



High
WAY

API d'envoi de SMS

Version du document	Date	Commentaires
2.0	16/12/2022	Début de la documentation
2.0.1	13/02/2023	Finalisation de la version initiale de la documentation V60
2.2.0	10/03/2023	Ajout de l'authentification hmac
2.3.0	07/06/2023	Mis en place du SFTP + sms_operator dans les webhook
2.3.1	05/12/2023	Préconisation sur la découpe d'appels par paquets de 10 000 SMS maximum
2.9.0	03/04/2024	Activation des fonctions de gestion des SMS longs (éléments nb_sms et truncature) Ajout des fonctions suivantes : • Récupération du statut d'un SMS avec un ret_id (en 4.2) • Requête globale sur un push (highpush_stat.php) (en 4.3) • Récupération des réponses au format CSV (en 5.2) • Annulation de campagne (en 6.1) • Vérification de crédit via l'API (en 6.2) Ajout des webservices suivants • /campaign/sms/status [GET] (en 7.3) • /campaign/status [GET] (en 7.4) • /campaign/credit [GET] (en 7.5)
2.10.0	14/05/2024	Ajout des webservices suivants • retrieveFilterStatus (en 4.1.2) • clicks] (en 6.3)
2.13.0	06/09/2024	Mise à jour de la partie 2.1 relative au retour API, qui peut être récupéré à l'identique de la version V50 de notre API (TXT), ou sous un nouveau format JSON

SOMMAIRE

1	<u>PRESENTATION.....</u>	5
1.1	PRESENTATION GENERALE	5
1.2	LEXIQUE	5
1.3	CREATION DE VOTRE COMPTE	5
1.4	NOTION DE ROUTE_TYPE	6
1.5	SECURISATION DE VOTRE COMPTE PAR RESTRICTION IP :	6
2	<u>SOUSSION DE SMS VIA L'API.....</u>	7
2.1	APPEL SIMPLE EN API.....	7
2.1.1	APPEL API	7
2.1.2	RETOUR API.....	7
2.2	DETAILS DE LA STRUCTURE (XML/JSON)	8
2.2.1	STRUCTURE XML/JSON.....	8
2.2.2	ELÉMENT<PUSH>	8
2.2.3	ELÉMENT <MESSAGE>.....	11
2.2.4	ELÉMENT <TEXT>	12
2.2.5	ELEMENT <TO>	13
2.2.6	ELEMENT <PARAM>.....	13
2.2.7	LES TINYURL	13
2.2.8	ELEMENT <BINARY> (GESTION UNICODE)	14
2.3	METHODES D'APPEL.....	15
2.3.1	EXEMPLE XML SIMPLE	15
2.3.2	EXEMPLE JSON SIMPLE	15
2.3.3	ENVOI D'UN XML VIA HTTPS (POST OU GET).....	16
2.3.4	ENVOI D'UN XML EN HTTPS VIA LA FASTXML	16
2.3.5	ENVOI DU XML/JSON VIA (S)FTP	16
2.4	SPECIFICITES DE L'ENVOI A L'INTERNATIONAL.....	18
3	<u>TRAITEMENT ONESHOT.....</u>	19
4	<u>GESTION DES ACCUSES DE RECEPTION.....</u>	20
4.1	GESTION DES ACCUSES DE RECEPTION EN TEMPS REEL	20
4.1.1	UTILISATION STANDARD.....	20
4.1.2	UTILISATION D'UN SCRIPT INTERMEDIAIRE DE FILTRAGE	20
4.2	RECUPERATION DU STATUT D'UN SMS AVEC UN RET_ID	21
4.3	REQUETE GLOBALE SUR UN PUSH (HIGH PUSH_STAT.PHP)	23
4.4	RECEPTION DES WEBHOOKS	25
	SPECIFICATION DU JSON	25
5	<u>GESTION DES REPONSES.....</u>	28
5.1	TRANSMISSION DES REPONSES A LA RET_MO_URL.....	28
5.2	RECUPERATION DES REPONSES AU FORMAT CSV	28
6	<u>AUTRES APPELS A L'API.....</u>	29
6.1	ANNULATION DE CAMPAGNE.....	29
6.2	VERIFICATION DE CREDIT VIA L'API.....	30
6.3	RECUPERATION DES LIENS CLIQUES	31
7	<u>API REST / JSON.....</u>	33
7.1	/CAMPAIGN [POST]	33
7.2	/CAMPAIGN/CANCEL [POST]	34

7.3	/CAMPAIGN/SMS/STATUS [GET]	35
7.4	/CAMPAIGN/STATUS [GET]	35
7.5	/CAMPAIGN/CREDIT [GET]	36
7.6	/CONTACT_LIST/CREATE_LIST[POST]	37
7.7	/CONTACT_LIST/ADD_CONTACTS[POST].....	37
7.8	/CONTACT_LIST/GET_LISTS[GET]	37
7.9	/CONTACT_LIST/DELETE_CONTACTS[POST]	37
7.10	/CONTACT_LIST/DELETE_LIST[GET].....	37
7.11	/CONTACT_LIST/PREPARE_PUSH[POST].....	37
7.12	/CONTACT_LIST/SEND_PUSH[POST].....	37
7.13	/CONTACT_LIST/STAT_PUSH[GET]	37
7.14	/CONTACT_LIST/CANCEL_PUSH[POST].....	37
7.15	/CONTACT_LIST/CONCAT_LISTS[POST].....	37
8	<u>SECURISATION PAR TOKEN POUR LES IP TOURNANTES.....</u>	38
8.1	METHODE CONSEILLEE.....	38
8.1.1	AUTHENTIFICATION LORS DES APPELS API :	38
8.1.2	GENERATION D'UN HMAC :	38
8.2	METHODE DEPRECIEE	39
8.2.1	DESCRIPTION :	39
8.2.2	CONSTRUCTION DU PARAMETRE HMAC.....	40
9	<u>ANNEXE.....</u>	43
9.1	EXEMPLES	43
9.1.1	MESSAGE COMPLEXE EN UNICODE :	43
9.1.2	COMMENT CONSTRUIRE UN XML.....	44
9.1.3	APPEL EN POST AVEC VARIABLE XML.....	44
9.1.4	RECUPERATION SR ET MO	45
9.1.5	CONTENU DES WEBHOOKS	46
9.2	EXEMPLE D'APPEL CURL HMAC :	49
9.3	EXEMPLE D'UN SCRIPT DE CALCUL HMAC EN PHP :	49
9.4	TABLE DE CARACTERES.....	50
9.5	ERREURS XML.....	54
9.6	LISTE DES CODES STATUTS.....	55

1 Présentation

1.1 Présentation générale

HighSMS est une API proposée par HighConnexion qui permet d'envoyer des SMS. Cette API communique avec les opérateurs mobiles à travers notre routeur, pour soumettre les SMS et récupérer les accusés de réception.

Ce document décrit cette API et les différentes utilisations et paramétrages associés.

1.2 Lexique

Les termes suivants sont utilisés à travers la documentation, voici leur définition :

push :

On parle de push quand plusieurs messages, au sein d'un même appel, sont envoyés à l'API. Le push sert de référence à chaque message grâce au push_id.

push_id :

Le push_id est le nombre qui vous est retourné lors de l'appel à l'API /campaign. N'hésitez pas à l'enregistrer, si vous avez besoin d'aide le push_id permet de trouver plus vite les informations nécessaires.

OneShot :

Le OneShot est un push d'un seul message, la particularité est qu'il est envoyé directement et ne prend pas en compte les paramètres start_date et start_time. Si vous ne souhaitez pas avoir ce comportement vous pouvez ajouter le paramètre noone. Dès lors, le message sera traité comme un push.

one_id :

Idem que push_id mais pour le OneShot.

ret_id :

Le ret_id est un paramètre qui vous permettra d'identifier votre message, pensez à mettre un unique id afin de retrouver le message souhaité. Attention, si vous le mettez au niveau du push puis du message puis du to, la prise en compte est la suivante : D'abord le ret_id du to, s'il n'y en a pas, le ret_id du message sinon celui du push.

1.3 Création de votre compte

Pour utiliser HighSMS, 2 comptes sont nécessaires, un compte de facturation et un compte applicatif. Ils sont créés et crédités par HighConnexion.

Compte technique ou compte de facturation

Ce compte permet de gérer un crédit mensuel, de paramétrer les routes qui seront ouvertes pour votre trafic SMS (France / International) et de choisir le type de trafic concerné par les contenus de vos messages (marketing ou notification).

La facturation est basée sur le trafic constaté sur ce compte.

Compte applicatif (HighSMS)

A la création de ce compte, HighConnexion vous communique les paramètres suivants :

- Un login (appelé aussi accountid)
- Un password
- L'url de l'API à appeler

URL livrée ultérieurement par votre responsable de compte High Connexion

Le couple login/password est utilisé comme paramètre dans l'API.

Un compte HighSMS peut être lié à plusieurs comptes de facturation, et inversement.

1.4 Notion de route_type

Ces types de route sont nommées d'après leur qualité. Selon l'usage pour du trafic français, l'équivalence suivante est mise en place :

Bronze => Envoi de type « Marketing Direct », utilisation de short code typé Marketing France

Silver => Envoi de type « Notification », utilisation de short code typé Notification France

Gold => Route gérant l'international

Un compte technique sera créé pour chaque route nécessaire à la gestion de votre trafic.

Un compte applicatif sera donc fonctionnel pour 3 usages différents, suivant le choix du route_type.

1.5 Sécurisation de votre compte par restriction IP :

Si vous possédez des adresses IP fixes, elles devront être communiquées à High Connexion afin de les mettre sur liste blanche et ainsi optimiser la sécurisation de votre trafic.

2 Soumission de SMS via l'API

L'API permet de soumettre des SMS sans passer par un navigateur internet. Cette API peut être sollicitée de différentes manières.

Pour réutiliser les exemples qui vous seront proposés dans l'ensemble de notre documentation, merci de remplacer ses valeurs par celles de votre implémentation.

2.1 Appel simple en API

2.1.1 Appel API

Il vous est possible de faire un test rapide en POST en utilisant les paramètres suivants :

Nom	Obligatoire	Exemple	Commentaire
push	O		
accountid	O	fred	Valeur donnée par HighConnexion
password	O	password	Valeur donnée par HighConnexion
message			
text	O	Hello%20World	
to	O	0619896895	Numéro de téléphone : (+33619896895) ou en format local pour la France (0619896895, et même sans le zéro de tête (619896895)

2.1.2 Retour API

Quand l'API est sollicitée elle crée un conteneur (push) auquel sont rattachés unitairement les SMS à envoyer.

Le numéro qui est renvoyé est l'Id interne de ce conteneur appelé push_id.

Quand le conteneur n'est pas un push mais un OneShot, c'est l'Id de ce conteneur qui est renvoyé (one_id).

Dans des cas très marginaux il pourrait arriver que l'API ne puisse finaliser la création du conteneur et renvoie autre chose qu'un Id.

Dans tous les cas il ne faut pas refaire en boucle un appel à l'API si celle-ci renvoie autre chose qu'un numérique*.

*Le retour de l'API peut être configuré soit en JSON soit en TEXT. Cette configuration doit être définie lors de la création de votre compte.

Réponse JSON :

- Si une erreur s'est produite :

```
{
  "status": "error"
  "data": {
    "error": "ERROR_TYPE"
  }
}
```

- Sinon :

```
{
  "status": "success",
}
```

```

"data":{
  "id": 1871016995
}

```

Réponse TXT :

- Si une erreur s'est produite :

ERROR_TYPE

- Sinon :

1871016995

2.2 Détails de la structure (XML/JSON)

2.2.1 Structure XML/JSON

Le XML / JSON contient un item <push> suivi d'un ou plusieurs items <message>.
Certains attributs peuvent être positionnés dans la partie <push>, ou dans la partie <message>.

Si un paramètre est défini dans les deux, le paramètre de la partie <message> prend le pas sur celle de la partie <push>.

2.2.2 Élément<push>

L'élément <push> contient un ou plusieurs éléments messages <message>.

