Email Tracking

Projet : Ubiposting.Worker.EmailTracking

## Introduction

EmailTracking va récolter dans la boite mail 1&1 les emails envoyés par les JobBoard qui concerne des candidatures.

On va trier ces mails en fonction de plusieurs critères et renvoyer les notifications aux clients

## Récolte des mails anciennement GDC

Cette méthode récupère les mails de type « exemple.M12345 » mais aujourd’hui, ce format d’adresse mail n’est plus utilisé

## Récolte des autres mails

On commence par se connecter à la boite mail 1&1 pour sélectionner les 100 premiers mails dans la « Boite de réception »

Ensuite on va boucler sur chaque mail :

## Création d’une copie

Pour chaque mail, on va d’abord en créer une copie car il y a un bug dans le client « Lenikowski »

// creating another mail message type because of bug in lenikowski client. In some cases, subject doesn't display

* On copie les pièces jointes
* On copie le corps du message
* On copie les destinataires (To, cc et Bcc)
* On copie le sujet

Pour le sujet, on va d’abord remplacer les charactères spéciaux issus de

## Check des adresses de destination

On regarde pour chaque destinataire si l’adresse mail correspond aux formats specific ou regular

Le but est de retrouver l’ID de l’offreJobBoard

**Format Specific ( exemple :** [**stm.585@applicount.com**](mailto:stm.585@applicount.com)**) :**

On recherche l’ID de l’entreprise dans l’adresse mail puis l’ID de l’offre JobBoard dans le sujet

Regex entrepsie ID : @"^stm.(?<entid>\d{2,9})"

Regex OffreJobBoard ID : @"\s{1}(?<ojid>\d{5,9})"

ATTENTION : Si on a au moins l’adresse entreprise, on continue quand même le traitement

**Format Regular ( exemple :** [**aeos-consultants.3668397@applicount.com**](mailto:aeos-consultants.3668397@applicount.com)**) :**

On recherche l’Offre JobBoard ID directement dans l’adresse mail

Regex OffreJobBoard ID : @"(?<ojid>\d{3,9})"

Dans les deux cas, si l’adresse mail ne Matche pas, on quitte la récolte

## JunkMail

## Filtre des mails en fonctions des règles Globales et entreprise

## On va vérifier s’il y a des mails qui ne doivent pas être pris en compte en fonction de l’adresse mail de l’expéditeur et du sujet du mail

**Règles globales :**

On va commencer par aller charger la liste des règles dans la table « CP\_JunkMailRules ».

Chaque item de cette liste possède une adresse mail et un sujet de mail.

On va les comparer avec ceux de notre mail selon ces cas :

* Si on a un sujet et une adresse mail, on check les 2
* Si on a un sujet mais pas d’adresse mail, on check uniquement le sujet
* Si on a une adresse mail mais pas de sujet, on check uniquement l’adresse mail

**Règles d’entreprise :**

On applique les règles d’entreprise uniquement si on a un OffreJobBoardID

On va commencer par aller charger la liste des règles dans la table « CP\_JunkCompanyRules » pour notre entrepriseID.

Chaque item de cette liste possède une adresse mail et un sujet de mail.

On va les comparer avec ceux de notre mail selon ces cas :

* Si on a un sujet et une adresse mail, on check les 2
* Si on a un sujet mais pas d’adresse mail, on check uniquement le sujet
* Si on a une adresse mail mais pas de sujet, on check uniquement l’adresse mail

**Traitement des règles :**

Si nous avons une règle qui Matche et que nous avons bien un OffreJobBoardID, on fait ce traitement :

On créé un « BlockedMessage » et on le remplie avec les infos de notre mail.

