

Welco - Development #22550

interface avec Maarch

15 mars 2018 17:23 - Benjamin Dauvergne

Statut:	Fermé	Début:	15 mars 2018
Priorité:	Normal	Echéance:	20 avril 2018
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		

Description

Courrier envoyé à Maarch pour commencer la collaboration suite à la réception des premières spécifications (attachées):

Bonjour Laurent,

Je viens de parcourir votre document et j'ai déjà quelques remarques:

- il faudrait pouvoir paginer le listing retourné par WS1, cela me paraît risqué de demander éventuellement plusieurs centaines de Mo de scan en un seul appel, ou au moins avoir un paramètre `limit=n` pour ne renvoyer par exemple que les `n` premiers courriers (mais donc il faut un ordre déterministe sur le SELECT, par exemple ORDER BY id),
- le processus d'ingestion se fera en deux temps chez nous, on ne va pas pousser les demandes directement dans w.c.s. une fois qu'elles sont reçues dans notre outil de qualification (welco) il nous faudrait donc 1 statut en plus du genre 'GRCRECEIVED' pour ne pas les télécharger les courriers plusieurs fois, entre 'GRC' et 'GRCSSENT' pour pouvoir fonctionner comme il suit:
 1. j'appelle WS1 pour le statut 'GRC' (éventuellement avec le `limit=n` en bouclant sur les étapes 1 et 2, ou alors en paginant si c'est possible), j'injecte dans welco les courriers en conservant le `res_id`
 2. j'appelle WS2 pour passer le statut des courriers que je viens d'injecter en 'GRCRECEIVED' (et ne plus les retrouver à mon prochain appel à WS1) : impose que `external_id` et `external_link` ne soient pas obligatoires
 3. un agent qualifie la demande en demande GRC, attribution d'un `external_id` et d'un `external_link`, j'appelle WS2 pour pousser tout ça et passer le statut de la demande en 'GRCSSENT' (en fait cela se fera 1 demande à la fois, action humaine oblige, donc le fait de pouvoir pousser tout cela en une seule fois via WS2 ne nous sert pas vraiment, mais c'est un détail le WS2 tel qu'il est nous convient bien et pourrait servir pour des use case différents)
- vous spécifiez bien le statut final pour un courrier dont le traitement est terminé (END) et cela me va, par contre je n'ai pas le statut pour une erreur d'aiguillage, peut-on statuer sur 'GRCRETURNE' (aucun fétichisme, on pourra toujours changer tout ça mais c'est pour qu'on ait un formalisme pour discuter) ?
- les différences entre WS2 et WS3 me semblent assez arbitraires est-ce qu'on pourrait imaginer un seul web-service de mise à jour en général des demandes (et réintégrer le statut dans le dico de chaque demande dans `external_infos` plutôt que dans un champ statut à côté) et donc pouvoir modifier le statut, ajouter un message à l'historique, définir l'`external_id` ou l'`external_link` avec le même type d'appel ?
- juste pour vous embêter: PUT vs POST ? Et pourquoi pas POST partout ? (PUT c'est normalement pour envoyer une description complète d'une ressource, ici on n'a pas vraiment de ressource et donc de REST, ici on est plus dans du Web-RPC, PUT me paraît inutile) Dans le même ordre d'idée le WS1 serait normalement un GET car on ne fait que lire, en plus les arguments passeraient sans problème dans une querystring.
- encore une remarque de forme: ce n'est pas spécifié mais il semble que le body soit du `application/x-www-form-urlencoded` avec dans chaque champ du JSON, est-ce le cas ? Pourquoi ne pas prendre directement du JSON ?
- Pour récapituler j'essaie de faire un diagramme d'état d'un courrier géré par la GRC:

```
Error executing the plantuml macro (Missing partial wiki_external_filter/_macro_image with {:locale=>[:fr, :en], :formats=>[:pdf], :variants=>[], :handlers=>[:raw, :erb, :html, :builder, :ruby, :rsb]}. Searched in: *
"/usr/share/redmine/plugins/wiki_external_filter/app/views" * "/usr/share/redmine/plugins/wiki_external_filter/app/views" *
"/usr/share/redmine/plugins/redmine_tags/app/views" * "/usr/share/redmine/plugins/redmine_entrouvert/app/views" *
"/usr/share/redmine/plugins/plantuml/app/views" * "/usr/share/redmine/plugins/localizable/app/views" *
"/usr/share/redmine/app/views" )
```

- Dernière question : qu'en est-il de la gestion des courriers sortants ? Avez-vous des WS à nous proposer que ce soit pour la

constitution d'un courrier sortant à partir de données (adresse, contenu, etc... à vous de me dire ce qui serait nécessaire) ou d'un courrier complet déjà construit (.ODT ou .PDF) sachant que le choix entre ces deux possibilités n'a pas encore été fait mais qu'il serait bien pour nous de connaître les pistes possibles ?

