Connecteur Maarch

Fonctionnalités

Le connecteur est capable de récupérer les courriers dans un certain statut, permettant de lui associer des demandes. Le courrier peut aussi être refusé s'il est jugé qu'il n'est finalement pas du ressort de la GRC. En fin de traitement de la demande il sera possible de lui apporter une réponse, pour l'instant une simple réponse sous forme de texte injecté dans l'historique du courrier.

Scénario détaillé

- un courrier, enregistré dans Maarch, est déposé dans une bannette spécifique, bannette "GRU", par exemple
- ce courrier est alors présenté dans Publik (écran Welco)
- il faut l'associer à un (ou plusieurs) formulaire existant
- compléter le formulaire avec les informations présentes dans le courrier
  traitement classique de la demande via le workfow de Publik
- en cours de traitement (appel webservice) possible
  - d'informer Maarch que le courrier, lié à la demande, a été "traité"
  - de transférer à Maarch un champ texte pour insertion de celui-ci dans un éventuel courrier réponse construit dans Maarch

Cf. recette complète sur Ville d'Avray :
https://dev.entrouvert.org/projects/ville-d-avray/wiki/Sc%C3%A9nario_de_recette_du_connecteur_Maarch%3EPublik

Limitations

1. On ne peut pas répondre à un courrier par un document
2. Coté Publik nous pouvons associer plusieurs démarches à un courrier ce qui dans Maarch ne sera pas possible puisque les champs external_id et external_link n'accepte qu'une unique valeur,
3. La création d'une demande pour un courrier fait passer le statut du courrier dans Maarch dans le statut "GRC_TRT" (voir plus bas), de même on ne peut faire ça qu'une seule fois pour que cela ait un sens,
4. De la même manière le fait de produire plusieurs réponses à un courrier dans Maarch obligera à passer plusieurs fois dans le statut "Réponse émise par la GRC" rendant l'historique peu cohérent.

Pour le deuxième point le ticket #30046 évoque un contournement possible.

Configuration de Maarch

Les ressources (ou courrier, le modèle de donnée n'est pas bien clair) de Maarch doivent posséder les attributs suivants:

- external_id : pour stocker le numéro d'une demande dans w.c.s.
- external_link : pour stocker un lien direct vers la demande dans w.c.s.

Un utilisateur est créé sur Maarch ayant accès aux web-services REST, il servira à welco pour moissonner l'instance de Maarch.

Ensuite il faudra disposer des statuts suivants (qu'on pourra nommer comme on veut, je donne juste les valeurs par défaut du connecteur qui sont configurable):

- Courrier à traiter par la GRC : "GRC"
- Courrier injecté dans la queue de courrier de la GRC : "GRC_TRT"
- Demande créé pour le courrier : "GRC_SENT"
- Erreur d'aiguillage, la demande n'est pas à traiter par la GRC : "GRCREFUSED"
- Une réponse à été apportée par la GRC : "GRC_RESPONSE"
  - ce dernier statut devrait toujours être accompagné d'une note dans l'historique du courrier, contenant le message réponse de la GRC (une simple chaîne de caractère)

Configuration du connecteur (de la brique welco)

La configuration se fait via la clé de configuration Django MAARCH_FEEED, comme cela (à configurer par exemple via hobo en créant une variable SETTING_MAARCH_FEED dans le service courrier)

```
MAARCH_FEED = {
   'ENABLE': True  # optionnel, permet de désactiver le raccordement, par défaut à True
```

```
    'URL': 'https://maarch-api-endoint' # Maarch API endpoint
    'USERNAME': 'xxx'  # obligatoire, username de l'utilisateur dans Maarch
    'PASSWORD': 'yyy':  # obligatoire, mot de passe de l'utilisateur dans Maarc
    'STATUS_GRC': 'GRC'  # optionnel, voir plus haut, valeur par défaut identique
    'STATUS_RECEIVED': 'GRC_TRT'  # optionnel, voir plus haut, valeur par défaut identique
    'STATUS_SEND': 'GRC_SENT'  # optionnel, voir plus haut, valeur par défaut identique
    'STATUS_REFUSED': 'GRC_REFUSED'  # optionnel, voir plus haut, valeur par défaut identique
    'STATUS_RESPONSE': 'GRC_RESPONSE'  # optionnel, voir plus haut, valeur par défaut identique
}
```

Utilisation dans un workflow w.c.s.

