

Guide utilisateur API EMAIL



Support API	Version du guide API	Date de publication
Arnaud Gaudillere Email : a.gaudillere@mediasflow.com	V1.4	02/03/2023

Table des matières

1	PRESENTATION	3
1.1	Contexte	3
1.2	Spécifications générales de l'API	3
2	FONCTIONS DISPONIBLES.....	4
2.1	Demande d'envoi EMAIL (/campaigns/inject)	4
2.1.1	Exemple de requête d'envoi simple - SANS validation préalable.....	7
2.1.2	Exemple de requête d'envoi simple - AVEC validation préalable.....	8
2.2	Récupération de rapport d'une campagne au format CSV (/campaigns/report/{campaignid})	11
2.3	Récupération de rapport d'une campagne au format JSON (/campaigns/reportjson/{campaignid})..	11
2.4	Récupération de rapport de campagnes sur une période donnée au format JSON (/campaigns/reportjson/)	12
2.5	Récupération de rapport de campagnes sur une période donnée au format CSV (/campaigns/report/)	12
2.6	Annulation d'une campagne programmée (/campaigns/{campaignid}/cancel)	13
2.7	Ajout d'une adresse EMAIL en blacklist (/blacklist).....	13
2.7.1	Exemple de requête pour ajouter une adresse EMAIL en blacklist	14
2.8	Déposer une liste de destinataires (/lists/upload)	14
2.8.1	Exemple de retour envoyée par l'API suite à une requête pour déposer une liste de destinataire....	14
2.8.2	Liste des paramètres dans le retour API suite à une requête pour déposer une liste de destinataire	16
3	CODE ERREURS.....	18
4	DESCRIPTION DES COLONNES DANS LES RAPPORTS AU FORMAT CSV.....	18

1 PRESENTATION

1.1 Contexte


Ce guide a pour objectif de vous accompagner dans l'implémentation de notre API EMAIL.

Les APIs / webservices proposés par Mediasflow permettent aux développeurs d'intégrer l'envoi de message EMAIL (et de SMS) unitaire ou par lots depuis vos propres logiciels, CRM, ERP, site web et applicatifs métiers.

Cette API se compose de fonctions accessibles directement par HTTPS selon une architecture de type REST.

Afin d'utiliser notre API EMAIL, il est nécessaire d'être familiarisé avec au moins un langage de programmation tels que : .NET, JAVA, PHP, NodeJS, (liste non exhaustive).

1.2 Spécifications générales de l'API

URLs	Environnement de production : https://app.mediasflow.com/MediasflowApi/v1/
Méthode d'appel	Méthode POST, PUT GET et DELETE avec encodage UTF-8 et Content-Type: application/x-www-form-urlencoded. Le paramétrage des différentes fonctions est faite par des structures JSON intégrées dans les POST.
Sécurité	Requêtes HTTPS uniquement. Authentification par nom d'utilisateur / mot de passe. Les requêtes depuis des adresses IP inconnues sont interdites (IP(s) à nous communiquer pour autorisation). Vérification d'intégrité des données reçues par checksum SHA-256.
Support en ligne	Un support en ligne complet est disponible via un Swagger  accessible sur https://app.mediasflow.fr/MediasflowApi/ <i>Ce Swagger vous permet de récupérer les informations contenues dans ce guide, de vous authentifier, de tester les différentes fonctions de l'API, de récupérer des exemples de codes selon votre langage de développement, etc...</i>

2 FONCTIONS DISPONIBLES

Chaque fonction est accessible via une URI. Le paramétrage se fait par des structures JSON intégrées dans les POST.

Chaque fonction a pour URL d'origine :

<https://app.mediasflow.com/api/v1>

2.1 Demande d'envoi EMAIL (/campaigns/inject)

Cette fonction permet l'envoi d'EMAIL, unitaire ou sur une liste de destinataire. L'API retourne un objet JSON contenant entre autre un numéro d'ID de campagne. Cet ID de campagne est à conserver, il vous sera utile pour récupérer le rapport de campagne, annuler une campagne programmée ou encore injecter une campagne (voir ci-dessous).

Il existe 2 méthodes d'envoi :

1/ envoi **SANS** validation préalable :

La campagne est injectée de suite ou à la date/heure prévue, sans vérification préalable des éléments (paramètre startNow = TRUE).