Si on a une OffreJobBoardID :

On créé le répertoire suivant s’il n’existe pas (sur Web1) :

c:/Logs/Applicount/ApplicountAttachement

Et pour chaque pièce jointe qui possède un nom et qui a une extension comprise dans la liste suivante (".doc",".docx",".pdf",rtf",".txt",odt",ott",oth",odm",docm",wri"), on fait les traitements suivants :

* On créé le répertoire lié à nos pièces jointes s’il n’existe pas :

C:\\Logs\\Applicant\\ApplicantAttachment\\2020\_08\\3652706\\

Avec :

« 2020\_08 » correspond à l’année + mois

« 3652706 » correspond à l’OffreID

* On créé un fichier (FileStream) à cet emplacement en copiant le memoryStream de notre pièce jointe.   
  Le nom de ce fichier sera sous ce format : OffreJobBoardID\_NewGUID.Extension

32951941\_6dc9e698-2da5-4be8-ae94-00e56cf2f109.docx

* Dans la liste « attachementsfileNames » de notre objet « BlockedMessage », on ajoute le lien qu’aura le fichier une fois synchronisé sur App.

D:\IIS\Applicant\ApplicantAttachment\2020\_08\3652706\32951941\_67686286-31e9-4984-ba24-c33ab6c06cae.docx

Avec :

« 2020\_08 » correspond à l’année + mois

« 3652706 » correspond à l’OffreID

« 32951941\_6dc9e698-2da5-4be8-ae94-00e56cf2f109.docx » correspond au nom du fichier

On termine par sauvegarder le BlokedMessage dans la table « CP\_BlockedMessages »

## Initialisation des settings de l’entreprise

Si le mail n’a pas de règle Globales ou d’entreprise ou si nous n’avons pas d’OffreJobBoardID,   
On initialise l’entreprise et l’entrepriseSetting

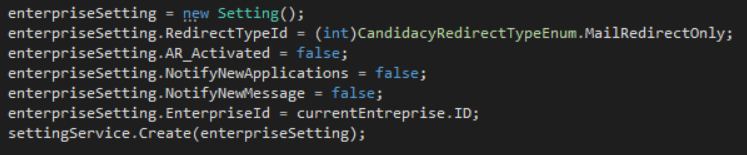
2 façons de récupérer l’entreprise :

* Si on possède l’EntrepriseID mais pas d’offreJobBoardID, on va récupérer l’entreprise dans la table « Entreprises »
* Sinon si on a un OffreJobBoardID et que c’est un INT, On récupère l’OffreJobBoard dans la table « OffreJobBoard ». On pourra donc y récupérer l’Offre puis l’Entreprise.

offreJobboard.offre.entreprise

Pour récupérer les settings, on va les chercher dans la table « CP\_Settings » grâce à EntrepriseID

Si on ne trouve pas de Settings pour cette entreprise, on en créé un puis on l’insert en base.



## Traitement des mails Candidature

## Traitement de l’offre archivée

Si on a l’offreJobBoardID et que la lastCommande est une commande « Delete » OU que l’offre est archivée, alors on prévient le candidat que l’offre n’est plus à pourvoir.

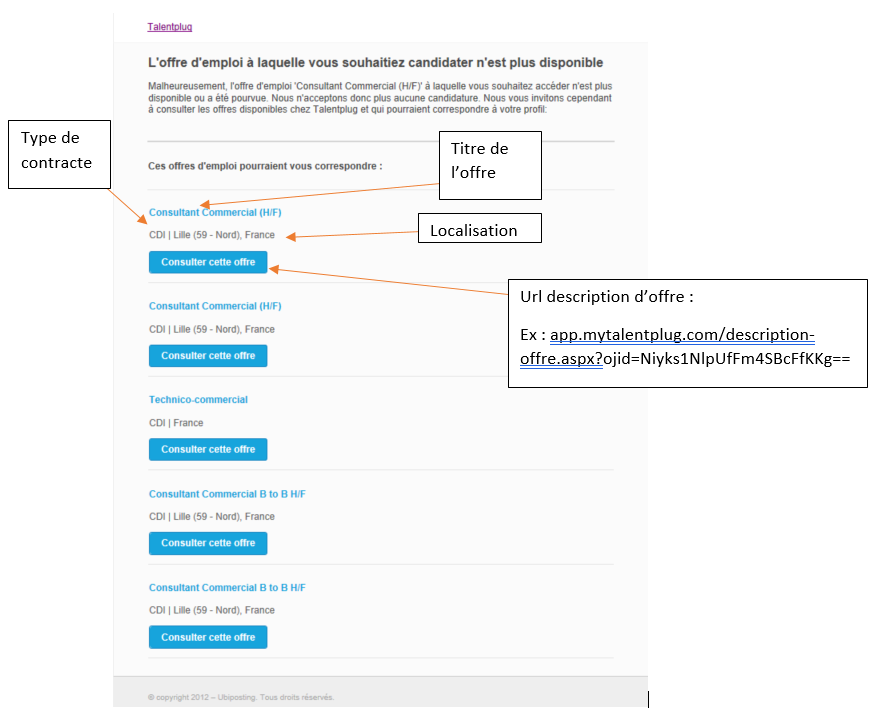
On créé un mail (Lesnikowski) avec :

