

Le «connecteur» PDF

Ce connecteur est en fait une interface d'accès à un logiciel de traitement des fichiers PDF, pdftk : <https://gitlab.com/pdftk-java/pdftk>

Le logiciel pdftk est installé sur la même machine que Passerelle, et on peut l'appeler par webservice pour :

- assembler des fichier PDF
- remplir un formulaire PDF avec des valeurs

Assemblage de PDF

Le webservice « /assemble » reçoit un dictionnaire de cette forme :

```
{
  "filename": "output.pdf",
  "files/0": {
    "filename": "example-1.pdf",
    "content_type": "application/pdf",
    "content": "JVBERi0xL... (base64 PDF) ..."
  },
  "files/1": {
    "filename": "example-2.pdf",
    "content_type": "application/pdf",
    "content": "//4lUERGL... (base64 PDF) ..."
  },
  "files/2": ...
}
```

Il retourne en réponse un document PDF, assemblage de files/0 + files/1 + files/2 + ...

La traduction dans le workflow d'une démarche sera de faire une action webservice :

- vers l'URL {{ passerelle_url }}pdf/<slug>/assemble
- utilisant la méthode POST (JSON)
- avec dans le corps de la requête au moins une donnée filename et un files/0 :
 - filename : un gabarit représentant le nom du fichier attendu en retour
 - files/0 : un gabarit représentant un champ fichier, par exemple {{ form_var_premier_fichier }}
 - files/1 : un gabarit représentant un champ fichier, par exemple {{ form_var_second_fichier }}
 - etc, jusqu'à
 - files/n : un gabarit représentant un champ fichier, par exemple {{ form_var_nieme_fichier }}
- dans l'onglet Réponse de l'action : la réponse attendue est de type « fichier joint »
- pour stocker le résultat, le workflow contiendra une donnée de traitement de type Fichier dans laquelle on enverra ce résultat (toujours dans l'onglet Réponse)

Remplissage de formulaire PDF (fill-form)

Le webservice « fill-form » permet de remplir les champs d'un fichier PDF de type « formulaire PDF » (acroform), par exemple tel que celui visible sur <https://www.service-public.fr/particuliers/vosdroits/R1492> (cerfa 10072-02, Déclaration d'intention d'aliéner ou demande d'acquisition d'un bien soumis à un droit de préemption / formulaire 10072*02)

Le connecteur dispose du fichier PDF formulaire au niveau de sa configuration, dans « Fichier formulaire PDF »

Un bouton « Remplissage de formulaire: éditer les correspondances de champ » est accessible sur le connecteur, qui présente tous les champs disponibles dans une vue du fichier PDF. Pour chacun, il faut écrire un gabarit ou une condition, en Django. Les gabarits pour les champs textes, et les conditions pour les cases à cocher.

Ces gabarits et conditions vont recevoir toutes les données envoyées lors de l'appel au webservice « fill-form ».

Le webservice « fill-form » doit donc recevoir un dictionnaire JSON, lequel sera envoyé aux gabarits et conditions pour le calcul de chacun des champs à remplir. Le webservice peut donc éventuellement recevoir « toutes les données du formulaires » depuis une action appel webservice (case « Envoyer les données du formulaire »)

La traduction dans le workflow d'une démarche sera de faire une action webservice :

- vers l'URL {{ passerelle_url }}pdf/<slug>/fill-form
- utilisant la méthode POST (JSON)
- en cochant, éventuellement, la case « Envoyer les données du formulaire »
- en envoyant dans le corps de la requête :
 - une donnée filename, gabarit représentant le nom du fichier attendu en retour
 - toute donnée supplémentaire qui serait utile aux gabarits et conditions des champs
- dans l'onglet Réponse de l'action : la réponse attendue est de type « fichier joint »
- et pour stocker le résultat, le workflow contiendra une donnée de traitement de type Fichier, dans laquelle on enverra ce résultat (toujours à configurer dans l'onglet Réponse)

Note : le format du dictionnaire envoyé par la case à cocher « Envoyer les données du formulaire » est documenté sur <https://doc-publik.entrouvert.com/dev/wcs/api-webservices/recuperation-des-donnees-d-un-formulaire/#mode-pull>

Ajout de filigrane (watermark)

Le webservice « /watermark » reçoit un dictionnaire de cette forme :

```
{
  "filename": "output.pdf",
  "file": {
    "filename": "example-1.pdf",
    "content_type": "application/pdf",
    "content": "JVBERi0xL... (base64 PDF) ..."
  },
  "stamp": {
    "filename": "filigrane.pdf",
    "content_type": "application/pdf",
    "content": "//4lUERGL... (base64 PDF) ..."
  }
}
```

Il retourne en réponse un document PDF, qui est le fichier "file" avec en filigrane "stamp".

La traduction dans le workflow d'une démarche sera de faire une action webservice :

- vers l'URL {{ passerelle_url }}pdf/<slug>/watermark
- utilisant la méthode POST (JSON)
- avec dans le corps de la requête au moins une donnée filename, un file et un stamp :
 - filename : un gabarit représentant le nom du fichier attendu en retour
 - file : un gabarit représentant un champ fichier, par exemple {{ form_var_fichier }}
 - stamp : un gabarit représentant un champ fichier, par exemple {{ form_var_filigrane }}
- dans l'onglet Réponse de l'action : la réponse attendue est de type « fichier joint »
- pour stocker le résultat, le workflow contiendra une donnée de traitement de type Fichier dans laquelle on enverra ce résultat (toujours dans l'onglet Réponse)