Révisions associées

Révision b231dee2 - 12 juillet 2018 22:55 - Benjamin Dauvergne

setup.py: limit django-haystack to <2.8 until we are officialy django 1.11 compatible (#22550)

Révision d2361b5e - 12 juillet 2018 22:55 - Benjamin Dauvergne

qualif: store keep formdata backoffice URL (#22550)

Révision fc5a4900 - 12 juillet 2018 22:55 - Benjamin Dauvergne

settings: remove alfortville default validation steps (#22550)

Révision a4ae8845 - 12 juillet 2018 22:55 - Benjamin Dauvergne

mail: feed from MaarchCourrier (#22550)

Historique

#1 - 15 mars 2018 18:08 - Benjamin Dauvergne

- *Echéance mis à 10 avril 2018*
- *Assigné à mis à Benjamin Dauvergne*

#3 - 16 avril 2018 11:36 - Benjamin Dauvergne

- *Echéance changé de 10 avril 2018 à 20 avril 2018*

#4 - 17 avril 2018 16:06 - Benjamin Dauvergne

- *Fichier Maarch-STD-WS-Publik-Maarch_V1.1(1).odt ajouté*

Dernière version des spécifications.

#5 - 17 avril 2018 16:06 - Benjamin Dauvergne

- *Fichier Maarch-STD-WS-Publik-Maarch.odt supprimé*

#7 - 19 mai 2018 16:50 - Benjamin Dauvergne

- *Fichier 0003-mail-feed-from-MaarchCourrier-22550.patch ajouté*
- *Fichier 0002-qualif-store-keep-formdata-backoffice-URL-22550.patch ajouté*
- *Fichier 0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch ajouté*
- *Patch proposed changé de Non à Oui*

Première impression, manque des tests avec un mock de Maarch.

- Ça se configure dans les settings via:

```
MAARCH_FEED = {
    'ENABLE': True,
    'URL': 'http://...',
    'USERNAME': 'xxx',
    'PASSWORD': 'yyy',
}
```

- Je mets à jour les informations coté Maarch de manière synchrone (via signal post_save sur le modèle Association) lors de l'association d'un courrier avec une démarche, pas forcément une bonne idée sachant que Maarch n'a pas besoin de l'information en temps réel; mais je le fais en tâche de fond il faudra que je conserve quelque part le statut "notifié/non-notifié de l'association".
- L'identifiant du courrier dans maarch (res_id) est stocké dans "Mail.registered_mail_number" un truc initialement prévu pour Alfortville,
- J'ajoute un champ dans Association pour stocker l'URL BO du formulaire (c'est pas super utile coté Welco mais c'est pour pouvoir le passer à Maarch plus tard)

Voilà, j'attends les premières critiques, je finis les tests.

#8 - 22 mai 2018 09:16 - Benjamin Dauvergne

- Fichier 0003-mail-feed-from-MaarchCourrier-22550.patch ajouté
- Fichier 0002-qualif-store-keep-formdata-backoffice-URL-22550.patch ajouté
- Fichier 0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch ajouté
- Fichier 0004-handle-rejection-of-incoming-mails-to-be-rebased.patch ajouté

ajout de la possibilité de rejeter un courrier (mais donc uniquement au niveau welco, j'avais proposé de pouvoir le faire aussi au niveau w.c.s. à voir:

1. on ne le fait pas finalement,
2. on le fait mais en deux temps: un bouton de rejet de w.c.s. vers welco, ensuite à l'agent trieur welco de renvoyer vers maarch ou de requalifier vers une autre demande w.c.s, ça demande à revoir le fonctionnement de la vue reject pour ne plus simplement supprimer l'objet Mail)
3. on le fait mais on peut directement renvoyer vers Maarch depuis w.c.s. sans repasser par la case qualification dans welco.)

Solution 2. et 3. nécessite un web-service coté welco et certainement d'envoyer dans le submission context une cancel_url qui permettra à w.c.s. de proposer un bouton de renvoie (à déterminer si c'est à faire dans chaque workflow ou si w.c.s. propose directement un bouton).