Attributs de <push>

Nom	Obligatoire	Valeur exemple	Commentaire
accountid	O	Fred	Valeur donnée par HighConnexion
password	O	Xpasswordx	Valeur donnée par HighConnexion
class_type Non existant sur High SMS V60 Développement non prévu	N	1	Dans des cas particuliers vous pouvez être amené à positionner cette variable à 0 ce qui indique « Mode Flash ». Dans les autres cas standard, 1 veut dire « Dans la mémoire du téléphone »
route_type	N	G Default = « your default account route »	Si votre compte vous autorise plusieurs routes, vous pouvez choisir spécifiquement une route. Dans le cas général, vous n'avez qu'une route pour un compte donné et ce paramètre peut être laissé vide. Sinon la valeur possible peut être : 'P', 'G', ou 'S' suivant votre configuration.

start_date	N	20014-12-12 Default = <i>today</i>	Si positionné, le SMS ne sera pas envoyé avant cette date. Pour que le SMS parte immédiatement, ne pas positionner de date.
start_time	N	13 :00 Default = '00 :00'	Dans des cas particuliers, si la start_date est positionnée mais dans le passé, la start_time est utilisée avec la date du jour. Attention : si la start_time n'est PAS positionnée et que start date est positionnée à une valeur > à la date du jour cela est interprété comme start_time à MINUIT (00 :00 :00) Les dates et heures sont en local time (Paris)
stop_time	N		Si stop_time est positionné, aucun SMS n'est transmis à la Gateway d'envoi après cette heure et les éventuels SMS restants, sont envoyés le jour suivant, à partir de start_time. Notez que les SMS transmis à la Gateway avant cette heure peuvent être transmis à l'opérateur après cette heure (peu de décalage, toutefois).
userdata	N	User Data Multiple Sent	C'est la référence qui est renvoyée dans les fichiers de statuts.
name	N	Push Multiple	Nom qui est reporté pour les statistiques et les exports.
sender	N	API_HIGHSMS	Si positionné, remplace le champ 'Expéditeur' dans le SMS. La valeur ne DOIT PAS être 100% numérique et doit être de 11 caractères maximum. Si vide, il sera remplacé par le short code. Si vous positionnez cette valeur, et si le SMS est de type "marketing", vous devez intégrer une mention de type 'STOP au 36105' dans le texte de votre message, pour permettre à l'utilisateur de se désabonner.
ret_id	N	Push_1	S'il est positionné, il est stocké dans push.psh_ext_id. <push>.ret_id est associé avec la campagne (push). C'est un champ différent de <to>.ret_id qui est un id associé avec un SMS unitaire et défini par le client.
ret_url [DEPRECATED] (Cf webhook URL)	N	https://{your_endpoint} .hcnx.eu/receivedDlr	S'il est positionné, quand le statut du SMS change, un appel à cette url est fait. Les spécifications de cet appel sont détaillées en <u>4.1</u> de ce document
ret_mo_url [DEPRECATED] (Cf webhook URL)	N	https://{your_endpoint} .hcnx.eu/receivedMo	Url qui sera appelée en cas de réponse de l'utilisateur (MO) C'est une url de votre plate-forme.
webhook_url	N	https://{your_endpoint} .hcnx.eu /receivedWebhook	URL de réception des notifications de suivi des SMS. Si positionnée, elle est prioritaire sur les champs ret_url et ret_mo_url.

			Nous conseillons donc de ne plus utiliser ces champs ret_url et ret_mo_url si webhook_url est utilisée.
priority Non existant sur High SMS V60 Développement non prévu	N	2	Permet d'augmenter la priorité du push. Ne doit pas être augmenté dans le cadre d'envoi en masse de type marketing.
noone	N	0 ou 1 (default 0)	S'il est positionné à 1, force le push à un seul destinataire à être considéré comme un push et non pas comme un OneShot.
nb_sms	N	1 à 10	Nombre de SMS maxi. Si positionné à une valeur supérieure à 0, le message ne pourra pas être transmis en plus de nb_sms SMS
truncature	N	0 ou 1	Troncature automatique. Si positionné à 1, un message trop long par rapport nb_sms sera tronqué (et envoyé). Sinon le message trop long ne sera pas créé.

2.2.3 Élément <message>

<message> contient un élément <text>, un ou plusieurs éléments <to> et éventuellement des éléments <param>

Attributs de l'élément <message>

Nom	Obligatoire	Valeur exemple	Commentaire
route_type	N	G <i>Default = « your default account route »</i>	Si votre compte vous autorise plusieurs routes, vous pouvez choisir spécifiquement une route. Dans le cas général, vous n'avez qu'une route pour un compte donné et ce paramètre peut être laissé vide. Sinon la valeur possible peut être : 'P', 'G', ou 'S' suivant votre configuration.
class_type Non existant sur High SMS V60 Développement non prévu	N	1	Dans des cas particuliers vous pouvez être amené à positionner cette variable à 0 ce qui indique « Mode Flash ». Dans les autres cas standard, 1 veut dire « Dans la mémoire du téléphone »
start_date	N	20014-12-12 <i>default=today</i>	Si positionné, le SMS ne sera pas envoyé avant cette date.
start_time	N	13:00 <i>default='00 :00'</i>	Dans des cas particuliers, si la start_date est positionnée mais dans le passé, la start_time est utilisée avec la date du jour. Attention : si la start_time n'est PAS positionnée et que start date est positionnée à une valeur > à la date du jour cela est interprété comme start_time à MINUIT (00 :00 :00) Les dates et heures sont en local time (Paris)
stop_time	N		Si stop_time est positionné, aucun SMS n'est transmis à la Gateway d'envoi après cette heure et les éventuels SMS restants, sont envoyés le jour suivant, à partir de de start_time. Notez que les SMS transmis à la Gateway avant cette heure peuvent être transmis à l'opérateur après cette heure (peu de décalage en général, toutefois).
sender	N	API_HIGHSMS	Si positionné, remplace le champ 'Expéditeur' dans le SMS. La valeur ne DOIT PAS commencer par un chiffre et doit être de 11 caractères maximum (1FOE est interdit, FOE12345678 est autorisé). Si vous positionnez cette valeur, et si le SMS est de type "marketing", vous devez intégrer une mention de type 'STOP au 36105' dans

			le texte de votre message, pour permettre à l'utilisateur de se désabonner.
ret_url [DEPRECATED] (Cf webhook URL)	N	https://{your_endpoint}.hcnx.eu/receivedDlr	S'il est positionné, quand le statut du SMS change, un appel à cette url est fait. Les spécifications de cet appel sont détaillées en 4.1 de ce document
ret_mo_url [DEPRECATED] (Cf webhook URL)	N	https://{your_endpoint}.hcnx.eu/receivedMo	Url qui sera appelée en cas de réponse de l'utilisateur (MO) C'est une url de votre plate-forme.
webhook_url	N	https://{your_endpoint}.hcnx.eu/receivedWebhook	URL de réception des notifications de suivi des SMS.
Ref_model Non existant sur High SMS V60 Développement non prévu	N	Y default='N'	S'il est positionné à 'Y', indique que <text> n'est pas le texte d'un message, mais le nom d'un modèle qui a été créé via l'interface Web. Voir les exemples spécifiques.
Priority Non existant sur High SMS V60 Développement non prévu	N	2	Permet d'augmenter la priorité du push. Ne doit pas être augmenté dans le cadre d'envoi en masse de type marketing.
userdata	N	myClientName	Permet de positionner des informations supplémentaires qui seront restituées par SMSPUSHER ou d'autres rapports
target_url	N	http://www.mysite.com/confirmation_parisien?nom=%NOM%&nom_ville=%NOM_VILLE%&access=phone	Permet de positionner une variable %URL% dans le texte du message qui sera remplacée par une tiny url (url courte) qui fera la redirection vers cette url. La tiny url qui remplacera cette url est composée de 17 caractères.
target_url_salt Non existant sur High SMS V60 Développement non prévu	N	4	Si positionné, on rajoute un nombre aléatoire à la tiny_url, pour éviter qu'un scan séquentiel des tiny_url potentielles donne régulièrement un résultat valide.

2.2.4 Élément <text>

L'élément <text> est utilisé pour le texte du message.

<text> n'a pas d'attribut, juste une valeur.

Dans le cas général, c'est simplement le texte du SMS.

Exemples de valeurs de <text> :

- Le texte du message

<text>Hello world</text>

- le texte du message avec une variable

<text>Hello world, %FIRST_NAME%</text>

Comme dans l'exemple ci-dessus, <text> peut contenir des variables. Vous pouvez utiliser n'importe quelle chaîne de caractères comme nom de variable. Cependant nous vous conseillons d'utiliser %VAR_NAME% pour la variable nommée 'var_name' (voir [2.2.6 Élément<param>](#)).

Attention : La variable %URL% est réservée pour les tiny url (voir [2.2.7 Les tiny url](#)).

2.2.5 Element <to>

<to> est utilisé pour les informations du destinataire. C'est un élément de <message>.

<to> peut contenir des éléments <param>, utilisés pour personnaliser le texte du message.

Pour la valeur du numéro du téléphone (mobile), vous pouvez utiliser soit le format international (+33619896895) soit le format local pour les numéros français (0619896895, éventuellement sans le 0 de tête, comme 619896895).

Pour les envois internationaux, validez avec votre contact commercial que le pays cible est bien configuré pour votre compte.

Attributs de l'élément <to>

Nom	Obligatoire	Valeur exemple	Commentaire
value	O	+33619896895	Numéro du destinataire
ret_id	N	231467	Vous pouvez utiliser ce champ pour associer un Id unique pour chaque SMS. Le ret_id est associé à un SMS (différent de <push>.ret_id) , est stocké en database et renvoyé dans les informations de statut.

Il y a une limite maximale du nombre de valeurs intégrées dans l'éléments <to>. **Le nombre de valeurs (donc de numéros destinataires) est de 10 000 maximum.**

Cela permet d'optimiser la prise en compte des appels, et donc à terme votre trafic global.

2.2.6 Element <param>

<param> est utilisé pour donner les paramètres du message. C'est un élément de <to> ou de <message>

<param> doit avoir 2 attributs, "var" pour le nom de la variable , "value" pour sa valeur

Attributs de l'élément <param>

Nom	Obligatoire	Valeur exemple	Commentaire
var	O	%NAME%	Nous vous conseillons d'utiliser % pour encadrer le nom de la variable et d'utiliser des majuscules
value	O	Peps	

Attention : La variable %URL% est réservée pour les tiny url (voir [2.2.7 Les tiny url](#)).

2.2.7 Les tinyurl

Si le texte du message ("text") contient %URL% et que le message comporte un attribut "target_url" le mécanisme de génération de tiny urls se met en œuvre :

- une tiny url, unique pour chaque destinataire de message, est générée
- %URL% est remplacée par cette tiny_url
- la target_url est fabriquée (si elle contient des variables, les contenus sont remplacés)
- le redirect entre la tiny_url (celle qui est dans le SMS reçu par le destinataire et sur laquelle il va cliquer) et la target_url avec les variables personnalisées est préparée.

La tiny url qui remplacera cette url est composée de 17 caractères. (Exemple : zv.hcnx.fr/urolhp)

Par exemple :

```
{
  "push": {
    "accountid": "fred",
    "password": "xpasswordx",
    "noone": 1,
    "message": [{
      "target_url":
"https://www.mysite.com/confirmation?origine=%VAR_MESSAGE%&nom=%NOM%&nom_ville=
%NOM_VILLE%&access=phone ",
      "text": "Hello, %NOM% from %VAR_MESSAGE% (%NOM_VILLE%). Click on %URL%",
      "to": [{
        "value": "+33619896895",
        "param": [
          {
            "var": "%PRENOM%",
            "value": "Fred"
          },
          {
            "var": "%NOM_VILLE%",
            "value": "Lyon"
          }
        ]
      }
    ]
  }
}]
}
```

Le texte du SMS sera :

« Hello Fred, from Province (LYON). Click on b.hcnx.fr/3Fs5lpD »

Le clic sur la tiny url (b.hcnx.fr/3Fs5lpD) va fera un redirect vers :

http://www.mysite.com/confirmation?origine=Province&nom=Fred&nom_ville=LYON&access=phone

Notez qu'il est indispensable que le caractère '&' dans la partie value (l'url cible) soit échappé en '&'.
Si ce n'est pas le cas vous aurez une ERROR_PARSING_XML avec un « ; expected »

2.2.8 Element <binary> (gestion Unicode)

La gestion de l'Unicode permet aux utilisateurs d'envoyer des messages avec des caractères non disponibles dans l'UTF8. Tel que des caractères russes, japonais, arabe ou encore les smileys.