- Destinataire : Candidat si possible, sinon l’adresse du JobBoard

- Expéditeur : [no-reply@talentplug.com](mailto:no-reply@talentplug.com)

- Sujet : Merci de votre candidature. Malheureusement le poste n’est plus disponible...

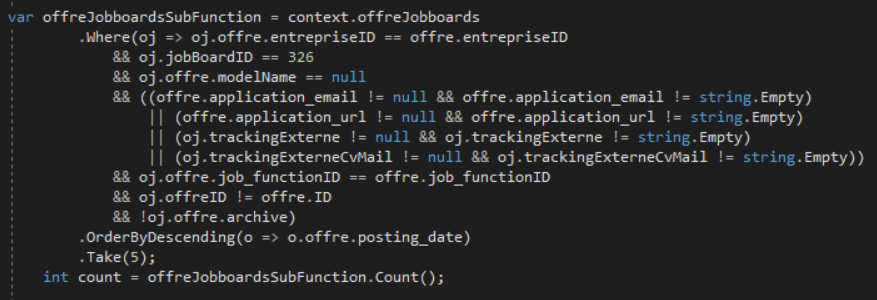
- Corps :



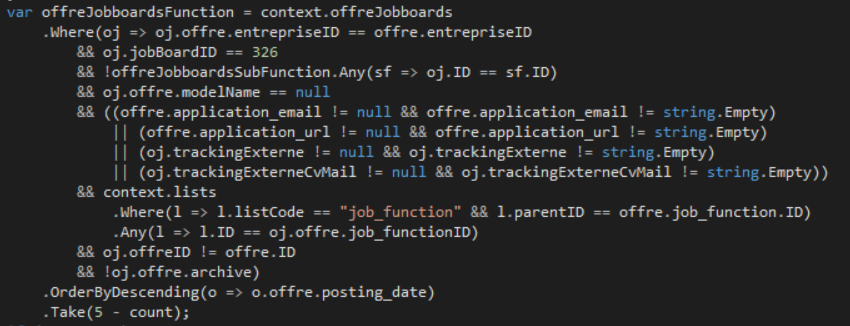
* Le logo
  + Si logo : On intègre le logo s’il y en a un (« entreprise.logo ») avec le lien de l’entreprise (si possible)
  + Si pas de logo : On met le lien de l’entreprise (si possible, sinon on ne met que le nom)
* Le nom de l’entreprise : Si l’entreprise a choisie d’être anonyme, on n’affiche pas la phrase « Nous vous invitons cependant à consulter les offres disponibles chez " +

offre.entreprise.raisonSociale + @" et qui pourraient correspondre à votre profil: »

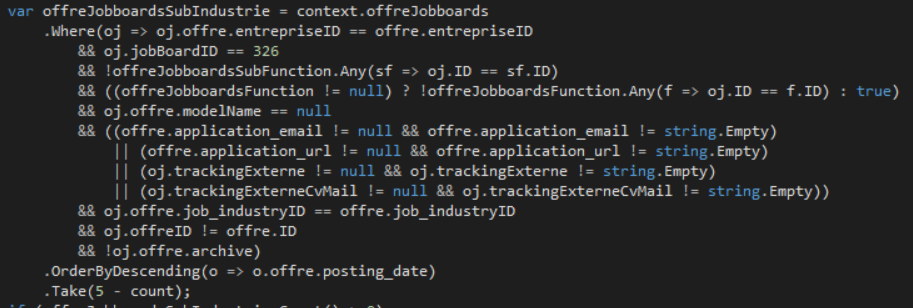
* Les propositions diverses (5 max)
  + Les propositions avec la même fonction



* + Les propositions qui ont le même parent de fonction (job\_function)



* + Les propositions qui ont le même secteur d’activité (job\_industry)



Les propositions redirigent le candidat vers la page « description-offre » qui est une page TalentPlug.

