

Mail-Manager — Manuel test

Méthodes utiles aux tests

- `essai_send_mail(<path>)`. Simulation de l'envoi d'un mail

`method-essai_send_mail(<path>)`

Cette méthode procède à l'envoi du mail défini dans le `<path>`. Au lieu de les envoyer, il met les mails dans le dossier `tmp/history/mails` (le dossier normal où sont toujours placés les mails envoyés). Noter que chaque test (chaque appel à cette méthode) nettoie ce dossier, c'est-à-dire le vide, pour savoir toujours exactement les mails envoyés au cours du test qui utilise cette méthode.

Si le fichier mail localisé à `<path>` se trouve dans `test/assets/source_files` alors on peut obtenir `<path>` à l'aide de :

```
path = Factory.source_file_path(`<name>`)
```

Par exemple :

```
path = Factory.source_file_path('with_exclusions')
```

Attention : aucune vérification de l'existence du fichier n'est exécutée ici. Il ne faut pas se tromper de nom ou bien vérifier que le fichier existe.

Vérification des mails

La vérification des mails envoyés se fait à l'aide des méthodes d'assertion propres aux mails :

Assertions

```
assert_mail_received_by(mail[, <subject>)  
# => Produit un succès si <mail> a bien reçu un message  
  
Ou  
  
assert_mail_received_by(mail, {params})  
  
ET  
  
refute_mail_received_by
```

Avec `<params>` qui peut contenir :

```
{
  subject: "Le sujet" ou /Le sub?jte/,
  content: ["contenu", "contenu", /contenu/, etc.],
  html_content: "<div>Bonjour</div>",
  from: "mail_du@sender.com",
  after: date, # le mail doit être envoyé après cette date
  before: date # le mail doit être envoyé avant cette date
}
```

NOTES

`content` ne recherche que dans le contenu text/plain du mail

`html_content` ne recherche que dans le contenu text/html du mail.