L'élément <binary> est utilisé pour le contenu du message **à la place de l'élément <text>** dans le cas où le message doit être envoyé en Unicode.

Les octets dans le champ <binary> ne peuvent être codés que sous 2 formes (en Unicode un caractère est codé sur 2 octets) :

- La forme **hh** :

Le caractère Q est par exemple codé 0051

- La forme **%hh** :

Le caractère Q est par exemple codé %00%51

Exemples de valeurs de <binary> :

```
<binary>%00%42%00%6f%00%6e%00%6a%00%6f%00%75%00%72%00%20%00%65%00%6e%00%20%00%6a%00%61%00%70%00%6f%00%6e%00%61%00%69%00%73%00%20%00%73%00%27%00%e9%00%63%00%72%00%69%00%74%00%20%00%3a%00%20%3053%3093%306b%3061%306f</binary>
```

```
<binary>0042006f006e006a006f0075007200200065006e0020006a00610070006f006e00610069007300200073002700e900630072006900740020003a002030533093306b3061306f</binary>
```

Pour afficher sur le téléphone :

Bonjour en japonais s'écrit : こんにちは

Précisions :

La gestion de l'Unicode prend en compte l'élément <param> à condition que var et value soit **codés sous la même forme** que le contenu dans <binary>.

Quant aux paramètres à changer dans l'url, ceux-ci doivent impérativement être codés en UTF8 afin de pouvoir être remplacés correctement.

Vous pouvez consulter en [annexe 9.1.1](#) l'exemple d'un message complexe en unicode.

2.3 Méthodes d'appel

2 méthodes d'appel possible : XML et JSON

Ce qui est envoyé en XML, peut être soumis avec les mêmes valeurs de paramètres en JSON.

Cette structure peut être transmise dans une requête HTTP(s), ou transmise dans un fichier, et envoyée par (S)FTP.

2.3.1 Exemple XML simple

```
< ?xml version= "1.0 " encoding= "iso-8859-1 " ?>
<push
accountid= "fred "
password= "xpasswordx "
>
<message>
<text>Helllo world !</text>
<to>0619896895</to>

</message>
</push>
```

2.3.2 Exemple JSON simple

```
{
```

```

"push": {
  "accountid": "fred ",
  "password": "xpasswordx",
  "message": {
    "text": "Hello world !",
    "to": [
      {
        "value": "0619896895"
      }
    ]
  }
}

```

2.3.3 Envoi d'un XML via https (POST ou GET)

Vous pouvez envoyer les différents paramètres, structurés dans un XML via https en le positionnant en POST ou GET. **[envoi en GET deprecated]**

Le contenu du XML doit être mis dans une variable appelée **xml**.

Ci-dessous l'exemple avec un passage en GET

```

https://{your_endpoint}.hcnx.eu/api?xml=<push accountid="fred"
password="xpasswordx"><message><text>Hello world
xml!</text><to>0619896895</to></message></push>

```

Vous pouvez consulter en [annexe 9.1.3](#) l'exemple complet en POST.

2.3.4 Envoi d'un XML en https via la FastXML

Non existant sur High Push V60

Développement non prévu

2.3.5 Envoi du XML/JSON via (S)FTP

Pour utiliser cette fonctionnalité, il est nécessaire de créer un dépôt SFTP dédié.

Merci de vous rapprocher de nos équipes si cette partie vous intéresse.

La création de campagne via dépôt SFTP permet de traiter des campagnes volumineuses.

Une fois la campagne envoyée un fichier de statistique qui évoluera dans le temps sera disponible.

Il est possible de créer une campagne en utilisant deux formats différents : XML et JSON.

Il y a une limite maximale sur la taille du fichier que peut traiter le dépôt SFTP pour chacun des formats.

Pour le XML la taille du fichier ne doit pas dépasser 100 mega (environ 1 000 000 de destinataire).

Pour le JSON la taille du fichier ne doit pas dépasser 40 mega (environ 400 000 destinataire).

Au-delà de cette limite la campagne ne sera pas traitée.

Nota : S'il y a un seul numéro de téléphone dans le fichier XML/JSON, et qu'il n'y a pas de start_date et start_time positionnées, le fichier sera traité comme « oneshot ». Le SMS sera envoyé instantanément.

Pour être traité, l'extension du fichier source doit être le type du format de la donnée.
Un fichier XML devra avoir l'extension **.xml**, un fichier JSON devra avoir l'extension **.json**.

Une fois le dépôt du fichier fini (et pas avant) et afin de lancer le traitement, il faudra déposer un fichier **.ack** portant **le même nom** (extension non incluse) que le fichier source.

La présence de ce fichier indiquera que le transmission du fichier source est terminé et que le traitement peut commencer.

Le fichier .ack doit donc être déposé après le fichier source.

Exemple :

32016 déc. 17:26 PushCampaign.xml

0 16 déc. 17:26 PushCampaign.ack

Ou

32016 déc. 17:26 PushCampaign.json

0 16 déc. 17:26 PushCampaign.ack

Lorsque le traitement du fichier commence, un fichier vide .run est créé.
Ce fichier indique que la campagne est en cours de traitement.

A ce moment vous pouvez donc voir dans votre répertoire ftp :

32016 déc. 17:26 PushCampaign.json

0 16 déc. 17:26 PushCampaign.ack

0 16 déc. 17:26 PushCampaign-230427092806.run

A savoir, lors du traitement de la campagne plusieurs type de fichier différents vont être créés : .run, .ret, .ok et .nok.

Afin de garantir une unicité sur le nom des fichiers créés, le nommage de ses fichiers se basera sur le nom du fichier source ainsi que la date du traitement du fichier.

Ainsi, le nom des fichiers créés seront: le nom du fichier source, d'un tiret, de la date de traitement (année mois jour heure minute seconde) puis de l'extension.

Exemple :

Si un fichier PushCampaign.xml est déposé le 27 avril 2023 à 15H30 et que le traitement a commencé, le nom du fichier créé sera : **PushCampaign-230427153000.run**.

Une fois le traitement terminé, un fichier .ok ou .nok est créé indiquant que le traitement de la campagne est terminé.

Un fichier .ok indiquera que le traitement de la campagne s'est bien passé, à l'inverse, un fichier .nok indiquera qu'il y a eu un problème lors du traitement.

Dans le fichier .nok, on y retrouvera l'erreur rencontré lors du traitement du fichier.

Une fois le traitement terminé, les fichiers sources (.xml/json et .ack) seront déplacés dans un répertoire d'archive non accessible.

A ce moment vous pouvez donc voir dans votre répertoire ftp :

0 16 déc. 17:27 PushCampaign-230427092806.ok

Une fois que les SMS sont envoyés (soumis à l'opérateur) un fichier .ret est créé.
Il reprend les informations de la gestion statistique des accusés de communication.
Ce fichier est mis à jour toutes les heures pendant 72 heures (délai au bout duquel les SMS non acheminés sont considérés comme expirés).
Les premiers SMS apparaissent avec le statut 'SENT'

PushCampaign-230427092806.ret :

```
"2014-12-16 17:27:02";"2";"";"1";"";"1"
```

```
"2014-12-16 17:27:02";"+33619896895";"SENT";"2014-12-16 17:31:04";"1866810063";"";"Hello world  
Multiple";"";"User
```

```
Data Multiple Sent"
```

```
"2014-12-16 17:27:02";"+33699999999";"SENT";"2014-12-16 17:31:04";"1866810063";"";"Hello world  
Multiple";" ERROR_INVOP";"User Data Multiple Sent"
```

Dans cet exemple 1866810063 est ce que nous appelons le "push_id"

Le .ret est mis à jour et notre PushCampaign.ret va ressembler à cela au bout d'un certain temps :

```
"2014-12-16 17:27:02";"2";"";"1";"";"1"
```

```
"2014-12-16 17:27:02";"+33619896895";"RECEIVED";"2014-12-16 17:31:04";"1866810063";"";"Hello world  
Multiple";"";"User Data Multiple Sent"
```

```
"2014-12-16 17:27:02";"+33699999999";"ERROR";"2014-12-16 17:31:04";"1866810063";"";"Hello world  
Multiple";" ERROR_INVOP";"User Data Multiple Sent"
```

2.4 Spécificités de l'envoi à l'international

Si les paramètres de votre compte SMSPusher le permettent, vous pouvez envoyer des SMS dans n'importe quel pays.

Dans ce cas :

- Vous devez impérativement indiquer les numéros au format international (champ **to**) avec le + suivi du code indicatif du pays (+33619896895 par exemple pour un numéro français)
- Il est fortement conseillé de positionner un émetteur (champ **sender**). En fonction du pays pour lequel vous voulez envoyer des SMS, merci de contacter votre CSM High Connexion pour vous conseiller car certains pays nécessitent de déclarer les Sender. Certains pays ont leurs propres spécificités.

3 Traitement OneShot

Dans le cas d'un push, un enregistrement est créé en base de données et chaque SMS envoyé est rattaché à ce push.

Vous pouvez alors disposer de statistiques consolidées sur cette campagne.

Quand le push est créé en base de données, un processus séparé scanne la base jusqu'à ce que la start_date et la start_time déclenchent l'envoi. Si cette date de déclenchement n'est pas positionnée, le push part au premier scan (scan toutes les minutes).

Le push est alors envoyé en totalité.

Dans le cas d'un envoi OneShot, un seul enregistrement est créé et le SMS est envoyé directement.

Le mode OneShot est automatiquement utilisé si dans la soumission à l'API il n'y a qu'un SMS à envoyer immédiatement (que ce soit en API GET, en XML via GET, POST ou fichier).

Un paramètre spécifique (noone) permet de désactiver ce fonctionnement automatique.

La meilleure façon de faire de l'envoi OneShot est d'utiliser le web service suivant : **FastApi**, autre solution proposée par High Connexion et plus adaptée.

N'hésitez pas à demander la documentation spécifique FastApi à votre contact privilégié High Connexion.

Dans le cas de l'appel de cette FastApi, l'appel est fait directement à la Gateway (SMSPusher), sans aucun enregistrement créé dans le backoffice (HighSMS). Ce mode de communication est donc réservé à certains emplois, avec de fortes fréquences d'appel.

4 Gestion des accusés de réception

L'API permet de couvrir deux besoins fonctionnels très différents :

- Envoi de masse et suivi temps réel, information transmise en temps réel par nos serveurs sur vos serveurs
- Envoi ponctuel avec suivi statistique à l'initiative de votre serveur ou manuel

4.1 Gestion des accusés de réception en temps réel

4.1.1 Utilisation standard

Dans ce cas, notre serveur appelle via le protocole http(s), votre serveur à chaque changement d'état du SMS, c'est la ret_url qui est appelée.

[ret_url DEPRECATED (utilisation de la fonction webhook_url conseillée)]

La ret_url est donc appelée en http GET avec les paramètres suivants :

- push_id (transmis en réponse à l'appel)
- ret_id (positionné par vos soins dans l'appel initial)
- to (numéro de téléphone du destinataire au format international)
- statut (un statut ou un code d'erreur en cas d'ERROR)
- text (le texte du message envoyé)

La date de réception du SR (**attention en UTC**) peut aussi être envoyée si on positionne __DATE__ dans la ret_url.

De même l'opérateur peut être récupéré si __OPERATOR__ figure dans la ret_url.

L'opérateur est renvoyé sous la forme MCC MNC (Exemple 20801 pour Orange)

Si ret_url n'est pas positionnée, aucun appel n'est fait.

Si le webhook_url est positionné, la ret_url et ret_mo_url seront ignorées. Voir la partie [4.4 Réception des webhook](#) pour plus de détails.

Dans tous le cas, le push_id, correspondant à l'identifiant unique de votre push, est renvoyé.

Le texte du SMS est renvoyé dans le champ <text>

L'ensemble des valeurs du statut pouvant se retrouver dans la ret_url est précisée en [annexe 9.6](#) de ce document.