On termine par envoyer le mail au candidat

Sinon on envoie les mails au candidat et aux client

## Sauvegarde de l’Email en BDD et des fichiers joints sur le serveur

Dans le cas où nous avons une offre qui est toujours disponible et que le mode de réception (Application\_mode) de l’offre est « GDC » et que le mode de candidature (RedirectTypeID) est différent de « Email uniquement » (MailRedirectOnly) alors on sauvegarde le mail en base de données ainsi que les pièces jointes sur le serveur.

Création de la candidature

On commence par vérifier si la candidature n’existe pas avec les éléments suivants :

* date de réception (date d’aujourd’hui si date vide)
* Nom, prénom et mail du candidat (si email contient un expéditeur)
  + Si expéditeur vide : nom = « Anonyme », prénom et mail vide

Si on n’a pas le mail du candidat, on prend le mail du JobBoard.

On lance la recherche dans la table « CP\_candidaties » afin de voir si cette candidature existe déjà

Les critères de recherche : nom, prénom, email, offreJoboardId, date de réception

Dans le cas où la candidature n’existe pas en base, on la créé avec les infos suivantes :

|  |  |
| --- | --- |
| **Nom de la propriété** | **Valeur** |
| Email | Mail du candidat si existe, sinon mail jobBoard |
| Rating | 0 |
| RecruitementStatus | InProgress |
| ApplicationStatus | L’ID première valeur de la table « CP\_applicationstatus », sinon 1 |
| ReceipDate | Renseignée si la date du mail reçu est connue |
| OffreJobBoardID | OffreJobBoardID |
| FirstName | Si l’expéditeur du mail (From) de candidature est connu et qu’il possède un « name » :  La première partie du nom de l’expéditeur du mail de la candidature  Sinon si l’expéditeur est connu mais pas le « name »:  On prend le préfixe de l’adresse mail (avant @)  Sinon, on met « Anonyme » |
| LastName | Si l’expéditeur du mail (From) de candidature est connu et qu’il possède un « name » :  La deuxième partie du nom de l’expéditeur du mail de la candidature  Sinon :  On met cette donnée à vide |
| IPAdresse | On récupère l’adresse IP du mail de candidature |
| EmailExternalID | On récupère le dernier EmailExternalID de la table « CP\_candidaties » puis on l’incrémente de 1 |

Création du message

On enregistre ensuite le message en base dans la table « CP\_Messages » (même si la candidature n’existait pas en base)

|  |  |
| --- | --- |
| **Nom de la propriété** | **Valeur** |
| Subject | Mail.subject |
| Body | Si le mail contient un « HtmlDataString », on le prend  Sinon si le mail contient un « TextDataString », on le prend (en remplaçant « \r\n » par « <br /> » |
| SendData | Mail.Data |
| TypeID | 0 |
| MessageModelID | 1 |
| CandidatyID | ID de la candidature créé précédemment |
| ThreadId | L’adresse IP du mail si c’est du Int.  Sinon on prend l’EmailExternalID de la candidature |

Création des pièces jointes

On regarde maintenant si le mail de candidature possède des pièces jointes.  
Si c’est le cas, on sauvegarde les fichiers sur le serveur et les liens sur la base :

On créé le répertoire suivant s’il n’existe pas (sur Web1) :

c:/Logs/Applicount/ApplicountAttachement

Et pour chaque pièce jointe qui possède un nom et qui a une extension comprise dans la liste suivante (".doc",".docx",".pdf",rtf",".txt",odt",ott",oth",odm",docm",wri"), on fait les traitements suivants :

* On créé le répertoire lié à nos pièces jointes s’il n’existe pas :

C:\\Logs\\Applicant\\ApplicantAttachment\\2020\_08\\3652706\\

Avec :

« 2020\_08 » correspond à l’année + mois

« 3652706 » correspond à l’OffreID

* On créé un fichier (FileStream) à cet emplacement en copiant le memoryStream de notre pièce jointe.   
  Le nom de ce fichier sera sous ce format : OffreJobBoardID\_NewGUID.Extension