On retrouvera dans le contexte de soumission de la demande le numéro du courrier dans Maarc via la variable form_submission_context_external_id sour la forme 'maarch-%(res_id)s'.

Envoyer une réponse depuis w.c.s.

Pour cela on utilisera une API exposée par la brique courrier, en supposant que la brique courrier se nomme courrier, on pourra créer un appel de web service dans une action de traitement dans w.c.s. avec les paramètres suivants :

- URL: {{courrier_url}}api/mail/response/
- Méthode: POST
- Données à envoyer dans le corps de la requête

Nom de la clé	Valeur
content	le message que l'on souhaite envoyer, une chaîne de caractère simple
mail_id	{{ form_submission_context_external_id }}

Description du code dans welco

Le numéro du courrier dans Maarch (le res_id) est conservé sous la forme formatée 'maarch-%(res_id)s' dans le champ external_id (à ne pas confondre avec l'external_id dans Maarch) du modèle Mail (dans weco/sources/mail/models.py). C'est la seule donnée venant de Maarch qui soit conservée dans welco.

Fichier	Description
welco/sources/mail/maarch.py	l'objet de base pour interagir avec Maarch, implémente les appels de WS à Maarch
welco/sources/mail/utils.py	un adapteur pour intégrer Maarch aux besoin de Welco, récupérer la configuration depuis les settings Django
welco/sources/mail/management/commands/feed_mail_maarch.py	commande lancé par cron pour moisonner régulièrement Maarch
welco/sources/mail/__init__.py	réaction au signal créant une association entre un courrier et une demande dans w.c.s. pour le signaler à Maarch
welco/sources/mail/views.py	modifie le comportement en cas de rejet d'un courrier venant de Maarch (notification de Maarch), implémentation du WS à destination de w.c.s. pour émettre une réponse à un courrier Maarch

Description des WS Maarch utilisés

Nous utilisons les web-service suivants:

- rest/res/list pour obtenir la liste des courriers dans un certain statut avec leur contenu (withFile=true)
  - dans welco.sources.mail.maarch.MaarchCourrier.get_couriers(clause, fields=None, limit=None,·include_file=False,·order_by=None)
- rest/res/externalInfos pour modifier le statut d'un courrier et définir les attributs external_id et external_link
  - dans welco.sources.mail.maarch.MaarchCourrier.update_external_infos(self, courriers, status)

- rest/res/resource/status pour modifier le statut d'un courrier et ajouter un message à son historique
  - dans welco.sources.mail.maarch.MaarchCourrier.update_status(self, courriers, status, history_message=None)

Documentation : https://docs.maarch.org/gitbook/html/MaarchCourrier/18.10/guat/guat_architecture/API_REST/home.html

Quelques requêtes :

```
# Courriers en statut GRC
curl -u 'login:passwd' -d "select=*&clause=status='GRC'&orderBy[]=res_id" -H "Content-Type: applic
ation/x-www-form-urlencoded" -X POST https://maarch-api/rest/res/list

# Courriers en statut  GRC*
curl -u 'login:passwd' -d "select=*&clause=status like 'GRC%'&orderBy[]=res_id" -H "Content-Type:
application/x-www-form-urlencoded" -X POST https://maarch-api/rest/res/list

# Tous les courriers
curl -u 'login:passwd' -d "select=*&clause=1=1&orderBy[]=res_id" -H "Content-Type: application/x-w
ww-form-urlencoded" -X POST https://maarch-api/rest/res/list
```

Admin système

Récupération des courriers :

```
sudo -u welco welco-manage tenant_command feed_mail_maarch --domain nom_du_tenant -v2
```