4.1.2 Utilisation d'un script intermédiaire de filtrage

Il peut arriver qu'on ne désire pas recevoir tous les statuts, mais uniquement ceux qui nécessitent un traitement dans le CRM.

Par exemple, ne recevoir que les « hard bounces » du type NPAI ou INVOP, correspondant aux numéros à supprimer de la liste d'envoi.

Un script répondant à ce besoin est : RetrieveFilterStatus.php

En remplaçant :

```
ret_url="http://{your_webhook_url}"
```

ou

```
webhook_url="http://{your_webhook_url}"
```

par

ret_url=https://{your_endpoint}.hcnx.eu/RetrieveFilterStatus.php?target_ret_url=http://{your_ret_url}&status=11,13

OU

webhook_url=https://{your_endpoint}.hcnx.eu/RetrieveFilterStatus.php?target_ret_url=http://{your_webhook_url}&status=11,13

Ne seront soumis que les statuts ERROR_NPAI et ERROR_INVOP à http://{your_ret_url}

Cela permet d'éviter de saturer le serveur de traitement des SR lors des envois avec des statuts RECEIVED qui peuvent être sans intérêt.

4.2 Récupération du statut d'un SMS avec un ret_id

Pour obtenir le statut d'un SMS, vous devez avoir envoyé un XML ou un JSON qui a un ret_id pour chaque attribut <to>.

Exemple XML:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<push
  ....accountid="fred"
  ....password="xpasswordx"
  ....name="PushName"
  ....userdata="User Data Multiple Sent"
  ....ret_id="12345"
  ....sender="API_6"
  ....ret_url="http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php"
>
  ....<message>
  ....  ....<text>Hello world : CheckStatus</text>
  ....  ....<to ret_id="123">0619896895</to>
  ....  ....<to ret_id="124">0699999999</to>
  ....</message>
</push>
```

Même exemple en JSON:

```
{
  ...."push":{
    ...."accountid":"fred",
    ...."password":"xpasswordx",
    ...."name":"PushName",
    ...."ret_id":"12345",
    ...."userdata":"User Data Multiple Sent",
    ...."ret_url":"http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php",
    ...."sender":"API_6",
    ...."message":[{
      ...."text":"Hello world : CheckStatus",
      ...."to":[
        ....{
          ...."ret_id":"123",
          ...."value":"0619896895"
        },
        ....{
          ...."ret_id":"124",
          ...."value":"0699999999"
        }
      ]
    }]
  }
}
```

Vous pouvez alors appeler **ws_status_for_ret_id**.

Les paramètres à fournir sont l'accountid, le password et le ret_id du <to>.

Dans notre exemple:

`https://{your_endpoint}.hcnx.eu/ws_status_for_ret_id?accountid=fred&password=xpasswordx&ret_id=123`

La réponse est un CSV avec l'entête suivante : Vers, Statut, Status description, Opérateur, MVNO, Date.

Exemple:

```
"Vers";"Statut";"Status description";"Opérateur";"MVNO";"Date"
"+33619896895";"2";"Erreur envoi de message";"";"2015-12-04 10:04:52"
```

Nota : Vous pouvez aussi faire cette requête en POST.

4.3 Requête globale sur un push (highpush_stat.php)

Si vous ne voulez pas avoir votre propre base de données temps réel mais voulez ponctuellement connaître le statut des SMS de votre campagne, vous pouvez appeler notre serveur en http.

Le script à appeler est **highpush_stat.php**

Pour ne pas charger les serveurs NE PAS APPELER ce script trop fréquemment, et ne pas mettre en place de système d'appel automatique et récurrent.

Nom	Obligatoire	Valeur exemple	Commentaire
accountid	O	fred	Valeur donnée par HighConnexion.
password	O	xpasswordx	Valeur donnée par HighConnexion.
push_id	O si ret_id non défini	1871011352	Valeur interne du push_id, renvoyé quand le XML/JSON est fourni par une requête HTTP.
ret_id	O si push_id non défini	12345	Valeur fournie dans le XML/JSON, associée au push, pas au message individuel

Pour utiliser ce script vous devez fournir votre accountid, votre password et soit le push_id soit le ret_id .

Exemple XML :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<push
  ....accountid="fred"
  ....password="xpasswordx"
  ....name="PushName"
  ....userdata="User Data · Multiple Sent"
  ....ret_id="12345"
  ....sender="API_6"
  ....ret_url="http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php"
>
  ....<message>
  ....  ....<text>Hello world : CheckStatus</text>
  ....  ....<to>0619896895</to>
  ....  ....<to>0699999999</to>
  ....</message>
</push>
```

Même exemple JSON :

```
{
  ...."push":{
    ...."accountid":"fred",
    ...."password":"xpasswordx",
    ...."name":"PushName",
    ...."ret_id":"12345",
    ...."userdata":"User Data · Multiple Sent",
    ...."ret_url": "http://highsms-rec.hcnx.eu/peps_sample/RetrieveStatus.php",
    ...."sender": "API_6",
    ...."message":[
      ...."text":"Hello world : CheckStatus",
      ...."to":[
        ....{
          ...."value":"0619896895"
        },
        ....{
          ...."value":"0699999999"
        }
      ]
    ]
  }
}
```

On peut requêter soit en utilisant le push_id :

https://{your_endpoint}.hcnx.eu/highpush_stat.php?accountid=fred&password=xpasswordx&push_id=1871011352

Soit en utilisant le ret_id

https://{your_endpoint}.hcnx.eu/highpush_stat.php?accountid=fred&password=xpasswordx&ret_id=12345

Nota :

Dans le cas de plusieurs pushes avec le même ret_id, seules les données associées au dernier ret_id sont renvoyées.

La requête http renvoie un fichier CSV.

Les lignes en gras italique ne sont pas envoyées

<i>Send date</i>	<i>Total</i>	<i>Net_error</i>	<i>Waiting</i>	<i>NPAI</i>	<i>expired</i>	<i>Other error</i>	<i>received</i>		
08/01/15 11:38	2			1			1		
<i>Send date</i>	<i>Phone Number</i>	<i>Status</i>	<i>Status Update</i>	<i>push_id</i>		<i>text</i>	<i>Error Status</i>	<i>userdata</i>	<i>Parent_id</i>
08/01/15 11:38	33619896895	RECEIVED	08/01/15 12:22	1871011352		Hello world : CheckStatus		User Data Multiple Sent	User Data Multiple Sent
08/01/15 11:38	33699999999	ERROR	08/01/15 12:22	1871011352		Hello world : CheckStatus	INVOP	User Data Multiple Sent	User Data Multiple Sent

La première ligne est un résumé :

- PushDate
- Nombre total de Messages (qui peut être différent du nombre de SMS)
- Total de net_error (ERROR_NETWORK => 14)
- Total waiting (pas de réponse Error ou Received)
- Total npai (ERROR_NPAI=>11)
- Total expired (ERROR_EXPIRED=>12)
- Total autres erreurs
- Total Received

Il y a ensuite une ligne par SMS (nombre qui peut être différent du nombre de message)

- PushDate
- Numéro téléphone (to)
- Status
 - SENT
 - RECEIVED
 - ERROR
- Dernière mise à jour de statut (la mise à jour du statut RECEIVED peut arriver bien après la réception du SMS)
- Vide
- Texte du message
- Statut d'erreur (si status = Error)
 - NPAI
 - EXPIRED

- INVOP
- NETWORK
- BLOCKED
- TOO_LONG
- UNKNOWN
- Userdata (field userdata of the xml)
- Id du SMS Parent dans le cas de la deuxième partie d'un SMS long

4.4 Réception des webhooks

Les webhooks sont des appels api qui vous notifient avec une structure json uniforme pour les évènements suivants :

- Un clic
- L'envoi du sms (SENT)
- Le statut de retour du sms (RECEIVED ou ERROR)
- La réponse de l'utilisateur

Chaque type de webhook est optionnel, si vous souhaitez recevoir seulement les sms reçus et pas le reste, une configuration peut être mise en place pour sélectionner le type de webhook à recevoir.

Il n'est pas possible de demander que le webhook d'un sms envoyé soient transmis sur une url et que le webhook d'un sms reçu soit transmis sur une url différente. Tous les webhooks ne peuvent être envoyés qu'à une seule adresse. L'uniformisation de ceux-ci ont pour but de simplifier le traitement de votre côté.

Si la ret_url et/ou la ret_mo_url est positionnée, le webhook_url sera ignoré et donc par conséquent il ne sera pas possible de bénéficier de quelconque webhook.

Spécification du JSON

champ dans le JSON du Webhook	Type	Explication du champ
event	Object	Objet évènement
sms	Object	Objet SMS
url	Object	Objet URL

Objet "event"

champ dans le JSON du Webhook	Type	Explication du champ
event_status	String	Evènement
event_http_id	Integer	Id de l'appel HTTP (Id HCNX)
event_date	Datetime	Date de l'évènement

Liste des events :

Event_status = [MT, DLR, MO, CLICK]

Objet "sms"

champ dans le JSON du Webhook	Type	Explication du champ
sms_hcnx_id	String	Id du SMS (Id HCNX (mt_id))
sms_client_id (ref doc : ret_id du <to>)	String	Id du SMS (Id du client)
sms_client_campaign_id (ref doc : ret_id du push)	String	Id de la campagne du Client
sms_msisdn	String	MSISDN
sms_status	String	Statut du SMS
sms_message	String	Message compilé
sms_sender	String	Sender Id compilé
sms_response	String	Message en réponse
sms_user_data (ref doc : userdata)	String	Donnée facultative fournie par le client (exemple : client du client)
sms_counter	int	Nombre de sms contenu dans le message
sms_operator	String	MccMcn permettant d'identifier l'opérateur qui a envoyé le message.

Liste des status (MT)

```
sms_status = [  
    CREATED,                // MT pris en compte chez HCNX  
    SENT,                   // MT envoyé à l'opérateur  
    ERROR_INVOP,            // MT: Opérateur invalide pour ce Msisdn  
    ERROR_NOCREDIT,         // Limite de crédit dépassé pour le compte applicatif  
    ERROR_BLACKLIST,        // MT : Le numéro est en blacklist sur la plate-forme HCNX  
    ERROR_BADPASSWORD,      // accountid et password incorrect  
    ERROR_UNKNOWN           // Autres erreurs MT  
]
```

Liste des status (DLR)

```
sms_status = [  
    RECEIVED,               // MT reçu par le terminal  
    ERROR_NPAI,             // MT : Msisdn n'habite pas à l'adresse indiqué  
    ERROR_EXPIRED,          // MT: expiré, en général au bout de 24h  
    ERROR_INVOP,            // MT: Opérateur invalide pour ce Msisdn  
    ERROR_NETWORK,          // MT en erreur sur le réseau opérateur  
    ERROR_CREDIT,           // Le compte utilisé n'a plus de crédit  
    ERROR_NPAIPORTED,       // MT : Msisdn n'habite pas à l'adresse indiqué et le numéro a été porté  
    ERROR_UNKNOWN           // Autres erreurs MT  
]
```

Liste des status pour event MO

```
sms_status = [  
INCOMING,           // MO reçu de la part du terminal  
]
```

Liste des status pour event CLICK

```
sms_status = [  
RECEIVED,           // MT reçu par le terminal  
]
```

Objet "url"

champ dans le JSON du Webhook	Type	Explication du champ
short_url	String	url courte
target_url	String	url longue

Vous trouverez en [annexe 9.1.5](#) un exemple pour chaque type d'événement et notification possible.

5 Gestion des réponses

5.1 Transmission des réponses à la ret_mo_url

[ret_mo_url DEPRECATED (utilisation de la fonction webhook_url conseillée)]

Si l'utilisateur répond au SMS (envoie un MO en réponse à un MT), le contenu du SMS est renvoyé à l'url de gestion des MO, soit celle explicitement positionnée dans l'appel, soit celle configurée par défaut dans le compte technique de l'utilisateur.

Dans notre exemple c'est <http://myaddress.com/TreatMO.php>

Cette url est appelée en http au format GET avec tous les paramètres positionnés par la plate-forme.

