Pour interroger les serveurs mail afin de savoir si une adresse est invalid sans envoyer de mail on a le code node de pierre.

# Campaigns

Récupérer les campagnes déjà existante :

Il suffit d’instancier une liste d’objets Campaign et un objet CampaignService(String apiKey), puis grace à la méthode retrieveAllCampaigns() de la classe CampaignService on pourra récupérer la liste des campagnes déjà. La seul information à fournir est la clé API de l’utilisateur lors de l’instanciation de l’objet CampaignService.

Récupérer une campagne en particulier :

On instancie un objet CampaignService(String apiKey) à l’aide de la clé API. Puis on peut utiliser la méthode retrieveSingleCampaign(String campaignId). Cette méthode renvoie un objet Campaign.

Delete une campagne :

On instancie un objet CampaignService(String apiKey) à l’aide de la clé API. Puis on peut utiliser la méthode deleteSingleCampaign(String campaignId). Cette méthode ne renvois rien, elle supprimer la campagne de l’id correspondant.

# Recipients

On instancie un objet RecipientService(String apiKey, File file) et c’est à partir de cet objet qu’on fera toute les manipulations sur les recipients (les contacts de la liste).

Envoyer un email à une liste de contact :

|  |
| --- |
| publicvoidsendTo(List<RecipientFromCsvFile>pRecipientList,EmailpFrom, String pSubject, Content pContent) |

Cette méthode ne renvois rien, elle enverra le mail (pContent) à la liste de contacts (pRecipientList) à partir de l’adresse que l’on lui aura indiqué (pFrom) avec son objet (pSubject).

Récupérer tous les contacts :

public List<RecipientFromSendgrid> retrieveAllRecipientsFromSendgrid()   
Cette méthode renvoie une liste d’objet contact. Après avoir instancier l’objet RecipientService, il suffit d’appeler cette méthode pour récupérer tous nos contacts dans une liste.

# Les stats

Les blocks :

public void FindAllBlock()

A utiliser à partir d’un objet RecipientService. Affiche dans la console tous les contacts qui sont détectés comme des blocks.

Les bounces :

public void findAllBounce()

A utiliser à partir d’un objet RecipientService. Affiche dans la console tous les contacts qui sont détectés comme des bounces.

Les invalids :

public void findAllInvalid()

A utiliser à partir d’un objet RecipientService. Affiche dans la console tous les contacts qui sont détectés comme des invalids.

Les spams :

public void findAllSpam()

A utiliser à partir d’un objet RecipientService. Affiche dans la console tous les contacts qui sont détectés comme des spams.

# Template

Créer un nouveau template :

public void createTemplate(String name)

Apres avoir instancié un objet TemplateService avec une clé API, il suffit d’utiliser la méthode createTemplate(String name) à partir de l’objet TemplateService en renseignant dans le paramètre name le nom que l’on veut donner à notre nouveau template.

Créer une nouvelle version de template :

|  |
| --- |
| public void createNewTransactionalTemplateVersion(StringversionName,FilehtmlContent,File plainContent,int active, String templateId, String subject)  Toujours après avoir instancié un objet TemplateService. On peut se servir de cette méthode pour créer une version de ce template. On rentre en parametre le fichier texte html du template, le nom que l’on veut donner à cette version, le fichier texte du plain content, l’id du template dont on veut créer une nouvelle version et enfin l’objet du mail dans le template. |
|  |

Récupérer tous les templates :

public List<Template> retrieveAllTemplates()

Après avoir instancié un objet TemplateService. On peut se servir de cette méthode pour récupérer tous les templates existants.

En résumé :

* On sait comment créer un template/une liste de contact à partir d’un fichier csv/ une campagne/ envoyer un mail à la liste
* On sait envoyer un mail à des contacts ainsi qu’un template
* On sait comment récupérer les stats sur les emails Bounces / Blocks / Invalid / Spam

Créer une campagne :

Request.method = Method.POST ;

Request.endpoint = « campaigns »

Request.body = "{\"custom\_unsubscribe\_url\":\"\",\"html\_content\":\"<html><head><title></title></head><body><p>Check out our spring line!</p></body></html>\",\"list\_ids\":[110,124],\"sender\_id\":124451,\"subject\":\"New Products for Spring!\",\"plain\_content\":\"Check out our spring line!\",\"suppression\_group\_id\":42,\"title\":\"March Newsletter\",\"segment\_ids\":[110],\"categories\":[\"spring line\"],\"ip\_pool\":\"marketing\"}";