2/ envoi **AVEC** validation préalable :

La campagne est préparée et vérifiée (paramètre startNow = FALSE). Les éléments de la campagne vous sont retournés via l'API, avec le n° d'ID de campagne. Si les éléments vous conviennent, vous pouvez alors utiliser la fonction /campaigns/inject en indiquant ce n° d'ID dans la valeur "campaignId"

Voici la liste de tous les paramètres possibles pour l'envoi d'une campagne EMAIL :

Paramètre	Type	Obligatoire	Description
campaignId	integer(\$int64)	Oui si la campagne a été déposée au préalable pour préparation via l'API	Cette valeur doit contenir le n° ID de campagne à envoyer, <u>uniquement</u> si celle-ci a été préparée au préalable via l'API avec la valeur startNow à FALSE.
userOwnerID	string	Non	Référence du compte utilisateur. Seul un compte ayant des droits ADMIN peut envoyer une campagne pour le compte d'un utilisateur.
brand	string	Non	Intitulé de la marque pour laquelle la campagne va être envoyée (si le compte est autorisé à envoyer des campagnes pour plusieurs marques). Attention : Toujours utiliser la même valeur brand pour une même marque car les STOP EMAIL seront rattachés au compte de cette marque.

campaignName	string	Non	Nom de la campagne. Si elle n'est pas indiquée, le nom par défaut sera "Date de création Campaign API"
listsId	integer(\$int64)	Non	N° de la liste déposée au préalable via l'API /lists/upload. La liste au format CSV doit contenir au minimum une colonne nommée "EMAIL" ou "MAIL" et peut évidemment comporter d'autres colonnes (nom, prénom, adresse, cp, ville, etc...)
listRecipient	string	Non	Destinataire(s) de l'EMAIL : L'adresse mobile du destinataire peut être inséré dans une balise "EMAIL" (voir exemple ci-dessous). Exemple : "listRecipient": [{ "title": "Mr", "lastName": "Your Lastname", "firstName": "Your Firstname", "email": "destinataire@domaine.com", "mobile": "336xxxx", "phone": "string", "fax": "string", "groupID": 0, "userOwnerID": "string", "extendedKeys": [{ "Adresse": "3 av Victor Hugo", "CP": "92200" }] }]
htmlPart	string	Oui (doit contenir au minimum une des deux balises htmlPart ou textPart ou la balise zipContent)	Message EMAIL à envoyer au format HTML <u>en Base64</u> . Exemple : "htmlPart": " VGHpcyBtdXNOIGJIGFuIEhUTUwgJHtFTUFJTH0" Pour ajouter des champs de personnalisation faisant appel à des champs de votre liste, vous pouvez utiliser la syntaxe BASIC par défaut \${NOM_DU_CHAMP} (voir la valeur renderer).
textPart	String	Oui (doit contenir au minimum une des deux balises htmlPart ou textPart ou la balise zipContent)	Message EMAIL à envoyer au format TEXTE <u>en Base64</u> . Exemple sans personnalisation : "textPart": " TWVvY2kgZGUyY29uZmlybWVvIFRoXMGaXMgYSB0ZXNOIGZyb20g QVBJICR7RU1BSUx9" Pour ajouter des champs de personnalisation faisant appel à des champs de votre liste, vous pouvez utiliser la syntaxe BASIC par défaut \${NOM_DU_CHAMP} (voir la valeur renderer).
zipContent	File	Non	Le fichier .zip peut contenir : - le message à envoyer au format HTML (nommé <u>htmlpart.html</u>) - le message à envoyer au format TEXTE (nommé <u>textpart.txt</u>) - la/les liste(s) d'envoi au format CSV (toutes les listes seront prises en compte automatiquement pour l'envoi de la campagne)