Ci-dessous les paramètres principalement utilisés :

Nom	Valeur exemple	Commentaire
FROM	+33619896895	Numéro de téléphone du destinataire, au format international (+336...) équivalent du champ <to> de la requête initiale.
TO	36105	Short code utilisé par le destinataire pour répondre
MESSAGE	STOP%20Thanks%20but%20stop	Texte du message, url escaped.
RECEPTION_DATE	2014-12-12T13:00:05	Date de réception, format ISO.
TO_OP_ID	20810	Id de l'opérateur. <i>Principales références opérateurs français :</i> 20801: Orange 20810 : SFR 20820: Bouygues 20815 : Free 20823: Virgin Mobile 20826: NRJ MOBILE 20827 :Coriolis
ORIG_ID	99134567	push_id (celui retourné à la requête HTTP initiale)
RET_ID	123456	ret_id tel que positionné dans la requête http initiale
ORIG_MESSAGE	Hello%20World	Texte du SMS MT initial

Vous pouvez voir en détail un exemple, en [annexe 9.1.4 Récupération SR et MO](#). En dehors des paramètres listés ci-dessus, le restant des données représente peu d'intérêt.

5.2 Récupération des réponses au format CSV

Pour récupérer les réponses (dont les STOP) correspondant à un envoi via l'API, il est nécessaire de faire un appel au webservice « answer » avec le ret_id du push.

Dans notre exemple:

https://{your_endpoint}.hcnx.eu/answer.php?accountid=fred&password=xpasswordx&ret_id=12345&start_date=2024-03-10

Ce Webservice est prévu pour être utilisé après une campagne pour en récupérer les STOPS.
 Il peut être pénalisant si on cherche sur une plage de dates trop importante.
Aussi dans tous les cas, la recherche ne s'effectue que sur les 40 derniers jours.

Paramètres

Nom	Obligatoire	Valeur exemple	Commentaire
accountid	O	fred	Valeur donnée par HighConnexion (Compte HighSMS).
password	O	xpasswordx	Valeur donnée par HighConnexion (Compte HighSMS).
ret_id	O	12345	ret_id que vous avez fourni dans le XML/JSON, associé au push, pas au message individuel
start_date	O	2024-03-10	Date de début du filtre
end_date		2024-03-10	Date de fin du filtre

Le fichier généré est au format CSV et contient les colonnes suivantes :

Nom	Valeur exemple	Commentaire
Date et heure	10/03/2024 12:16:41	Date et heure de la réponse (JJ/MM/AAAA HH :MM :SS)
From	33619896895	Numéro de l'émetteur
To	36105	Numéro du destinataire
Stop	1	1 si STOP sinon 0
Message	STOP	Texte de la réponse

Il est à noter que ce Webservice peut également être utilisé pour récupérer les réponses correspondant à une campagne créée avec l'interface. Dans ce cas il faut donner pour le ret_id de la requête le nom de la campagne.

6 Autres appels à l'API

6.1 Annulation de campagne

Annuler une campagne n'est possible que si elle a été programmée avec un ext_id et si elle n'a pas encore commencé.

Dès lors, il est possible d'annuler une campagne en appelant l'URL suivant dont les paramètres sont détaillés ci-après :

https://{your_endpoint}.hcnx.eu/api?accountid=fred&password=xpasswordx&ext_id=12345&action=cancel_campaign

Paramètres

Nom	Obligatoire	Valeur exemple	Commentaire
accountid	O	fred	Valeur donnée par HighConnexion (Compte HighSMS).

password	O	xpasswordx	Valeur donnée par HighConnexion (Compte HighSMS).
action	O	cancel_campaign	
ext_id	O	12345	ret_id que vous avez fourni dans le XML/JSON, associé au push, pas au message individuel

En réponse, vous recevrez le nombre de campagne qui ont été annulées* ou un message d'erreur.

```
{
  "status": "success",
  "data": {
    "nb_canceled": 0
  }
}
```

Dans cet exemple, comme la campagne a été déjà envoyée, aucune campagne n'est annulée.

*Il peut y avoir plusieurs campagnes annulées si vous avez scindé une même campagne en plusieurs petites pour des raisons de performance.

6.2 Vérification de crédit via l'API

Vous pouvez vérifier votre crédit avant de soumettre un push. Pour cela il faut utiliser la requête **credits**. Les seuls paramètres à fournir sont l'accountid et le password

Par exemple :

https://{your_endpoint}.hcnx.eu/credits?accountid=fred&password=xpasswordx

La réponse est un XML qui indique, pour chaque route du compte, le crédit autorisé.

Le crédit peut être global, mensuel ou annuel.

- -1 indique qu'il n'y a pas de limitation de crédit.
- 0 indique "plus de crédit"

Ci-dessous la réponse à l'appel pour le compte de fred qui a une route « Gold », « Silver », « Bronze » et « Premium » avec 1514 crédits.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<credits>
  <route>
    <type>MarketingDirect</type>
    <credit>1514</credit>
    <credit_month>1514</credit_month>
    <credit_year>1514</credit_year>
  </route>
  <route>
    <type>Notification</type>
    <credit>1514</credit>
    <credit_month>1514</credit_month>
    <credit_year>1514</credit_year>
  </route>
  <route>
    <type>International</type>
    <credit>1514</credit>
    <credit_month>1514</credit_month>
    <credit_year>1514</credit_year>
  </route>
  <route>
    <type>DomTom</type>
    <credit>1514</credit>
    <credit_month>1514</credit_month>
    <credit_year>1514</credit_year>
  </route>
</credits>
```

6.3 Récupération des liens cliqués

Il est possible pour un push dont les MT comportent une URL courte générée par nos soins, d'obtenir le détail des clics : destinataire ayant cliqué et nombre de clics par destinataire. Pour cela il faut faire appel au Webservice « clicks » avec le ret_id du push. Dans notre exemple avec https://{your_endpoint}.hcnx.eu comme root url :

https://{your_endpoint}.hcnx.eu/clicks.php?accountid=fred&password=xpasswordx&ret_id=12345&start_date=2024-05-01

Il peut être pénalisant si on cherche sur une plage de dates trop importante.

Aussi dans tous les cas, la recherche ne s'effectue que sur les 40 derniers jours.

Paramètres

Nom	Obligatoire	Valeur exemple	Commentaire
accountid	O	fred	Valeur donnée par HighConnexion (Compte HighSMS).
password	O	xpasswordx	Valeur donnée par HighConnexion (Compte HighSMS).
ret_id		12345	ret_id que vous avez fourni dans le XML, associé au push, pas au message individuel
start_date	O	2024-05-01	Date de début du filtre
end_date		2024-05-01	Date de fin du filtre

Le fichier généré est au format CSV et contient les colonnes suivantes :

Nom	Valeur exemple	Commentaire
Date et heure	2024-05-01 12:16:41	Date et heure du MT (AAAA-MM-JJ HH :MM :SS)
From	33619896895	Numéro de l'émetteur
Message	Recevez votre coupon Mes Vacances en cliquant	Texte du MT

	sur http://b.HCNX.eu/5f	
Tiny_url	http://b.HCNX.eu/5f	Tiny URL présente dans le message du MT
Target_url	http://mesvacances.fr/Reduction.html	URL à laquelle renvoie la Tiny_url
Date du dernier clic	2024-05-01 14 :15 :31	Date et heure du dernier clic effectué par l'émetteur
Nombre de clics	3	Nombre de fois que l'émetteur a cliqué sur ce lien
Userdata	Compte soleil	Identifiant permettant de positionner des informations supplémentaires qui seront restituées par SMSPUSHER ou d'autres rapports

Il est à noter que ce Web Service peut également être utilisé pour connaître les clics effectués en réponse à un OneShot. Il faudra alors veiller à fournir le ret_id positionné dans l'élément <to> lors de l'envoi du MT ou, si celui-ci a été omis, le ret_id positionné dans l'élément <push>.

De même ce Web Service peut être utilisé pour connaître les clics effectués en réponse à une campagne créée avec l'interface web Highway (anciennement HighPush ou HighSMS). Le ret_id fourni dans les paramètres devra alors correspondre au nom de la campagne.

7 API Rest / JSON

HighSMS implémente l'ensemble des fonctionnalités dans une API Rest / JSON.
Vous trouverez donc ici l'ensemble des Web Services constituant cette API Rest.

7.1 /campaign [POST]

Description :

Permet l'envoi de campagnes SMS.

Requête :

- Méthode : POST
- Données POST au format JSON :

```
{
  "push": {
    "accountid": "fred",
    "password": "xpasswordx",
    "start_date": "2017-03-08",
    "start_time": "17:25",
    "userdata": "MONDOR_Cardio",
    "sender": "MONDOR",
    "ret_id": "Push_Mondor_324",
    "ret_url": "",
    "priority": 2,
    "ret_mo_url": "",
    "noone": false,
    "nb_sms": 1,
    "truncature": 1,
    "message": [
      {
        "text": "Bonjour %NAME%, votre rendez-vous du %DATE% est confirmé",
        "to": [
          {
            "value": "+33623456789",
            "ret_id": "Mess4563",
            "param": [
              {
                "var": "%NAME%",
                "value": "DUPONT"
              },
              {
                "var": "%DATE%",
                "value": "Jeudi 9 mars 2017, 8h30"
              }
            ]
          },
          {
            "value": "+33623475352",
            "ret_id": "Mess4564",
            "param": [
              {
                "var": "%NAME%",
                "value": "DURAND"
              },
              {
                "var": "%DATE%",
                "value": "Jeudi 9 mars 2017, 8h30"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ],
  },
]
}

```

Réponse :

- Si une erreur s'est produite :

```

{
  "status": "error",
  "data": {
    "error": "ERROR_TYPE"
  }
}

```

- Sinon :

```

{
  "status": "success",
  "data": {
    "id": 1871016995
  }
}

```

Nota sur les caractères spéciaux

Des caractères spéciaux doivent être mis en syntaxe HTML :

"text": "Notification \ "<\ "& " pour Notification "<".&

7.2 /campaign/cancel [POST]

Description :

Permet d'annuler une campagne qui n'a pas encore démarré.

Requête :

- Méthode : POST
- Données POST au format JSON :

```

{
  "accountid": "fred",
  "password": "xpasswordx",
  "action": "cancel_campaign",
  "ext_id": 12345
}

```

Le **ext_id** (identifiant du push côté client) est le **ret_id** associé au push fourni dans l'appel au /campaign

Réponse :

Retourne le nombre de campagnes qui ont été annulées* ou un message d'erreur.