#9 - 29 juin 2018 11:57 - Benjamin Dauvergne

- Fichier 0003-mail-feed-from-MaarchCourrier-22550.patch ajouté
- Fichier 0002-qualif-store-keep-formdata-backoffice-URL-22550.patch ajouté
- Fichier 0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch ajouté
- Fichier 0004-settings-remove-alfortville-default-validation-steps.patch ajouté
- Statut changé de Nouveau à Solution proposée

Voilà avec des tests.

Le dernier commit retire la configuration par défaut des étapes de validation qui ne correspond qu'à l'usage d'alfortville (ça pète mes tests qui dépendent du comportement normal, sans étape de validation).

J'ai tout de suite reporter VALIDATION_STEPS_LABELS avec les valeurs traduites dans les settings.json de welco/alfortville en test et prod:

```
"VALIDATION_STEP_LABELS" : {
  "done-dgs" : "Transmettre au Dgd/CabMaire",
  "done-dga" : "Transmettre au service",
  "done-qualif" : "Transmettre au DGS"
},
```

#10 - 29 juin 2018 11:59 - Benjamin Dauvergne

- Fichier 0003-settings-remove-alfortville-default-validation-steps.patch ajouté
- Fichier 0002-qualif-store-keep-formdata-backoffice-URL-22550.patch ajouté
- Fichier 0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch ajouté
- Fichier 0004-mail-feed-from-MaarchCourrier-22550.patch ajouté

Forcément j'ai oublié d'intégrer mes tests.

#11 - 29 juin 2018 12:38 - Frédéric Péters

```
return data['id'], data.get('url_backoffice')
```

Mais dans [#23939](#) c'est backoffice_url finalement.

(pas lu 0004)

#12 - 29 juin 2018 12:39 - Frédéric Péters

Un mot sur la manière dont on pourrait tester 0004 ?

#13 - 29 juin 2018 13:42 - Benjamin Dauvergne

- Fichier 0003-settings-remove-alfortville-default-validation-steps.patch ajouté
- Fichier 0002-qualif-store-keep-formdata-backoffice-URL-22550.patch ajouté
- Fichier 0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch ajouté

- Fichier 0004-mail-feed-from-MaarchCourier-22550.patch ajouté

J'ai corrigé le 'url_backoffice' dans 0002 et lest test de 0004.

Pour 0004 à part se dire qu'à partir du moment où j'ai reporté cette configuration dans settings.json ça devrait aller je ne sais pas trop, mon autre idée ça aurait été de poser le settings via AppConfig.ready() de l'application alfortville puis je me suis dit que ça ne devait pas marcher avec les SettingsLoader (.ready()) s'exécute bien en dehors des tenants mais je ne suis pas sûr que ça touche vraiment le settings de base).

Enfin si, une fois en recette on verra tout de suite sur la recette d'Alfortville si tout se passe bien, est-ce que ce serait suffisant ?

L'alternative c'est de devoir penser à mettre VALIDATION_STEPS = [] dans tous les settings.json des villes qui utiliseront welco pour du courrier dans le futur.

#14 - 29 juin 2018 13:43 - Frédéric Péters

Pour 0004 à part se dire qu'à partir du moment (...)

Le 0004 c'est le code par rapport à Maarch; passer par settings.json etc. pour le contrib.alfortville ne me pose aucun problème.

#15 - 29 juin 2018 13:51 - Benjamin Dauvergne

Ah ok mais donc il y a des tests complets de toute la démarche via Maarch:

- réception des mails par la commande de management
- qualification puis envoi dans w.c.s.
- réception et stockage de backoffice_url
- notification de Maarch du changement de statut du courrier, du numéro de la demande dans w.c.s. et de l'URL BO du formulaire

Je reconnais que c'est fait via mock plutôt qu'en instanciant w.c.s, est-ce cela qui te manque ?

#16 - 29 juin 2018 13:56 - Frédéric Péters

Non, je parle d'exécuter le code, en vrai, pas de tests unitaires.

#17 - 29 juin 2018 13:58 - Benjamin Dauvergne

On a une instance Maarch de test, j'ai testé avec aussi mais en live, tu voudrais une instance Maarch de test chez nous ? Dès que c'est acké je comptais brancher la démo de dév dessus.

#18 - 29 juin 2018 14:00 - Frédéric Péters

Des indications quelconque sur la manière dont prendre le code et l'exécuter en local et avoir des données dedans etc. le paramétrage à mettre, peu importe pour moi qu'il pointe une instance chez nous ou ailleurs, ou qu'il faille installer un truc en local, etc.