senderEmail	String	Oui	Adresse email émettrice de l'email. Cette adresse doit être paramétrée au préalable sur votre compte. Exemple : noreply@mediasflow.com
fullSenderName	String	Oui	Emetteur de l'email. Exemple : MEDIASFLOW
imageTrackName	String	Non	La balise peut contenir (séparé par un ;) - la/les image(s) à tracker : pour obtenir le tracking d'ouverture dans vos rapports de campagne - le/les lien(s) URL à tracker : : pour obtenir le tracking sur les clics dans vos rapports de campagne Exemple : "imageTrackName":"mon_image.png;https://www.monURL.fr "
scheduleOn	string(\$date-time)	Non	Date & heure de programmation (doit être supérieur d'au moins 5 minutes à l'heure de réception de la commande) Format : "yyyy-MM-dd'T'hh:mm:ss'Z" Exemple : "2021-06-24T09:30:58Z"
renderer	string	Non	Syntaxe utilisée pour les champs de personnalisation. Syntaxe par défaut (BASIC) : \${NOM_DU_CHAMP} Exemple de message avec personnalisation : "textPart": "Bonjour \${PRENOM} \${NOM}" Autres valeurs/syntaxes possibles : - MEDIASFLOW : \${xxxxxxx} - MEDIASFLOW_B : \${[record]xxxxxxxxxxxxx}\${ - HANDLEBARS {{Handlebars}} ==> voir https://jknack.github.io/handlebars.java/helpers.html - EJS <%=xxxxxxx%> ==> voir https://ejs.co - VELOCITY \$xxxxxxx ==> voir https://velocity.apache.org/engine/2.2/vtl-reference.html Remarque : Si cette valeur n'est pas précisée ou si votre compte ne comporte pas un paramètre spécifique, la syntaxe utilisée est BASIC soit une syntaxe sous la forme \${NOM_DU_CHAMP} .
defaultForHeader	key:value	Non	Définition de la valeur utilisée par défaut pour la personnalisation du message si un champs est vide dans votre liste de destinataire ou dans le recipient. Exemple : "defaultForHeader": { "Title": "Mr/Mme", "Ville": "Paris" }

startNow	boolean	Oui	<p>2 valeurs possibles :</p> <ul style="list-style-type: none"> - TRUE : la campagne est injectée de suite ou à la date/heure prévue, sans vérification préalable des éléments. - FALSE : la campagne est préparée et vérifiée. Les éléments de la campagne vous sont retournés via l'API, avec le n° d'ID de campagne. Si les éléments vous conviennent, vous pouvez alors utiliser la fonction /campaigns/inject en indiquant ce n° d'ID dans la valeur "campaignId" <p>La valeur par défaut est FALSE.</p>
checkDuplicate	string	Non	<p>2 valeurs possibles :</p> <ul style="list-style-type: none"> - TRUE : les doublons (sur la base de l'adresse EMAIL du destinataire) seront supprimés. Ainsi, si votre liste contient plusieurs fois la même adresse EMAIL, l'e-mail ne sera envoyé qu'une seule fois. - FALSE : les doublons (sur la base de l'adresse EMAIL du destinataire) ne seront <u>pas</u> supprimés. Ainsi, si votre liste contient plusieurs fois la même adresse EMAIL, l'e-mail sera envoyé autant de fois que présent dans la liste. <p>Par défaut, si cette valeur n'est pas précisée ou si votre compte ne comporte pas un paramètre spécifique, la valeur par défaut sera TRUE.</p>
billingCode	string	Non	Code de facturation que vous retrouverez sur votre facture. Ce code peut être utile pour faciliter le regroupement d'opérations et ainsi faciliter la ventilation de la facture.

2.1.1 Exemple de requête d'envoi simple - **SANS** validation préalable

```
{
  "listRecipient": [
    {
      "email": email@domaine.fr
    }
  ],
  "textPart": " TWVY2kgZGUgY29uZmlybWVyIFRoaxMgaXMgYSB0ZXN0IGZyb20gQVBKICR7RU1BSUx9",
  "htmlPart": " VGhpcyBtdXN0IGJlIGFuIEhUTUwgJHtFTUFJTH0=",
  "startNow": true,
  "sendToSelect": "EMAIL",
  "subject": "test de sujet",
  "senderEmail": "noreply@mediasflow.fr",
  "fullSenderName": "Damien",
  "startNow": true
}
```

2.1.2 Exemple de requête d'envoi simple - **AVEC** validation préalable

```
{
  "listRecipient": [
    {
      "email": email@domaine.fr
    }
  ],
  "textPart": " TWVyY2kgZGUgY29uZmlybWVyIFRoZXN0IGZyb20gQVBJICR7RU1BSUx9",
  "htmlPart": " VGhpcyBtdXN0IGJlIGFuIEhUTUwgJHtFTUFJTH0=",
  "startNow": true,
  "sendToSelect": "EMAIL",
  "subject": "test de sujet",
  "senderEmail": "noreply@mediasflow.fr",
  "fullSenderName": "Damien",
  "startNow": true
}
```

