Création de brouillon de mail

```
def drafts_mails(self, category: str = 'test'):
"""
:param category: variable String correspond à une étiquette portant ce
nom
:return: permet de créé un brouillon de mail type aux mails portant
l'étiquette
"""
```

Ouverture fichier json

```
# lecture fichier json
    with open('brouillons_types.json') as json_file:
        data = json.load(json_file)
```

```
#fichier json
{
  "brouillons": [
    {
      "brouillonType": "test",
      "subject": "Mon object",
      "body": "message <br> en format html"
    },
    {
      "brouillonType": "categorie assurance",
      "subject": "Mon object",
      "body": "message <br> en format html"
    },
    {
      "brouillonType": "categorie carte de grise",
      "subject": "Mon object",
      "body": "message <br> en format html"
    }
  ]
}
```

Récupération des informations

```
# récupération des infos du brouillon selon la catégorie
header, body = None, None
for draftType in data['brouillons']:
    if draftType['brouillonType'] == category:
        header, body = draftType['subject'], draftType['body']
        break
```

récupération des infos du DataFrame

```
if self.dic_target_folders == {}:
    self.mail_connexion()
    self.target_selection()
```

NB: permet de récupérer les dossiers cibles

Création des brouillons

```
nb_drafts_total = 0
print("\n Création des brouillons ")
for folder_name in self.dic_target_folders:
   nb_drafts_total +=
self.exchange_connector.draft_email(header=header, body=body,
mail_acc=folder_name, category=category)

print(f"      - Nombre de brouillon(s) enregistré(s) :
{nb_drafts_total}")
```

NB: Utilisation de la méthode draft_email créé au préable dans le fichier exchange de Melusine

Fichier Exchange

```
    def draft_email(  # TODO TITOUAN rajout de la méthode
            self, header: str = "", body: str = "",
            mail_acc=None, category: str = "test"
    ):

        """
        :param header: L'objet du mail
        :param body: Le corps du mail
        :param mail_acc: Le dossier au quel est rattaché le mail
        :param category: L'étiquette de triage des mails
        :return: le nombre de brouillon créé par dossier
        """
        if mail_acc is None:
            raise AttributeError(
                "Un chemin de dossier doit être spécifié ! "
            )

    def drafts_mails(self, category: str = 'test'):
        """
        :param category: variable String correspond à une étiquette
portant ce nom
        :return: permet de créé un brouillon de mail type aux mails
portant l'étiquette
        """
```

Récupération des mails portant l'étiquette

```
#récupération des message_id avec le filtre catégories (étiquette)
        logger.info(f"Reading new emails for mailbox
'{self.mailbox_address}'")
        base_folder = self._get_mailbox_path(mail_acc)
```

```
        all_new_data =
(base_folder.all().only("message_id").filter(Q(categories=category) &
Q(is_draft=False)))

        tab_id = [
            x.message_id
            for x in all_new_data
            if isinstance(x, Message)
        ]
# Création d'un tableau de message_id
```

class Account

On va créé notre brouillon au même endroit que son mail initial

```
#transformation de la donnée mail_acc en class Account
folder = mail_acc.split('/')
mail_acc = self.mailbox_account.inbox
for f in folder:
    mail_acc = mail_acc / f
```

Création du brouillon

On utilise la méthode reply() de la class Message pour créer notre brouillon de mail

```
#création du brouillon
i = 0
for message_id in tab_id:
    original_message = mail_acc.filter(message_id=message_id).all()[0]
# filtrer par le message id.
    #recherche de brouillon pour le message id
    has_draft = mail_acc.filter(references=message_id).all() #permet
de savoir si le mail à déjà un brouillon de créé
    if has_draft.count() == 0:
        original_message.reply(subject=header,
                               body=HTMLBody(body), folder=mail_acc)
        original_message.save()

        i += 1
if (i > 0):
    print(12 * " ", " - Nombre de brouillon(s) total enregistré(s)
dans le dossier", mail_acc, ":", i)

return i
```