#19 - 29 juin 2018 14:11 - Benjamin Dauvergne

Ah oui, il suffit d'ajouter cette configuration dans settings.json:

```
MAARCH_FEED = {
  'ENABLE': True,
  'URL': '...',
  'USERNAME': '...',
  'PASSWORD': '...',
}
```

Et c'est tout.

#24 - 29 juin 2018 15:37 - Benjamin Dauvergne

- Fichier 0001-mail-feed-from-MaarchCourier-22550.patch ajouté

Nouveau patch 0004, en intégrant la remarque de christophe (raise en trop ligne 204 de welco/views.py).

#25 - 29 juin 2018 16:14 - Benjamin Dauvergne

- Fichier 0001-mail-feed-from-MaarchCourier-22550.patch ajouté

En enlevant le script inline, name == '__main__'.

#26 - 29 juin 2018 16:15 - Benjamin Dauvergne

- Fichier 0003-mail-feed-from-MaarchCourrier-22550.patch supprimé

#27 - 29 juin 2018 16:15 - Benjamin Dauvergne

- Fichier 0003-mail-feed-from-MaarchCourrier-22550.patch supprimé

#28 - 29 juin 2018 16:15 - Benjamin Dauvergne

- Fichier 0003-mail-feed-from-MaarchCourrier-22550.patch supprimé

#29 - 29 juin 2018 16:15 - Benjamin Dauvergne

- Fichier 0004-mail-feed-from-MaarchCourrier-22550.patch supprimé

#30 - 29 juin 2018 16:15 - Benjamin Dauvergne

- Fichier 0004-mail-feed-from-MaarchCourrier-22550.patch supprimé

#31 - 29 juin 2018 16:15 - Benjamin Dauvergne

- Fichier 0001-mail-feed-from-MaarchCourrier-22550.patch supprimé

#32 - 02 juillet 2018 13:08 - Christophe Siraut

Quand la configuration (MAARCH_FEED) est absente/invalide, get_maarch() ne produit pas d'erreur:

```
>>> from welco.sources.mail.utils import get_maarch
>>> maarch = get_maarch()
>>> print(maarch)
None
```

Avec la bonne config, j'ai tenté:

```
>>> from welco.sources.mail.utils import get_maarch
>>> maarch = get_maarch()
>>> c = maarch.new_courrier_with_file(open('mon.pdf').read(), format='pdf', status='GRC', type_id='101')
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: 'WelcoMaarchCourrier' object has no attribute 'new_courrier_with_file'
```

#33 - 02 juillet 2018 15:41 - Benjamin Dauvergne

new_courrier_with_file() avec deux 'r'.

#34 - 11 juillet 2018 16:05 - Benjamin Dauvergne

Poussé sur une branche <http://git.entrouvert.org/welco.git/log/?h=wip/22550-interface-avec-Maarch>

#35 - 11 juillet 2018 17:06 - Christophe Siraut

new_courrier_with_file() avec deux 'r'.

ah oui; par la suite:

```
>>> c = maarch.new_courrier_with_file(open('mon.pdf').read(), format='pdf', status='GRC', type_id='101')
>>> maarch.post_courrier(c)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "/home/chris/src/welco/welco/sources/mail/maarch.py", line 227, in post_courrier
    response = self.post_json(self.post_courrier_url, courrier.post_serialize())
  File "/home/chris/src/welco/welco/sources/mail/maarch.py", line 85, in post_serialize
    payload['fileFormat'] = self.fileFormat
AttributeError: 'Courrier' object has no attribute 'fileFormat'
```

#36 - 11 juillet 2018 17:13 - Benjamin Dauvergne

Remplace format='PDF' par fileFormat='PDF' l'API a un peu bougé mais ça ne fait pas vraiment partie de la cible voulu (à aucun moment on est amené à poster des fichiers dans Maarch depuis Welco, ça facilite juste les tests ici).