2.1.2.1 Exemple de retour envoyée par l'API EMAIL suite à une requête d'envoi **AVEC** validation préalable

```
{
  "allowUCS2": false,
  "autoREPLACEEMOJI": false,
  "autoREPLACETO GSM7": true,
  "autogenerateTXT": false,
  "autotrackURL": false,
  "campaignName": "2022-03-22 17:57 campaign API",
  "checkDuplicate": true,
  "countBlackList": 0,
  "countBlackListBRAND": 0,
  "countBlackListCUSTOMER": 0,
  "countBlackListMEDIASFLOW": 0,
  "countDuplicate": 0,
  "countDuplicateRecipient": 0,
  "countErrorRendering": 0,
  "countInBlackListRecipient": 0,
  "countInvalidRecipient": 0,
  "countMaxCreditFound": 0,
  "countMaxCrediteOver": 0,
  "countNotAllow": 0,
  "countNotValid": 0,
  "countPosted": 1,
  "countRecipient": 1,
  "countValid": 1,
  "creationDate": "2022-03-22T16:57:15.078Z [UTC]",
  "creditAfterInjection": 1,
  "creditCount": 0,0
  "customerID": "5e1f1fa1392b6a759299b09a",
  "defaultForHeader": {},
}
```



```

"emailTrackings": [
  {
    "isFirst": true,
    "urlLookup": "monimage.png",
    "urlToTrack": "http://mondomaine.com/prevenance/images/monimage.png"
  }
],
"endDate": "2022-03-25T16:57:15.121Z[UTC]",
"fullSenderName": "Damien",
"globalCounter": 1,
"htmlPart": " VGhpcyBtdXN0IGJlIGFuIEhUTUwgJHtFTUFJTH0=",
"id": 1185,
"imageTrackName": "monimage.png",
"injectedDate": "2022-03-22T16:57:15.125Z[UTC]",
"injectionDate": "2022-03-22T16:57:15.119Z[UTC]",
"isInScheduler": false,
"listInCampaign": {
  "campaignId": 0,
  "charsetDetected": "",
  "countDuplicateEmail": 0,
  "countDuplicateMobile": 0,
  "globalInvalidLine": 0,
  "id": -1,
  "invalidNumberEMAIL": 0,
  "invalidNumberMOBILE": 0,
  "lineDuplicateEmail": [],
  "lineDuplicateMobile": [],
  "listItems": [],
  "listsItems": [],
  "maxSizeDataLine": 0,
  "recipients": [
    {
      "email": "dmaitrot@gmail.com",
      "extendedKeys": [
        {
          "keyName": "REF",
          "keyValue": "Ma reference"
        },
        {
          "keyName": "INS1",
          "keyValue": "FNAC LOGISTIQUE"
        },
        {
          "keyName": "INS2",
          "keyValue": "456167022660"
        },
        {
          "keyName": "INS3",

```

```

    "keyValue": "RSM GLOBAL SERVICE"
  },
  {
    "keyName": "INS4",
    "keyValue": "17 RUE DE BAGNOLET"
  },
  {
    "keyName": "INS5",
    "keyValue": "75020"
  },
  {
    "keyName": "INS6",
    "keyValue": "PARIS 20"
  },
  {
    "keyName": "INS7",
    "keyValue": "<a
href='https://mondomaine.com/wssuivicoliscrire/pagesuivi.aspx?ORIGINE=MAILRC&REF=45616702266
0'"
  },
  {
    "keyName": "INS8",
    "keyValue": "mardi 10 ao&ucirc;t"
  },
  {
    "keyName": "INS9",
    "keyValue": "<a href='https://www.google.com/maps?q=48.8568,2.3958'"
  }
],
"groupID": 0,
"id": 0
}
],
"size": 0,
"type": "VIRTUAL",
"validNumberEMAIL": 0,
"validNumberMOBILE": 0
},
"renderer": "MEDIASFLOW_B",
"scheduled": false,
"sendToSelect": "EMAIL",
"sender": "36184",
"senderEmail": "noreply@mediasflow.fr",
"shortCode": "36184",
"status": "INJECTED",
"subject": "test de sujet $$[record]EMAIL$$",
"tinyList": [],
"typeSelect": "MKG",
"updateDate": "2022-03-22T16:57:15.085Z [UTC]",

```

```
"userOwnerID": "5e1f4f4e7e9b504fcfd49eb2",
"vocalTime": 0
}
```

2.1.2.2 Exemple de requête pour injecter une campagne AVEC validation préalable

```
{
  "campaignId": 943,
  "startNow": true
}
```

2.2 Récupération de rapport d'une campagne au format CSV (/campaigns/report/{campaignid})

Cette requête permet de récupérer le rapport d'émission d'une campagne terminée ou en cours au format CSV.