```

{
  "status": "success",
  "data": {
    "nb_canceled": 0
  }
}

```

*Il peut y avoir plusieurs campagnes annulées si vous avez scindé une même campagne en plusieurs petites pour des raisons de performance.

La campagne est annulable jusqu'à 10mn avant le lancement de la campagne

Attention :

La campagne n'est pas vraiment annulée, son statut est forcé à `_CAMPAIGN_STATUS_FINISHED` ce qui fait que plus aucun SMS associé à cette campagne n'est soumis à l'opérateur.

7.3 /campaign/sms/status [GET]

Description :

Permet d'obtenir les informations concernant l'envoi d'un SMS.

Requête :

- Méthode : GET
- Paramètres GET à transmettre :

```
"accountid": "fred"
"password": "xpasswordx"
"ret_id": 12345
```

Réponse :

```
{
  "status": "success",
  "data": {
    "sms":
    [
      {
        "number": "+33619896895",
        "status": 6,
        "status_desc": "RECEIVED",
        "operator": "Orange France",
        "MVNO": "20801",
        "date": "2015-12-04 10:04:52",
      },
    ],
  }
}
```

7.4 /campaign/status [GET]

Description :

Permet d'obtenir le statut d'une campagne et de ses SMS.

Requête :

- Méthode : GET
- Paramètres GET à transmettre :

```
"accountid": "fred"
"password": "xpasswordx"
/* Renseigner l'un des deux obligatoirement */
"push_id": 1871011352
"ret_id": 12345
```

Réponse :

```
{
  "status": "success",
  "data": {
    "push_id": 1871016995,
    "ret_id": "Push_Mondor_324",
    "sender": "MONDOR",
    "userdata": "MONDOR_Cardio",
    "summary": {
      "send_date": "2017-03-08 17:25",
      "total_message": 2,
      "total_sms": 2,
      "net_error": 0,
      "waiting": 0,
      "npai": 1,
      "expired": 0,
      "other": 0,
      "received": 1,
    },
    "message": [
      {
        "send_date": "2017-03-08 17:25",
        "number": "33623456789",
        "status": "RECEIVED",
        "status_update": "2017-03-08 17:27",
        "ret_id": "Mess4563",
        "text": " Bonjour Dupont, votre rendez-vous du Jeudi 9 mars 2017, 8h30est confirmé ",
        "nb_sms": 1,
        "error_status": null,
      },
      {
        "send_date": "2017-03-08 17:25",
        "number": "33699999999",
        "status": "ERROR",
        "status_update": "2017-03-08 17:36",
        "ret_id": "Mess4564",
        "text": " Bonjour Durand, votre rendez-vous du Jeudi 9 mars 2017, 8h30est confirmé ",
        "nb_sms": 1,
        "error_status": "NPAI",
      }
    ]
  }
}
```

Si on passe le push_id (identifiant interne de HighConnexion) il est forcément unique.

Si on passe le ret_id (positionné par le client) et qu'il y a plusieurs pushes associés à ce ret_id on ne fournit que les infos du dernier.

Les valeurs de "status"

- RECEIVED
- ERROR
- SENT

Les valeurs de "error_status" (si "status" = ERROR)

- NPAI
- EXPIRED
- INVOP
- NETWORK
- BLOCKED
- TOO_LONG
- UNKNOWN

7.5 /campaign/credit [GET]

Description :

Permet d'obtenir les informations concernant votre crédit.

Requête :

- Méthode : GET
- Paramètres GET à transmettre :

```
"accountid": "fred"  
"password": "xpassword"
```

Réponse :

```
{  
  "status": "success",  
  "data": {  
    "route": [  
      {  
        "type": "Notification",  
        "credits": -1,  
        "credits_month": -1,  
        "credits_year": -1  
      }  
    ]  
  }  
}
```

Nota : -1 indique qu'il n'y a pas de limite.

Toutes les webservices suivants ne sont et ne seront pas développées sur High SMS V60

~~7.6 /contact_list/create_list[POST]~~

~~7.7 /contact_list/add_contacts[POST]~~

~~7.8 /contact_list/get_lists[GET]~~

~~7.9 /contact_list/delete_contacts[POST]~~

~~7.10 /contact_list/delete_list[GET]~~

~~7.11 /contact_list/prepare_push[POST]~~

~~7.12 /contact_list/send_push[POST]~~

~~7.13 /contact_list/stat_push[GET]~~

~~7.14 /contact_list/cancel_push[POST]~~

~~7.15 /contact_list/concat_lists[POST]~~



8 Sécurisation par Token pour les IP tournantes

8.1 Méthode conseillée

8.1.1 Authentification lors des appels API :

Description :

Afin de vérifier votre identité, un token d'authentification vous sera remis si vous souhaitez utiliser ce moyen d'authentification. Merci de joindre celui-ci dans tous les appels API que vous ferez par la suite.

Requête :

Inclure la clé **X-AUTH-TOKEN** et sa valeur dans le header de votre appel.

Exemple :

KEY	VALUE
X-AUTH-TOKEN	azerty

Voir annexe [9.2 Exemple d'appel cURL HMAC](#) pour un exemple d'appel complet.

8.1.2 Génération d'un HMAC :

Description :

Un HMAC est un code d'authentification calculé en utilisant une fonction de hachage cryptographique en combinaison avec une clé secrète (hmac_key). Il peut être utilisé pour vérifier simultanément l'intégrité des données et l'authenticité d'un message.

Pour permettre ceci un HMAC sera nécessaire à la fin de chaque appel. Celui-ci est calculé en fonction du contenu de l'appel + de la clé HMAC qui vous sera transmise lors de la création d'un compte.

Requête :

Prendre tous les champs de l'appel, calculer le HMAC avec la clé HMAC et intégrer le résultat dans le champ hmac du body.

Exemple :

```
1  {
2    "push":{
3      "accountid":"HIGH_TEST",
4      "message":[{
5        "text":"Message de test",
6        "to":[
7          {
8            "value":"+33629519546"
9          }
10       ]
11     }],
12     "hmac":"c02362f913ba3b25f94040c583fc5e01"
13   }
14 }
```

Voir annexe [9.3 Exemple d'un script de calcul Hmac en php](#) pour le script de calcul.

8.2 Méthode dépréciée

8.2.1 Description :

Un système de sécurisation par IP sur l'API a été mis en place.

Dans le cas où il ne peut être appliqué (utilisation d'IP tournantes), un système de sécurisation par token peut lui être substitué (on intervient en complément) pour les appels API (méthode GET|POST avec le format XML et JSON).

Ce token sera renseigné par le paramètre « hmac ».

Paramétrage API

Je n'ai pas d'ip fixe ☐

Limiter par adresse IP

API bloquée si absence de Token (HMAC) ☐

HMAC KEY

API KEY

Algorithme de hashage (HMAC)

Gestion des SR (si utilisation API)

Il vous est possible de choisir un algorithme de hashage pour ce HMAC (actuellement, sha256 ou md5). Par défaut, l'algorithme est sha256.

Paramétrage API

Je n'ai pas d'ip fixe ☐

Limiter par adresse IP

API bloquée si absence de Token (HMAC) ☐

HMAC KEY

API KEY

Algorithme de hashage (HMAC)

Gestion des SR (si utilisation API)

La présence de ce paramètre « hmac » lors d'un appel API peut être rendu obligatoire ou laissé optionnel le temps des tests. Cette option est également activable dans la configuration du compte applicatif.

Paramétrage API

Je n'ai pas d'ip fixe
☐

Limiter par adresse IP

Limiter par adresse IP

API bloquée si absence de Token (HMAC)
☐

HMAC KEY

hmac

API KEY

token

Algorithme de hashage (HMAC)

sha256

Gestion des SR (si utilisation API)

SR envoyés en différé

8.2.2 Construction du paramètre HMAC

Pour construire ce token « hmac », il faut

- Préparer la chaîne à encoder en concaténant certaines valeurs des paramètres utilisés lors de l'appel de l'API et la clé API (APIKEY) qui vous a été fournie avec un séparateur '&' entre chaque valeur.
- Utiliser un système hachage avec comme algorithme « SHA256 » et comme clé de codage l'APIKEY pour calculer le hmac
- Rajouter un paramètre nommé « hmac » lors de l'appel de l'API en lui donnant la valeur calculée.

Un exemple pour générer le paramètre « hmac » est donné ci-dessous.

Paramètres pris en compte pour préparer la chaîne à coder :

- accountid
- password
- ret_id
- push_id
- ret_url
- ret_mo_url

Exemple construction hmac pour un appel à highpush_stat.php

```

<?php
$DEBUG=0 ;

// Les paramètres de l'appel
$url = https://{your_endpoint}.hcnx.eu/highpush_stat.php';
$ACCOUNTID="HIGH_PUSH";
$PASSWORD= "xpasswordx";
$PUSH_ID="1871011352";

// Ce qui est propre au hmac
$API_KEY="azertyuiop"; // Clé fournie par HighConnexion
$WithHmac=1; // On utilise le hmac

// Préparation des paramètres

```



```

$body_req = array(
"accountid" => $ACCOUNTID,
"password" => $PASSWORD,
"push_id" => $PUSH_ID,
);
if ($WithHmac)
{
    $hmac=CalculHmac($API_KEY,$body_req);
    $body_req['hmac']=$hmac;
}
// Fabrication de l'url finale (si Get)
if ($method == 'get') {
    $url .= '?'.http_build_query($body_req);
}

// la fonction de calcul du hmac
function CalculHmac($apiKey,$params)
{
    global $DEBUG;
    $keyHmac = array('accountid', 'password','ret_id' , 'push_id', 'ret_url','ret_mo_url');

    $encode = "";
    foreach($params as $key => $val){
        $trouve=in_array($key, $keyHmac);
        if (in_array($key, $keyHmac)){
            $encode .= $val."&";
        }
    }
    $encode .= $apiKey;
    $hmacEncoderSHA256 = hash_hmac("sha256",$encode,$apiKey);
    $DEBUG && print "hash_hmac(sha256,$encode,$apiKey)=$hmacEncoderSHA256\n";
    return $hmacEncoderSHA256;
}

```

Ce qui donne l'url :

https://{your_endpoint}.hcnx.eu/highpush_stat.php?accountid=HIGH_PUSH&password=xpasswordx&push_id=1871011352&hmac=d181b7b30b7fd44724e132318e462a6f094472287de1dc88290ae4d4db59d362

Détail des paramètres :

Nom	Obligatoire	Valeur exemple	Commentaire
accountid	O	HIGH_PUSH	Valeur donnée par HighConnexion.
password	O	xpasswordx	Valeur donnée par HighConnexion.
push_id	O si ret_id non défini	1871011352	Valeur interne du push_id, renvoyé quand le XML/JSON est fourni par une requête HTTP.
Hmac		d181b7b30b7fd44724e132318e462a6f094472287de1dc88290ae4d4db59d362	Hachage généré en 'sha256' sur la concaténation de HIGH_PUSH&xpasswordx &1871011352&azertyuiop avec la clé azertyuiop pour le codage. (On suppose que azertyuiop est la valeur de APIKEY)

Exemple pour un appel POST (même principe de calcul du hmac)

Description :

Permet l'envoi de campagnes SMS.

Requête :

- Méthode : POST

- Données POST au format JSON :
- HMAC : Hachage généré en 'sha256' avec la concaténation de HIGH_TEST&xpasswordx&Push_Mondor_324&https://{your_endpoint}.hcnx.eu RetrieveStatus.php&azertyuiop

```
{
  "push": {
    "accountid": "HIGH_TEST",
    "password": "xpasswordx",
    "start_date": "2017-03-08",
    "start_time": "17:25",
    "userdata": "MONDOR_Cardio",
    "sender": "MONDOR",
    "ret_id": "Push_Mondor_324",
    "ret_url": "https://highpush-v50.hcnx.eu/RetrieveStatus.php",
    "hmac": "4a2840cead3c1b701c6a97be8d8c38276719954d8b45aac6282e102a87498",
    "priority": 2,
    "ret_mo_url": "",
    "noone": false,
    "nb_sms": 1,
    "truncature": 1,
    "message": [
      {
        "text": "Bonjour %NAME%, votre rendez-vous du %DATE% est confirmé",
        "to": [
          {
            "value": "+33623456789",
            "ret_id": "Mess4563",
            "param": {
              {
                "var": "%NAME%",
                "value": "DUPONT"
              },
              {
                "var": "%DATE%",
                "value": "Jeudi 9 mars 2017, 8h30"
              }
            ]
          },
          {
            "value": "+33623475352",
            "ret_id": "Mess4564",
            "param": {
              {
                "var": "%NAME%",
                "value": "DURAND"
              },
              {
                "var": "%DATE%",
                "value": "Jeudi 9 mars 2017, 8h30"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Réponse :

- Si une erreur s'est produite :

```
{
  "status": "error"
  "detail": "ERROR_TYPE"
}
```

- Si une erreur de token s'est produite :

```
{
  "status": "error"
  "detail": "Error has occurred : Hmac non valide"
}
```

- Sinon :

```
{
  "status": "success",
  "data": {
    "id": 1871016995
  }
}
```

9 Annexe

9.1 Exemples

9.1.1 *Message complexe en Unicode :*

Nous souhaitons envoyer ce message personnalisé :

こんにちは %PRENOM% 私たちのサイトを見に来てください: %URL%

Qui veut dire en français : Bonjour %PRENOM% viens voir notre site : %URL%

```
{
  "push":{
    "accountid":"account",
    "password":"password",
    "message":{
      "target_url" : "www.highconnexion.com?ville=%VILLE%",

      "binary":"30533093306b3061306f00200025005000520045004e004f004d0025002079c1305f3061306e30b
      530a430c83092898b306b67653066304f306030553044003a0020002500550052004c0025",

      "to":{
        "ret_id":"ret_id1",
        "value":"+336666666666",
        "param": [{
          "var":"0025005000520045004e004f004d0025", // %PRENOM% en unicode
          "value": "00500061007300630061006c" // Pascal en unicode
        },
        {
          "var": "%VILLE%",
          "value": "Lyon"
        }
      ]
    }
  }
}
```

Ce qui donnera le message suivant :

こんにちは Pascal 私たちのサイトを見に来てください: zi.hcnx.fr/TiWmRI

Et l'url sur laquelle redirige la tiny url :

<https://www.highconnexion.com/?ville=Lyon>

9.1.2 Comment construire un XML

Si l'exemple montré précédemment est très simple, le format XML a été conçu pour piloter des campagnes importantes de SMS avec de nombreux paramètres.