#37 - 11 juillet 2018 17:24 - Thomas Noël

qualif: store keep formdata backoffice URL ([#22550](#))

Ack

settings: remove alfortville default validation steps (#22550)

A mon avis il faut laisser les settings, parce qu'utilisés dans views.py, et donc juste les vider, avec des ptits exemples et zou, genre:

```
# Add validation circuit on specific sources, example:
# VALIDATION_STEPS = {'mail': ['done-qualif', 'done-dgs', 'done-dga']}
# VALIDATION_STEP_LABELS = {
#     'done-qualif': _('Send to DGS'),
#     'done-dgs': _('Send to DGA'),
#     'done-dga': _('Send to service')}

VALIDATION_STEPS = {}
VALIDATION_STEP_LABELS = {}
```

mail: feed from MaarchCourrier (#22550)

Faudrait pas utiliser "Mail.registered_mail_number" car ça s'affiche comme "numéro du recommandé", et c'est peut-être une feature pertinente dans un système de gestion de courrier (pas vraiment spécifique Alfortville, et donc plutôt à laisser comme tel). Plutôt créer un "Mail.external_mail_id" ?

J'aurais bien vu welco/sources/mail/__init__.py::AppConfig.association_post_save dans welco/sources/mail/maarch.py. Et j'y ajouterais un if not maarch: return juste après le maarch = get_maarch()

Allez, aussi, des "(C) 2015" qui trainent

Voilà en relecture de base, je teste quand même "grandeur nature" ce soir ; Christophe si ça t'amuse tu peux continuer aussi, mais uniquement si ça t'amuse.

#38 - 11 juillet 2018 17:30 - Christophe Siraut

Ok avec :

```
-         payload['fileFormat'] = self.fileFormat
+         payload['fileFormat'] = self.format

>>> maarch.post_courrier(c)
<Courrier pk:111 status:GRC>
>>> print c.pk
111
>>> maarch.get_mail(110)
<Courrier pk:110 status:GRC>
>>> maarch.get_mail(112)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "/home/chris/src/welco/welco/sources/mail/utils.py", line 41, in get_mail
    return self.get_courriers(clause="res_id=%s" % mail_id)[0]
IndexError: list index out of range
>>> maarch.get_mails()
[<Courrier pk:110 status:GRC>, <Courrier pk:111 status:GRC>]
```

#39 - 11 juillet 2018 18:23 - Benjamin Dauvergne

Voilà j'ai poussé sur la branche avec vos commentaires traités dans des commits distincts.

#40 - 12 juillet 2018 15:04 - Thomas Noël

Pour le lancement en cron de feed_mail_maarch, avec un tenant_command --all-tenants, il faudrait ajouter ça je pense :

```
--- a/welco/sources/mail/management/commands/feed_mail_maarch.py
+++ b/welco/sources/mail/management/commands/feed_mail_maarch.py
@@ -38,6 +38,8 @@ class Command(BaseCommand):
     def handle(self, *args, **kwargs):
         verbosity = kwargs['verbosity']
         maarch = get_maarch()
+
+         if not maarch:
+             if verbosity > 1:
+                 self.stdout.write('Maarch not configured')
+             return

         maarch_mails = maarch.get_mails()
         count = 0
```


0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch	718 octets	19 mai 2018	Benjamin Dauvergne
0002-qualif-store-keep-formdata-backoffice-URL-22550.patch	2,89 ko	22 mai 2018	Benjamin Dauvergne
0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch	718 octets	22 mai 2018	Benjamin Dauvergne
0004-handle-rejection-of-incoming-mails-to-be-rebased.patch	3,05 ko	22 mai 2018	Benjamin Dauvergne
0002-qualif-store-keep-formdata-backoffice-URL-22550.patch	2,89 ko	29 juin 2018	Benjamin Dauvergne
0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch	718 octets	29 juin 2018	Benjamin Dauvergne
0004-settings-remove-alfortville-default-validation-steps.patch	1,57 ko	29 juin 2018	Benjamin Dauvergne
0003-settings-remove-alfortville-default-validation-steps.patch	1,57 ko	29 juin 2018	Benjamin Dauvergne
0002-qualif-store-keep-formdata-backoffice-URL-22550.patch	2,89 ko	29 juin 2018	Benjamin Dauvergne
0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch	718 octets	29 juin 2018	Benjamin Dauvergne
0003-settings-remove-alfortville-default-validation-steps.patch	1,57 ko	29 juin 2018	Benjamin Dauvergne
0002-qualif-store-keep-formdata-backoffice-URL-22550.patch	2,89 ko	29 juin 2018	Benjamin Dauvergne
0001-setup.py-limit-django-haystack-to-2.8-until-we-are-o.patch	718 octets	29 juin 2018	Benjamin Dauvergne
0001-mail-feed-from-MaarchCourrier-22550.patch	29,7 ko	29 juin 2018	Benjamin Dauvergne
0001-jenkins-add-dependency-django-webtest.patch	896 octets	13 juillet 2018	Christophe Siraut