Paramètre	Type	Obligatoire	Description
campaignId	integer(\$int64)	Oui	N° d'ID de campagne qui vous a été communiqué par retour API après la demande de préparation ou d'envoi de la campagne.
status	string	Non	Liste des statuts souhaités, délimitée par un point-virgule [;] Par exemple 16;14
details	string	Non	Liste des champs de personnalisation dans votre fichier d'origine que vous souhaitez récupérer dans le rapport CSV qui sera généré . Délimitée par un point-virgule [;] si plusieurs champs. Par défaut tous les champs sont présents dans le rapport CSV.

2.3 Récupération de rapport d'une campagne au format JSON (/campaigns/reportjson/{campaignid})

Cette requête permet de récupérer le rapport d'émission d'une campagne terminée ou en cours au format JSON.

Paramètre	Type	Obligatoire	Description
campaignId	integer(\$int64)	Oui	N° d'ID de campagne qui vous a été communiqué par retour API après la demande de préparation ou d'envoi de la campagne.
status	string	Non	Liste des statuts souhaités, délimitée par un point-virgule [;] Par exemple 16;14

2.4 Récupération de rapport de campagnes sur une période donnée au format JSON (/campaigns/reportjson/)

Cette requête permet de récupérer le rapport d'émission au format JSON d'une ou plusieurs campagne(s) sur une période donnée, ou sur un critère en particulier (n° de tel, référence client, etc...). La/les campagne(s) doivent être terminées ou en cours.

Attention : La période par défaut est de 7 jours. Si plus de 7 jours, alors un message d'erreur est généré.

Paramètre	Type	Obligatoire	Description
userID	string	Non	ID du compte sur lequel la requête sera faite. Si non précisé et que vous êtes admin du compte, la requête sera faite sur tous User ID du compte.
status	string	Non	Liste des statuts souhaités, délimitée par un point- virgule [;] Par exemple : 110;111;112;113
types	string	Non	Media concerné par la requête. Par exemple : SMS;EMAIL ;VOCAL Si non précisé, la requête renverra tous les médias (SMS+EMAIL+VOCAL)
from	string	Oui	Date à partir de laquelle la requête s'applique. Format : YYYY-MM-DD ou YYYY-MM-DD HH:MM Par exemple : 2022-03-25 → requête effectuée sur tous les envois effectués depuis le 25 mars 2022
to	string	Non	Date jusqu'à laquelle la requête s'applique. Format : YYYY-MM-DD ou YYYY-MM-DD HH:MM Par exemple : 2022-03-25 to 2022-03-28 → requête effectuée sur tous les envois effectués entre le 25 et le 28 mars 2022
search	string	Non	Permet de faire une requête sur un ou plusieurs critère(s) donné(s). FIQL query, [, is or] [; is and] fields.REF==DRAFT;fields.REF==value Par exemple : item==client@domaine.fr → requête effectuée pour tous les envois vers l'email client@domaine.fr fields.prenom==Philippe → requête effectuée pour tous les envois vers les destinataires ayant comme prénom Philippe

2.5 Récupération de rapport de campagnes sur une période donnée au format CSV (/campaigns/report/)

Cette requête permet de récupérer le rapport d'émission au format CSV d'une ou plusieurs campagne(s) sur une période donnée, ou sur un critère en particulier (n° de tel, référence client, etc...). La/les campagne(s) doivent être terminées ou en cours.