Ci-dessous un exemple plus complet :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<push
accountid="fred"
password="xpasswordx"
userdata="User Data Multiple Sent"
ret_id="Push_1"
sender="API_HIGHSMS"
>
<message>
<text>Hello world Multiple</text>
<to>0619896895</to>
<to>0699999999</to>
</message>
</push>
```

9.1.3 Appel en POST avec variable XML

```
<?php
#Les paramètres de mon SMS
$accountid='fred';
$password='xpasswordx';
$urlapi= https://{your_endpoint}.hcnx.eu/api';
$ret_url='http://myaddress.com/RetrieveStatus.php';
$ret_mo_url='http://myaddress.com/sample/TreatMO.php';

$to='0619896895';

# Si Sender non positionné ou à vide le ShortCode (36105 dans notre exemple est utilisé)
# $sender='MYTEST';

#Si je positionne cette identifiant je le récupère dans les SR et les MO
$suryident='MYTEST_123';
$text='Faites votre réponse au 36105';

#La fabrication du XML

$xml= <<<EOXML
<?xml version="1.0" encoding="utf-8"?>
<push
accountid="$accountid"
password="$password"
ret_url="$ret_url"
ret_mo_url="$ret_mo_url"
sender="$sender"
ret_id="$suryident"
```

```
>
<message>
<text>$text</text>
<to>$to
</to>
</message>
</push>
```

9.1.4 Récupération SR et MO

[DEPRECATED]

Exemple de Code source de RetrieveStatus.php

```
# cat RetrieveStatus.php
<?php
$req_dump = print_r($_REQUEST, TRUE);
$fp = fopen('/tmp/request.log', 'a');
fwrite($fp, "RetrieveStatus:\n");
fwrite($fp, $req_dump);
fclose($fp);
```

Exemple de Code source de TreatMO.php

```
# cat TreatMO.php
<?php
$req_dump = print_r($_REQUEST, TRUE);
$fp = fopen('/tmp/request.log', 'a');
fwrite($fp, "TreatMO:\n");
fwrite($fp, $req_dump);
fclose($fp);
```

Exemple après exécution

Après réponse "Thanks mais été pourri" au 36105

```
# cat /tmp/request.log
RetrieveStatus:
Array
(
    [push_id] => 568300865
    [status] => 6
    [to] => +33619896895
    [ret_id] =>
        [text] => Il a fait beau cet été
)

TreatMO:
Array
(
    [FROM] => +33619896895
    [ID] => 568442694
    [TO] => 36105
    [MESSAGE] => Thanks mais été pourri
    [VALIDITY_DATE] => 2014-12-19T17:48:16
```

```

[GET_STATUS] => 0
[CLIENT] => HIGHPUSH
[CLASS_TYPE] => 0
[RECEPTION_DATE] => 2014-12-18T17:48:16
[TO_OP_ID] => 20810
[INITIAL_OP_ID] => 20810
[STATUS] => POSTING_18327_4807
[EMAIL] =>
[BINARY] => 0
[PARAM] => 33609003663
[USER_DATA] => HIGHPUSH
[USER_DATA_2] => Thanks mais été pourri
[BULK_ID] => 0
[MO_ID] => 0
[APPLICATION_ID] => 0
[ACCOUNT_ID] => 39
[GW_MESSAGE_ID] => 0
[READ_STATUS] => 0
[TARIFF] => 0
[REQUEST_ID] => 33609003663
[TAC] => (null)
[REASON] => 2014-12-18 18:48:15
[FORMAT] =>
[MVNO] =>
[DATE] => 2014-12-18T17:48:16
[ORIG_ID] => 568300865
[ORIG_MESSAGE] => Il a fait beau cet été.
[ORIG_DATE] => 2014-12-18 17:48:16
[RET_ID] =>
)

```

9.1.5 Contenu des webhooks

MT pris en compte par HCNX suite à une soumission par l'API

```

{
  "event" : {
    "event_status" : "MT",
    "event_http_id" : 56,
    "event_date" : "2022-04-10 10:30:22"
  },
  "sms" : {
    "sms_hcnx_id" : "1234567890", // MT_ID
    "sms_client_id" : "123e4567-e89b-12d3-a456-426655440000",
    "sms_client_campaign_id" : "ABC123",
    "sms_msisdn" : "+33635393605",
    "sms_status" : "CREATED",
    "sms_message" : "Bonjour Thibault, inscris toi ici : http://a.hcnx.eu/ABC123",
    "sms_sender" : "HCNX",
    "sms_response" : "",
    "sms_user_data" : "client_customer_id",
    "sms_counter" : 1,
    "sms_operator" : ""
  }
}

```

```
}
```

MT soumis à l'opérateur par HCNX

```
{
  "event": {
    "event_status": "MT",
    "event_http_id": 56,
    "event_date": "2022-04-10 10:30:23"
  },
  "sms": {
    "sms_hcnx_id": "123456789", // MT_ID
    "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
    "sms_client_campaign_id": "ABC123",
    "sms_msisdn": "+33635393605",
    "sms_status": "SENT", // SENT ou ERROR : {raison_erreur}
    "sms_message": "Bonjour Thibault, inscris toi ici : http://a.hcnx.eu/ABC123",
    "sms_sender": "HCNX",
    "sms_response": "",
    "sms_user_data": "client_customer_id",
    "sms_counter": 1,
    "sms_operator": "20801"
  }
}
```

DLR reçu de l'opérateur

```
{
  "event": {
    "event_status": "DLR",
    "event_http_id": 56,
    "event_date": "2022-04-10 10:30:25"
  },
  "sms": {
    "sms_hcnx_id": "123456789",
    "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
    "sms_client_campaign_id": "ABC123",
    "sms_msisdn": "+33635393605",
    "sms_status": "RECEIVED", // RECEIVED ou ERROR : {raison_erreur}
    "sms_message": "Bonjour Thibault, inscris toi ici : http://a.hcnx.eu/ABC123",
    "sms_sender": "HCNX",
    "sms_response": "",
    "sms_user_data": "client_customer_id",
    "sms_counter": 1,
    "sms_operator": "20801"
  }
}
```

MO reçu

```
{
  "event": {
    "event_status": "MO",
    "event_http_id": 56,
    "event_date": "2022-04-10 10:30:30"
  },
}
```

```

"sms": {
  "sms_hcnx_id": "123456789",
  "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
  "sms_client_campaign_id": "ABC123",
  "sms_msisdn": "+33635393605",
  "sms_status": "RECEIVED",
  "sms_message": "Bonjour Thibault, inscris toi ici : http://a.hcnx.eu/ABC123",
  "sms_sender": "HCNX",
  "sms_response": "STOP",
  // Message MO
  "sms_user_data": "client_customer_id",
  "sms_counter": 1,
  "sms_operator": "20801"
}
}
Click
{
  "event": {
    "event_status": "CLICK",
    "event_http_id": 56,
    "event_date": "2022-04-10 10:30:35"
  },
  "sms": {
    "sms_hcnx_id": "123456789",
    "sms_client_id": "123e4567-e89b-12d3-a456-426655440000",
    "sms_client_campaign_id": "ABC123",
    "sms_msisdn": "+33635393605",
    "sms_status": "RECEIVED",
    "sms_message": "Bonjour Thibault, inscris toi ici : http://a.hcnx.eu/ABC123",
    "sms_sender": "HCNX",
    "sms_response": "",
    "sms_user_data": "client_customer_id",
    "sms_counter": 1,
    "sms_operator": ""
  },
  "url": {
    "short_url": "http://a.hcnx.eu/ABC123",
    // Short Url compilée
    "target_url": "http://google.com?firstname=Thibault&id=123e4567-e89b-12d3-a456-426655440000"
    // Target Url compilée
  }
}

```


9.2 Exemple d'appel cURL HMAC :

```
curl --location --request POST https://{your_endpoint}.hcnx.eu/campaign
--header 'X-AUTH-TOKEN: azerty \
--header 'Content-Type: application/json' \
--data-raw '{
  "push":{
    "accountid":"HIGH_TEST",
    "message":{
      "text":"Message de test",
      "to":[
        {
          "value":"+33629519546"
        }
      ]
    },
    "hmac":"c02362f913ba3b25f94040c583fc5e01"
  }
}
```

9.3 Exemple d'un script de calcul Hmac en php :

Une clé hmac_key est associée au compte client puis transmise à celui-ci afin d'effectuer le même calcul des 2 côtés et vérifier l'intégrité des données transmises.

Exemple :

```
<?php
```

```
function hmac_calculate($array, $hmac_key = ""){
    // Eviter que les paramètres pour le calcul du HmacService intègre un HmacService...
    if (isset($array['hmac'])) {
        unset($array['hmac']);
    }

    // En fonction de l'api appelée
    if (isset($parameters['push']['hmac'])) {
        unset($parameters['push']['hmac']);
    }

    // Algo
    // ksort($array);