32951941\_6dc9e698-2da5-4be8-ae94-00e56cf2f109.docx

* Dans la table « CP\_Attachments », on créé un nouvel attachement (Entities.Attachment) avec le lien vers le dossier de stockage sur App

D:\IIS\Applicant\ApplicantAttachment\2020\_08\3652706\32951941\_67686286-31e9-4984-ba24-c33ab6c06cae.docx

Avec :

« 2020\_08 » correspond à l’année + mois

« 3652706 » correspond à l’OffreID

« 32951941\_6dc9e698-2da5-4be8-ae94-00e56cf2f109.docx » correspond au nom du fichier

Dans la table « CP\_CandidacyAttachment », on enregistre un nouvel élément

|  |  |
| --- | --- |
| **Nom de la propriété** | **Valeur** |
| CandidacyId | Candidature.Id |
| AttachmentId | Attahcment.Id (précédemment créé) |

Dans la table « CP\_MessageAttachment », on enregistre un nouvel élément

|  |  |
| --- | --- |
| **Nom de la propriété** | **Valeur** |
| MessageId | message.Id |
| AttachmentId | Attahcment.Id (précédemment créé) |

## Formatage du sujet du mail

## Si l’entreprise a un wsPartenaire (ATS), on récupère le format du sujet d’un mail (formatsujet).

S’il y en a un, on vient remplacer les variables par ces textes :

|  |  |
| --- | --- |
| **Variable dans le format** | **Valeur à insérer** |
| %sender% | Le nom de l’expéditeur du mail de candidature S’il y en a un  Sinon on prend le préfixe du mail de l’expéditeur (avant le @) |
| %sujet% | Le sujet du mail de candidature |
| %job\_reference% | Le « job\_Reference » de l’offre en remplaçant « \n » et « \r » par vide  Puis on supprime les espaces en début et fin |
| %job\_title% | Le « job\_title » de l’offre en remplaçant « \n » et « \r » par vide  Puis on supprime les espaces en début et fin |
| %jobboard% | Le nom du jobboard (offrejobboard.jobboard.name) |
| %jobboard\_id% | L’ID de l’offreJobBoard |
| %jobboard\_mapped% | Le nom du JobBoard mappé pour l’entreprise  (table mappingJobboards) S’il y a  Sinon, on affiche l’ID du JobBoard |

Une fois le sujet formaté, on va charger la liste des caractères spéciaux issue de la table « SpecialCharsMapping » (en cache) avec le listCode « track » qu’on réorganise par ID

On vient ensuite boucler sur chaque item de cette liste afin de remplacer les anciennes valeurs par les nouvelles.

Si le caractère spécial est un Regex (item.isRegex), on applique ce regex à notre sujet  
Sinon on remplace simplement l’ancien caractère par le nouveau.

En retourne le nouveau sujet.

Ensuite on remplace le sujet des mail de candidature « mail » ainsi que le mail que on reconstruit dans le début de traitement « mailMsg » par le nouveau sujet.

## Envoi d’un accusé de réception au candidat

Si l’entreprise qui a publiée l’offre est paramétrer avec un «forceUrl = false » en envoie un mail automatique à l’adresse de candidat que on recouper de mail de candidature « mail.ReplyTo » si il existe sinon on l’envoie a l’adresse de Sender de la candidature .

Construction du mail :

* On récupère à partir de l’entrepriseId le setting de l’entreprise dans la table « Cp\_Setting »
* Si « stting.AR\_Activated et setting.MessageModelId >0 » on fait ce qui suit :
* On récupère le messageModel dans la table « CP\_MessageModel » on utilisant setting.MessageModelId
* On créé une variable de addressSend qui va envoyer l’email sous la forme « [noreply@applicount.com](mailto:noreply@applicount.com) »
* On récupère la liste des attachements de message model dans la table « CP\_MessageModelAttachement » en utilisant messageModel. MessageModelId

## Finalisation de la candidature

## Envoi du mail au client (Application email au STEP 2)

## Envoi de la notification à l’utilisateur qui a créé l’offre

## On déplace le mail dans la boite mail