Paramètre	Type	Obligatoire	Description
userID	string	Non	ID du compte sur lequel la requête sera faite. Si non précisé et que vous êtes admin du compte, la requête sera faite sur tous User ID du compte.
keyField	string	Non	Liste des champs que vous souhaitez renvoyer dans le rapport séparés par [;] Par exemple : NOM ;PRENOM ;REFCLIENT → le rapport généré au format CSV contiendra ces colonnes.

status	string	Non	Liste des statuts souhaités, délimitée par un point- virgule [;] Par exemple : 110;111;112;113
types	string	Non	Media concerné par la requête. Par exemple : SMS;EMAIL ;VOCAL Si non précisé, la requête renverra tous les médias (SMS+EMAIL+VOCAL)
from	string	Oui	Date à partir de laquelle la requête s'applique. Format : YYYY-MM-DD ou YYYY-MM-DD HH:MM Par exemple : 2022-03-25 → requête effectuée sur tous les envois effectués depuis le 25 mars 2022
to	string	Non	Date jusqu'à laquelle la requête s'applique. Format : YYYY-MM-DD ou YYYY-MM-DD HH:MM Par exemple : 2022-03-25 to 2022-03-28 → requête effectuée sur tous les envois effectués entre le 25 et le 28 mars 2022
search	string	Non	Permet de faire une requête sur un ou plusieurs critère(s) donné(s). FIQL query, [, is or] [; is and] fields.REF==DRAFT;fields.REF==value Par exemple : item==0632545311 → requête effectuée pour tous les envois vers le mobile 0632545311 fields.prenom==Philippe → requête effectuée pour tous les envois vers les destinataires ayant comme prénom Philippe

2.6 Annulation d'une campagne programmée (/campaigns/{campaignid}/cancel)

Cette requête permet d'annuler une campagne programmée. Il est possible d'annuler une campagne jusqu'à 10 secondes avant sa date/heure d'envoi. Une fois annulée, la campagne est remise en statut « DRAFT » (en préparation).

Paramètre	Type	Obligatoire	Description
campaignId	integer(\$int64)	Oui	N° d'ID de campagne qui vous a été communiqué par retour API après la demande de préparation ou d'envoi de la campagne.

2.7 Ajout d'une adresse EMAIL en blacklist (/blacklist)

Cette requête permet d'ajouter un numéro de mobile sur la blacklist du compte. La blacklist est une liste repoussoir des numéros ne souhaitant plus recevoir d'EMAIL.

Paramètre	Type	Obligatoire	Description
item	string	Oui	Adresse EMAIL à ajouter à la blacklist. Exemple : "item": "email@domaine.com"
typeItem	string	Oui	EMAIL
createdDescription	string	Non	Permet d'ajouter une description à l'ajout de l'adresse EMAIL. Exemple : plainte en magasin le 22 juin 2022
brand	string	Non	Permet de préciser pour quelle marque on ajoute l'adresse EMAIL dans la blacklist (valable uniquement pour les comptes envoyant des EMAIL pour plusieurs marques).

2.7.1 Exemple de requête pour ajouter une adresse EMAIL en blacklist

```
{
  "item": "email@domaine.com",
  "typeItem": "EMAIL",
  "createdDescription": " plainte en magasin le 22 juin 2022"
}
```

2.8 Déposer une liste de destinataires (/lists/upload)

Cette requête permet de déposer une liste de destinataires au format CSV. La liste doit contenir au minimum une colonne intitulée "EMAIL", ou "MAIL" contenant le numéro de téléphone mobile du destinataire. Elle peut évidemment comporter d'autres colonnes (nom, prénom, adresse, cp, ville, etc...), mais pour des questions de performances et de respect du RGPD, nous recommandons de ne déposer que les colonnes utiles pour l'envoi et la personnalisation du message.

Paramètre	Type	Obligatoire	Description
userOwnerID	string	Non	Référence du compte utilisateur. Seul un compte ayant des droits ADMIN peut envoyer une campagne pour le compte d'un utilisateur.
description	string	Non	Permet d'ajouter une description en plus du nom de la liste. Exemple : liste clients moins de 6 mois
filename	string	Oui	Nom de la liste
File	file	Oui	Liste de destinataires au format CSV

2.8.1 Exemple de retour envoyée par l'API suite à une requête pour déposer une liste de destinataire

```
{
  "campaignId": 0,
  "charsetDetected": "WINDOWS-1252",
  "computedHeaderSize": {
    "Prénom": {
      "6": 5,
      "7": 2,
      "8": 3,
      "10": 1
    },
    "Adresse": {
      "13": 1,
      "15": 2,
      "17": 1,
      "18": 1,
      "20": 2,
      "21": 1,
      "22": 2,
      "25": 1
    },
    "Prix": {
      "0": 1,
      "2": 1,
      "3": 4,