    return hash_hmac('sha256', http_build_query($array), $hmac_key);
}
```

9.4 Table de caractères

Caractères autorisés dans le contenu des messages :

À l'exception des caractères LINE FEED et CARRIAGE RETURN, tous les caractères de la table ISO-8859-1 qui font également partie de la table ETSI GSM 03.38 peuvent être utilisés directement .

Lien vers la table ETSI GSM 03.38 : https://en.wikipedia.org/wiki/GSM_03.38

Précisions

- Les retours chariot doivent être représentés par un caractère 127 (0x7F) ;
- Le signe euro (EURO SIGN) doit être représenté par le caractère 128 (0x80) ou 164 (0xA4) ;
- Les caractères ETSI GSM 03.38 qui n'existent pas dans la table ISO-8859-1 n'ont pas de représentation smstext et ne peuvent donc pas être utilisés ;
- Les caractères ISO-8859-1 qui n'existent pas dans l'ETSI GSM 03.38 seront convertis dans le caractère le plus proche (par exemple : ç sera remplacé par c). Le détail est donné dans la table ci-dessous.
- **Les caractères de la table d'extension du GSM 03.38 : ^ { } \ [~] | € comptent comme deux caractères dans les 160 caractères autorisés dans un message.**

Spécificité des SMS longs :

Pour un SMS long, c'est à dire dépassant les 160 caractères, la base sur laquelle vous devez vous fier pour compter le nombre de SMS composant votre message est de 153 et non plus 160. En effet, l'équivalent de 7 caractères sont utilisés pour réaliser la jonction entre chaque SMS.

Exemples :

Un message de 160 caractères compte pour 1 SMS

Un message de 306 caractères compte pour 2 SMS (153+153)

Un message de 320 caractères compte pour 3 SMS (153+153+14)

La table suivante indique pour chaque caractère de la table ISO-8859-1 le caractère de la table GSM 03.38 utilisé pour le représenter :

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	N/A	<SH>	<SX>	<EX>	<ET>	<EQ>	<AK>	<BL>	<BS>	<HT>	<LF>	<VT>	<FF>	<CR>	<SO>	<SI>
	N/A	?	?	?	?	?	?	?	?		<LF>	?	<FF>	<CR>	?	?
1x	<DL>	<D1>	<D2>	<D3>	<D4>	<NK>	<SY>	<EB>	<CN>		<SB>	<EC>	<FS>	<GS>	<RS>	<US>
	?	?	?	?	?	?	?	?	?	?	?	<EC>	?	?	?	?
2x		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	?	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	<DT>
	p	q	r	s	t	u	v	w	x	y	z	{		}	~	<LF>
8x	<PA>	<HO>	<BH>	<NH>	<IN>	<NL>	<SA>	<ES>	<HS>	<HJ>	<VS>	<PD>	<PU>	<RI>	<S2>	<S3>
	<PA>	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

9x	<DC>	<P1>	<P2>	<TS>	<CC>	<MW>	<SG>	<EG>	<SS>	<GC>	<SC>	<CI>	<ST>	<OC>	<PM>	<AC>
	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Ax	<NS>	i	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
		i	c	£	¤	¥	¦	§	"	C	a	<	!	-	R	-
Bx	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
	o	?	2	3	'	u	?	.	,	i	o	>	?	?	?	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
	A	A	A	A	Ä	Å	Æ	Ç	E	É	E	E	I	I	I	I
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
	?	Ñ	O	O	O	O	Ö	x	Ø	U	U	U	Ü	Y	?	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
	à	a	a	a	ä	å	æ	c	è	é	e	e	ì	i	i	i
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
	?	ñ	ò	o	o	o	ö	?	ø	ù	u	u	ü	y	?	y

Et voici la table GSM 03.38 complète :

Hexa	Decimal	Name	Character	Decimal in smstext
0x00	0	COMMERCIAL AT	@	64
0x01	1	POUND SIGN	£	163
0x02	2	DOLLAR SIGN	\$	36
0x03	3	YEN SIGN	¥	165
0x04	4	LATIN SMALL LETTER E WITH GRAVE	è	232
0x05	5	LATIN SMALL LETTER E WITH ACUTE	é	233
0x06	6	LATIN SMALL LETTER U WITH GRAVE	ú	250
0x07	7	LATIN SMALL LETTER I WITH GRAVE	ì	236
0x08	8	LATIN SMALL LETTER O WITH GRAVE	ò	242
0x09	9	LATIN CAPITAL LETTER C WITH CEDILLA	Ç	199
0x0A	10	LINE FEED		127
0x0B	11	LATIN CAPITAL LETTER O WITH STROKE	Ø	216
0x0C	12	LATIN SMALL LETTER O WITH STROKE	ø	248
0x0D	13	CARRIAGE RETURN		127
0x0E	14	LATIN CAPITAL LETTER A WITH RING ABOVE	Å	197
0x0F	15	LATIN SMALL LETTER A WITH RING ABOVE	å	229
0x10	16	GREEK CAPITAL LETTER DELTA	Δ	
0x11	17	LOW LINE	—	95
0x12	18	GREEK CAPITAL LETTER PHI	Φ	
0x13	19	GREEK CAPITAL LETTER GAMMA	Γ	
0x14	20	GREEK CAPITAL LETTER LAMBDA	Λ	
0x15	21	GREEK CAPITAL LETTER OMEGA	Ω	
0x16	22	GREEK CAPITAL LETTER PI	Π	
0x17	23	GREEK CAPITAL LETTER PSI	Ψ	

0x18	24	GREEK CAPITAL LETTER SIGMA	Σ	
0x19	25	GREEK CAPITAL LETTER THETA	Θ	
0x1A	26	GREEK CAPITAL LETTER XI	Ξ	
0x1B	27	ESCAPE TO EXTENSION TABLE		
0x1B0A	27 10	FORM FEED		12
0x1B14	27 20	CIRCUMFLEX ACCENT	^	94
0x1B28	27 40	LEFT CURLY BRACKET	{	123
0x1B29	27 41	RIGHT CURLY BRACKET	}	125
0x1B2F	27 47	REVERSE SOLIDUS (BACKSLASH)	\	92
0x1B3C	27 60	LEFT SQUARE BRACKET	[	91
0x1B3D	27 61	TILDE	~	126
0x1B3E	27 62	RIGHT SQUARE BRACKET	]	93
0x1B40	27 64	VERTICAL BAR		124
0x1B65	27 101	EURO SIGN	€	27 101
0x1C	28	LATIN CAPITAL LETTER AE	Æ	198
0x1D	29	LATIN SMALL LETTER AE	æ	230
0x1E	30	LATIN SMALL LETTER SHARP S (German)	ß	223
0x1F	31	LATIN CAPITAL LETTER E WITH ACUTE		202
0x20	32	SPACE		32
0x21	33	EXCLAMATION MARK	!	33
0x22	34	QUOTATION MARK	«	34
0x23	35	NUMBER SIGN	#	35
0x24	36	CURRENCY SIGN	¤	164
0x25	37	PERCENT SIGN	%	37
0x26	38	AMPERSAND	&	38
0x27	39	APOSTROPHE	'	39
0x28	40	LEFT PARENTHESIS	(	40
0x29	41	RIGHT PARENTHESIS	)	41
0x2A	42	ASTERISK	*	42
0x2B	43	PLUS SIGN	+	43
0x2C	44	COMMA	,	44
0x2D	45	HYPHEN-MINUS	-	45
0x2E	46	FULL STOP	.	46
0x2F	47	SOLIDUS (SLASH)	/	47
0x30	48	DIGIT ZERO	0	48
0x31	49	DIGIT ONE	1	49
0x32	50	DIGIT TWO	2	50
0x33	51	DIGIT THREE	3	51
0x34	52	DIGIT FOUR	4	52
0x35	53	DIGIT FIVE	5	53
0x36	54	DIGIT SIX	6	54
0x37	55	DIGIT SEVEN	7	55
0x38	56	DIGIT EIGHT	8	56
0x39	57	DIGIT NINE	9	57
0x3A	58	COLON	:	58
0x3B	59	SEMICOLON	;	59

0x3C	60	LESS-THAN SIGN	<	60
0x3D	61	EQUALS SIGN	=	61
0x3E	62	GREATER-THAN SIGN	>	62
0x3F	63	QUESTION MARK	?	63
0x40	64	INVERTED EXCLAMATION MARK	!	161
0x41	65	LATIN CAPITAL LETTER A	A	65
0x42	66	LATIN CAPITAL LETTER B	B	66
0x43	67	LATIN CAPITAL LETTER C	C	67
0x44	68	LATIN CAPITAL LETTER D	D	68
0x45	69	LATIN CAPITAL LETTER E	E	69
0x46	70	LATIN CAPITAL LETTER F	F	70
0x47	71	LATIN CAPITAL LETTER G	G	71
0x48	72	LATIN CAPITAL LETTER H	H	72
0x49	73	LATIN CAPITAL LETTER I	I	73
0x4A	74	LATIN CAPITAL LETTER J	J	74
0x4B	75	LATIN CAPITAL LETTER K	K	75
0x4C	76	LATIN CAPITAL LETTER L	L	76
0x4D	77	LATIN CAPITAL LETTER M	M	77
0x4E	78	LATIN CAPITAL LETTER N	N	78
0x4F	79	LATIN CAPITAL LETTER O	O	79
0x50	80	LATIN CAPITAL LETTER P	P	80
0x51	81	LATIN CAPITAL LETTER Q	Q	81
0x52	82	LATIN CAPITAL LETTER R	R	82
0x53	83	LATIN CAPITAL LETTER S	S	83
0x54	84	LATIN CAPITAL LETTER T	T	84
0x55	85	LATIN CAPITAL LETTER U	U	85
0x56	86	LATIN CAPITAL LETTER V	V	86
0x57	87	LATIN CAPITAL LETTER W	W	87
0x58	88	LATIN CAPITAL LETTER X	X	88
0x59	89	LATIN CAPITAL LETTER Y	Y	89
0x5A	90	LATIN CAPITAL LETTER Z	Z	90
0x5B	91	LATIN CAPITAL LETTER A WITH DIAERESIS	Ä	196
0x5C	92	LATIN CAPITAL LETTER O WITH DIAERESIS	Ö	214
0x5D	93	LATIN CAPITAL LETTER N WITH TILDE	Ñ	209
0x5E	94	LATIN CAPITAL LETTER U WITH DIAERESIS	Ü	220
0x5F	95	SECTION SIGN	§	167
0x60	96	INVERTED QUESTION MARK	¿	191
0x61	97	LATIN SMALL LETTER A	a	97
0x62	98	LATIN SMALL LETTER B	b	98
0x63	99	LATIN SMALL LETTER C	c	99
0x64	100	LATIN SMALL LETTER D	d	100
0x65	101	LATIN SMALL LETTER E	e	101
0x66	102	LATIN SMALL LETTER F	f	102
0x67	103	LATIN SMALL LETTER G	g	103
0x68	104	LATIN SMALL LETTER H	h	104
0x69	105	LATIN SMALL LETTER I	i	105

0x6A	106	LATIN SMALL LETTER J	j	106
0x6B	107	LATIN SMALL LETTER K	k	107
0x6C	108	LATIN SMALL LETTER L	l	108
0x6D	109	LATIN SMALL LETTER M	m	109
0x6E	110	LATIN SMALL LETTER N	n	110
0x6F	111	LATIN SMALL LETTER O	o	111
0x70	112	LATIN SMALL LETTER P	p	112
0x71	113	LATIN SMALL LETTER Q	q	113
0x72	114	LATIN SMALL LETTER R	r	114
0x73	115	LATIN SMALL LETTER S	s	115
0x74	116	LATIN SMALL LETTER T	t	116
0x75	117	LATIN SMALL LETTER U	u	117
0x76	118	LATIN SMALL LETTER V	v	118
0x77	119	LATIN SMALL LETTER W	w	119
0x78	120	LATIN SMALL LETTER X	x	120
0x79	121	LATIN SMALL LETTER Y	y	121
0x7A	122	LATIN SMALL LETTER Z	z	122
0x7B	123	LATIN SMALL LETTER A WITH DIAERESIS	ä	228
0x7C	124	LATIN SMALL LETTER O WITH DIAERESIS	ö	246
0x7D	125	LATIN SMALL LETTER N WITH TILDE	ñ	241
0x7E	126	LATIN SMALL LETTER U WITH DIAERESIS	ü	252
0x7F	127	LATIN SMALL LETTER A WITH GRAVE	à	224
0x80	128	EURO SIGN	€	
0xA4	164	EURO SIGN	€	

9.5 Erreurs XML

Voici la liste des erreurs XML qui peuvent être rencontrées :

```

XML_ERROR_NONE
XML_ERROR_NO_MEMORY
XML_ERROR_SYNTAX
XML_ERROR_NO_ELEMENTS
XML_ERROR_INVALID_TOKEN
XML_ERROR_UNCLOSED_TOKEN
XML_ERROR_PARTIAL_CHAR
XML_ERROR_TAG_MISMATCH
XML_ERROR_DUPLICATE_ATTRIBUTE
XML_ERROR_JUNK_AFTER_DOC_ELEMENT
XML_ERROR_PARAM_ENTITY_REF
XML_ERROR_UNDEFINED_ENTITY
XML_ERROR_RECURSIVE_ENTITY_REF
XML_ERROR_ASYNC_ENTITY
XML_ERROR_BAD_CHAR_REF
XML_ERROR_BINARY_ENTITY_REF
XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF
XML_ERROR_MISPLACED_XML_PI
XML_ERROR_UNKNOWN_ENCODING
XML_ERROR_INCORRECT_ENCODING

```

XML_ERROR_UNCLOSED_CDATA_SECTION
XML_ERROR_EXTERNAL_ENTITY_HANDLING

9.6 Liste des codes statuts

Voici la liste des statuts qui peuvent vous être retournés :

NOT_SENT=>1 (envoyé au routeur, non encore envoyé à l'opérateur)
ERROR=>2 (impossible de contacter le routeur)
QUEUE=>3 (pris en compte par le routeur)
SENT=>4 (envoyé à l'opérateur)
RECEIVED=>6 (reçu par le destinataire)
ERROR_NPAI=>11 ("N'habites Pas à l'Adresse Indiquée" : l'opérateur contacté ne connaît pas ou plus ce numéro)
ERROR_EXPIRED=>12 (Pas de réception de RECEIVED après SENT au bout d'un certain temps, en général 24h)
ERROR_INVOP=>13 (Opérateur invalide, numéro non rattaché à une tranche opérateur)
ERROR_NETWORK=>14 (Problème réseau lors de l'acheminement)
ERROR_CREDIT=>15 (Crédit insuffisant)
ERROR_UNKNOWN=>16 (Autres erreurs, ...)
ERROR_BLOCKED=>23 (Destinataire bloqué par l'opérateur)
ERROR_NPAIPORTED=>25