V3 MAIL SEND

request.body = "{\"custom\_args\":{\"New Argument 1\":\"New Value 1\",\"activationAttempt\":\"1\",\"customerAccountNumber\":\"[CUSTOMER ACCOUNT NUMBER GOES HERE]\"},\"from\":{\"email\":\"sam.smith@example.com\",\"name\":\"Sam Smith\"},\"attachments\":[{\"name\":\"file1\",\"filename\":\"file1.jpg\",\"content\":\"[BASE64 encoded content block here]\",\"disposition\":\"inline\",\"content\_id\":\"ii\_139db99fdb5c3704\",\"type\":\"jpg\"}],\"personalizations\":[{\"to\":[{\"email\":\"john.doe@example.com\",\"name\":\"John Doe\"}],\"cc\":[{\"email\":\"jane.doe@example.com\",\"name\":\"Jane Doe\"}],\"bcc\":[{\"email\":\"sam.doe@example.com\",\"name\":\"Sam Doe\"}],\"custom\_args\":{\"New Argument 1\":\"New Value 1\",\"activationAttempt\":\"1\",\"customerAccountNumber\":\"[CUSTOMER ACCOUNT NUMBER GOES HERE]\"},\"headers\":{\"X-Accept-Language\":\"en\",\"X-Mailer\":\"MyApp\"},\"send\_at\":1409348513,\"substitutions\":{\"type\":\"object\",\"id\":\"substitutions\"},\"subject\":\"Hello, World!\"}],\"subject\":\"Hello, World!\",\"ip\_pool\_name\":\"[YOUR POOL NAME GOES HERE]\",\"content\":[{\"type\":\"text/html\",\"value\":\"<html><p>Hello, world!</p><img src=[CID GOES HERE]></img></html>\"}],\"headers\":{},\"asm\":{\"groups\_to\_display\":[1,2,3],\"group\_id\":1},\"batch\_id\":\"[YOUR BATCH ID GOES HERE]\",\"tracking\_settings\":{\"subscription\_tracking\":{\"text\":\"If you would like to unsubscribe and stop receiveing these emails <% click here %>.\",\"enable\":true,\"html\":\"If you would like to unsubscribe and stop receiving these emails <% clickhere %>.\",\"substitution\_tag\":\"<%click here%>\"},\"open\_tracking\":{\"enable\":true,\"substitution\_tag\":\"%opentrack\"},\"click\_tracking\":{\"enable\":true,\"enable\_text\":true},\"ganalytics\":{\"utm\_campaign\":\"[NAME OF YOUR REFERRER SOURCE]\",\"enable\":true,\"utm\_name\":\"[NAME OF YOUR CAMPAIGN]\",\"utm\_term\":\"[IDENTIFY PAID KEYWORDS HERE]\",\"utm\_content\":\"[USE THIS SPACE TO DIFFERENTIATE YOUR EMAIL FROM ADS]\",\"utm\_medium\":\"[NAME OF YOUR MARKETING MEDIUM e.g. email]\"}},\"mail\_settings\":{\"footer\":{\"text\":\"Thanks,/n The SendGrid Team\",\"enable\":true,\"html\":\"<p>Thanks</br>The SendGrid Team</p>\"},\"spam\_check\":{\"threshold\":3,\"post\_to\_url\":\"http://example.com/compliance\",\"enable\":true},\"bypass\_list\_management\":{\"enable\":true},\"sandbox\_mode\":{\"enable\":false},\"bcc\":{\"enable\":true,\"email\":\"ben.doe@example.com\"}},\"reply\_to\":{\"email\":\"sam.smith@example.com\",\"name\":\"Sam Smith\"},\"sections\":{\"section\":{\":sectionName2\":\"section 2 text\",\":sectionName1\":\"section 1 text\"}},\"template\_id\":\"[YOUR TEMPLATE ID GOES HERE]\",\"categories\":[\"category1\",\"category2\"],\"send\_at\":1409348513}";

Response response = sg.api(request);