```

```
"4": 5
},
"Civilité": {
  "0": 1,
  "2": 6,
  "3": 4
},
},
"SMS": {
  "0": 1,
  "9": 4,
  "11": 1,
  "12": 1,
  "13": 1,
  "14": 3
},
},
"Nom": {
  "3": 1,
  "5": 1,
  "6": 6,
  "7": 2,
  "8": 1
},
},
"CP": {
  "5": 11
},
},
"Ville": {
  "5": 6,
  "6": 3,
  "7": 1,
  "20": 1
}
}
},
"countDuplicateEmail": 0,
"countDuplicateMobile": 1,
"creationDate": "2021-06-25T08:46:45.937Z [UTC]",
"customerID": "5e1f1fa1392b6a759299b09a",
"description": "liste_demo",
"emptyKeys": [
  "Prix",
  "Civilité"
],
"filePath":
"/mnt/mediasflow/shared/5e1f1fa1392b6a759299b09a/6009b630d86ae22270c0f3d6/lists//EdKzVweeYahfC
iqt.tmp.csv",
"globalInvalidLine": 0,
"header": [
  "Civilité",
  "Prénom",
  "Nom",
  "Adresse",
  "CP",
  "Ville",
  "Prix",
  "EMAIL"
]
```

```

],
"id": 392,
"invalidCountryCounter": {},
"invalidLineList": [
  10
],
"invalidNumberEMAIL": 0,
"invalidNumberMOBILE": 1,
"lineDuplicateEmail": [],
"lineDuplicateMobile": [
  "11"
],
],
"listName": "liste_demo",
"maxDatasMap": {
  "Prénom": "Jacqueline",
  "Adresse": "12 Avenue des Antiquaires",
  "Prix": "17 €",
  "Civilité": "Mme",
  "SMS": "tel:0624170068",
  "Nom": "Gaillard",
  "CP": "75017",
  "Ville": "Boulogne-Billancourt"
},
"maxSizeDataLine": 66,
"size": 11,
"type": "CSV",
"userOwnerID": "6009b630d86ae22270c0f3d6",
"validCountryCounter": {
  "FR": 10
},
"validNumberEMAIL": 0,
"validNumberMOBILE": 10
}

```

2.8.2 Liste des paramètres dans le retour API suite à une requête pour déposer une liste de destinataire

Paramètre	Type	Description
userOwnerID	string	Complété automatiquement selon le compte utilisateur utilisé pour déposer la liste. Seul un compte ayant des droits ADMIN peut envoyer une liste pour le compte d'un autre utilisateur.
customerID	string	customerID complété automatiquement selon l'ID utilisé pour déposer l'API.
type*	string	2 valeurs possibles : - CSV : utilisé pour déposer une liste au format CSV - VIRTUAL : permet de créer une liste de liste et/ou une liste de "destinataire unique"
filePath	string	Chemin utilisé en interne - ne peut être modifié.
listName	string	Nom de la liste.
emptyKeys	string	Contient les colonnes de votre liste ayant au moins une valeur vide.
invalidLineList	integer(\$int32)	Contient la liste des lignes ayant un destinataire invalide.
globalInvalidLine	integer(\$int32)	Nombre de ligne invalide.

