			
Lié à Passerelle - Development #73544: connecteur pdf : avoir un endpoint "fi...		Fermé	18 janvier 2023

Historique

#1 - 07 juin 2018 18:09 - Benjamin Dauvergne

- Fichier 0001-workflows-add-pdf-filing-in-export-to-model-action-2.patch ajouté
- Patch proposed changé de Non à Oui

Manque les tests mais c'est juste parce qu'on en a parlé à Toulouse aujourd'hui, et que je n'avais rien à dire, relecture ne presse pas.

L'idée c'est de prendre un CERFA comme celui-là: <https://www.impots.gouv.fr/portail/formulaire/2494-sd/declaration-de-la-retenue-la-source>

On l'upload comme un ODT, le formulaire de l'action se modifie pour afficher pour chaque champ du PDF un champ expression calculé (la présentation n'est pas super ergonomique, pourrait être plus sympa sur plusieurs colonnes).

De plus un lien "PDF Form exemple" renvoie le même PDF avec chaque champ pré-rempli avec son nom de variable (souvent ce n'est pas évident de savoir quelle variable correspond à quoi).

J'en ai un peu chié pour faire tourner w.c.s. en standalone avec Django, je m'y prends certainement mal:

```
diff --git a/wcs/settings.py b/wcs/settings.py
index 0382d385..54eaaf65 100644
--- a/wcs/settings.py
+++ b/wcs/settings.py
@@ -60,7 +60,7 @@ MEDIA_URL = '/media/'
 # Don't put anything in this directory yourself; store your static files
 # in apps' "static/" subdirectories and in STATICFILES_DIRS.
 # Example: "/var/www/example.com/static/"
-STATIC_ROOT = os.path.join(PROJECT_PATH, 'static')
+STATIC_ROOT = os.path.join(PROJECT_PATH, 'app', 'collectstatic')

 # URL prefix for static files.
 # Example: "http://example.com/static/", "http://static.example.com/"
@@ -68,6 +68,7 @@ STATIC_URL = '/static/'

 # Additional locations of static files
STATICFILES_DIRS = (
+    '/home/bdauvergne/wd/eo/wcs/static/',
)

 # List of finder classes that know how to find static files in
@@ -75,7 +76,7 @@ STATICFILES_DIRS = (
STATICFILES_FINDERS = (
    'django.contrib.staticfiles.finders.FileSystemFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
-#    'django.contrib.staticfiles.finders.DefaultStorageFinder',
+    'django.contrib.staticfiles.finders.DefaultStorageFinder',
)

 # Make this unique, and don't share it with anybody.
@@ -138,6 +139,7 @@ INSTALLED_APPS = (
```

```

    'wcs.ctl',
    'wcs.common',
    'django.contrib.staticfiles',
+   'xstatic.pkg.jquery',
)

CACHES = {
-----
[main]
app_dir = /home/bdauvergne/wd/eo/wcs/app
data_dir = /home/bdauvergne/wd/eo/wcs/data
# Automatically create virtual host
auto_create_appdir = true
use_long_traces = true
#error_log = /var/lib/wcs/error.log
# Redirect on unknown virtual host
#missing_appdir_redirect = http://www.mysite.com
-----
# wcs/local_settings.py
DEBUG = True

WCS_LEGACY_CONFIG_FILE = '/home/bdauvergne/wd/eo/wcs/wcs.cfg'

```

Le problème principal a été d'obtenir que django avec runserver serve jquery (je pense que l'ajout de xstatic.pkg.jquery a suffit à corriger le souci, le reste de mes modifications c'est de l'exploration) pour que l'admin soit utilisable (surtout le menu Publik à gauche).

#2 - 07 juin 2018 18:21 - Benjamin Dauvergne

- Fichier *example-prereempli.pdf* ajouté
- Fichier *screenshot-localhost 8000-2018-06-07-18-20-30.png* ajouté

L'exemple du PDF avec les noms des champs.

Un screenshot du formulaire d'admin.

#3 - 07 juin 2018 18:21 - Benjamin Dauvergne

- *Priorité* changé de *Normal* à *Bas*

#4 - 14 octobre 2018 17:02 - Benjamin Dauvergne

- *Statut* changé de *Nouveau* à *Solution proposée*

#5 - 14 octobre 2018 17:39 - Benjamin Dauvergne

- *Statut* changé de *Solution proposée* à *Nouveau*

#6 - 21 janvier 2019 15:15 - Benjamin Dauvergne

- Fichier *out.pdf* ajouté

Quelques expériences sur une voie différente (et plus intéressante d'après moi), utiliser directement pdfwr et pdfd pour manipuler les PDFs finement : <http://git.entrouvert.org/misc-bdauvergne.git/tree/pdf-experiments>

Permet de générer un pdf comme celui que j'attache d'aide à la configuration du pré-remplissage (c'est le plus difficile dans l'histoire, arriver à indiquer aux fonctionnels où vont aller les données et sous quelle forme).

#7 - 04 mars 2019 11:14 - Benjamin Dauvergne

- *Projet* changé de *w.c.s.* à *Passerelle*

Je déplace sur passerelle, à mon avis c'est inutile de charger w.c.s. sur ces aspects.

#8 - 11 mars 2019 11:00 - Benjamin Dauvergne

Plan à long terme:

- un connecteur PDFfilling
- un objet fils "FillablePDF"
 - on choisit un PDF remplissable
 - on repère l'emplacement des champs via le code donné plus haut (coordonnées PDF)
 - on affiche le PDF via pdf.js
 - on utilise les coordonnées PDF pour placer des boutons sur le PDF le canvas pdf.js (voir API convertToPdfPoint dans pdf.js et la notion de

- viewport)
- on définit pour chaque champ l'expression de pré-remplissage par rapport aux données poussé (éviter d'exposer la complexité du PDF aux appelants)
- POINT D'ATTENTION : les tableaux demandent une attention particulière car généralement définis comme un paquet de champs sans liens entre eux qu'il faudrait pouvoir pré-remplir en bloc selon une logique (genre comme un tableur A1:D10). Il devrait être possible de sélectionner une zone de champs et de la définir comme étant un tableau (et reconstruire lignes/colonnes via les coordonnées relatifs des champs entre eux)

#9 - 21 octobre 2019 10:20 - Benjamin Dauvergne

- Assigné à Benjamin Dauvergne supprimé

#10 - 18 janvier 2023 11:15 - Thomas Noël

- Lié à Development #73544: connecteur pdf : avoir un endpoint "fill-form", remplissage de fichier PDF avec les données d'un formulaire ajouté

Fichiers

0001-workflows-add-pdf-filing-in-export-to-model-action-2.patch	16,8 ko	07 juin 2018	Benjamin Dauvergne
example-prerempli.pdf	356 ko	07 juin 2018	Benjamin Dauvergne
screenshot-localhost 8000-2018-06-07-18-20-30.png	157 ko	07 juin 2018	Benjamin Dauvergne
out.pdf	741 ko	21 janvier 2019	Benjamin Dauvergne