invalidCountryCounter	integer(\$int32)	Nombre de destinataire par pays non autorisé pour l'envoi de SMS depuis votre compte.
validCountryCounter	integer(\$int32)	Nombre de destinataire par pays autorisé pour l'envoi de SMS depuis votre compte.
computedHeaderSize	integer(\$int32)	Pour chaque colonne de votre liste, nombre de destinataires ayant X caractères. Exemple : "Prénom": { "6": 5, "7": 2, "8": 3, "10": 1 }, Dans cet exemple, comprendre que 5 destinataires ont un prénom de 6 caractères, 2 de 7 caractères, etc...
validNumberMOBILE	integer(\$int32)	Nombre de numéro de mobile valide.
validNumberEMAIL	integer(\$int32)	Nombre d'adresse email valide.
invalidNumberEMAIL	integer(\$int32)	Nombre d'adresse email invalide.
invalidNumberMOBILE	integer(\$int32)	Nombre de numéro de mobile invalide.
creationDate	string(\$date-time)	Date de création de la liste.
description		Nom de la liste.
header	integer(\$int32)	Liste des en-têtes de colonnes de votre liste.
size	integer(\$int32)	Nombre de ligne détectée dans votre liste.
maxSizeDataLine	integer(\$int32)	Champs le plus long pour chaque colonne de votre liste.
recipients		Destinataires ajouté (hors liste(s) de destinataire(s))
listsItems		Correspond à une liste de liste (total des listes dans votre campagne)
lineDuplicateMobile	string	Liste des lignes où se trouvent les doublons (n° de téléphone mobile).
countDuplicateMobile	integer(\$int32)	Nombre de doublons (n° de téléphone mobile).
maxDatasMap		Valeur la plus longue pour chaque colonne de votre liste. Exemple : "maxDatasMap": { "Prénom": "Jacqueline", "Adresse": "12 Avenue des Antiquaires", "Prix": "17 €", "Civilité": "Mme", "SMS": "tel:0624170068", "Nom": "Gaillard", "CP": "75017", "Ville": "Boulogne-Billancourt" },
lineDuplicateEmail	string	Liste des lignes où se trouvent les doublons (adresse EMAIL).
countDuplicateEmail	integer(\$int32)	Nombre de doublons (adresse EMAIL).
charsetDetected	string	Charset détecté dans votre liste.
brand	string	Permet d'associer une liste à son compte « brand »
idlistItems	integer(\$int64)	N°ID de la liste

3 CODE ERREURS

STATUS_TEXT	STATUS_CODE	Description
Email valide	101	La structure de l'email est valide
Email invalide	102	La structure de l'email est invalide
Non autorisé	103	L'envoi de l'email n'est pas autorisé
Doublons	104	L'adresse email est en doublon dans votre/vos liste(s) de destinataires
En cours d'envoi	105	L'email est en cours de diffusion
Posté	10	L'email a été déposé sur les routeurs pour envoi
Délivré	21	Le serveur MTA a fait l'envoi vers le domaine sans réception de message d'erreur (code DSN 2.0.0)
Erreur temporaire	107	Une erreur temporaire a été détectée
Erreur temporaire	108	Une erreur temporaire a été détectée
Erreur permanente	109	Une erreur permanente a été détectée
Domaine inconnu	110	Le domaine est inconnu (après le @)
Erreur inconnue	111	Une erreur inconnue a été détectée
Domaine non disponible	112	Le domaine est inconnu (après le @)
Destinataire inconnu	113	Le destinataire est inconnu (avant le @)
Erreur inconnue	114	Une erreur inconnue a été détectée
Erreur de protocole	115	Une erreur protocole a été détectée (erreur serveur, espace disque, protocole)

4 DESCRIPTION DES COLONNES DANS LES RAPPORTS AU FORMAT CSV

Les rapports contiennent les colonnes ci-dessous + toutes les colonnes de votre fichier d'origine.

Intitulé de colonne	Description
REF_LIST	Nom de votre liste de destinataire
ID_LINE	N° de ligne de votre liste de destinataire
EMAIL	N° de mobile du destinataire formaté selon la norme internationale

STATUS_CODE	Code statut (délivré, non délivré, etc...) Voir liste des codes statuts possibles en Annexe 3.
STATUS_TEXT	Libellé statut (délivré, non délivré, etc...) Voir liste des libellé statuts possibles en Annexe 3.
DELIVERYDATE	Date et heure de livraison du message au format JJ/MM/AAAA HH:MM:SS
DSN_CODE	Code DSN retourné par le serveur distant
SMTP_POSTED	Message posté (YES or NO)
NOM_DE_LA_PREMIERE_IMAGE.XXX	Date et heure de l'ouverture de l'image (prise en compte de la première image trouvée dans le HTML - la colonne reprend le nom indiqué dans la balise imageTrackName)
NOM_DE_LA_PREMIERE_IMAGE.XXX_COUNTER	Nombre de fois où l'image a été ouverte
NOM_DE_L'URL	Date et heure du clic sur l'URL (si option de tracking URL activée). Si le message contenait plusieurs URLs, alors le fichier contiendrait plusieurs colonnes (la colonne reprend le nom indiqué dans la balise imageTrackName)
NOM_DE_L'URL_COUNTER	Nombre de fois où l'URL courte a été cliquée
MF_ALLOW	Envoi autorisé par MediasFlow (true or false)
MF_VALID	Adresse valide pour envoi (true or false)