Caractéristiques Techniques d'Accès au Service

FREE PRO SmarText



Table des matières

Table des matières 2
Résumé 3
Document 3
Aperçu 3
Historique 3
Contributeurs 3
Glossaire 4
1. Introduction 5
2. Présentation du service Free Pro SmarText 5
2.1. Architecture générale 5
2.2. Éléments du Service Free Pro SmarText 7
2.2.1. Destinations 7
2.2.2. Affichage du nom d'émetteur 7
2.2.3. Fragmentation des messages 7
2.2.4. Capacité du service / délai de transmission 8
2.2.5. Rapport d'émission et réception 8
2.3. Prérequis nécessaires au Service FREE PRO SmarText 9
2.3.1. Accès à Internet 9
2.3.2. Bande passante 9
2.3.3. Adressage IP 9
2.3.4. Protocole SOAP 9
2.3.5. Redondance 10
2.3.6. Sécurité et confidentialité 10
2.4. Fonctions des WebServices Free Pro SmartText 11
2.4.1. Envoi en masse d'un message vers une liste de numéros 11
2.4.2. Consultation des rapports d'émission 11
3. Annexes 13
3.1. Fichier WSDL 13
3.2. Exemples de requêtes et réponses 17
3.2.1. Envoi d'un message 17
3.2.2. Consultation du rapport d'émission 17
3.3. Liste et signification des erreurs SMPP 18
3.3.1. Erreurs temporaires 18
3.3.2. Erreurs définitives 18

Résumé

Ce document décrit les spécifications techniques d'Accès au Service FREE PRO SmarText à savoir la fourniture d'envoi de SMS en masse.

Document

Aperçu

Document	Free Pro SmarText
Version	1.4
Date	12.02.2025
Auteur	Jean-Baptiste MASSON
Nom du fichier	STAS-FREEPRO_SMARTEXT.docx

Historique

DATE	VERSION	CRÉÉ PAR	DESCRIPTION DE LA MODIFICATION
14/09/2016	1.0	Jean-Baptiste MASSON	Création
07/10/2016	1.1	Jean-Baptiste MASSON	Évolution API – Simplification de la partie réseau
02/12/2016	1.2	Jean-Baptiste MASSON	Finalisation des accès HTTPS
12/10/2017	1.3	Jean-Baptiste MASSON	Délai d'historisation
12/02/2025	1.4	Jean-Baptiste MASSON	Rebranding Free Pro

Contributeurs

Nom	Département
Jean-Baptiste Masson	Pôle voix

Glossaire

Terme	Définition
SMS	Short Message Service
SMPP	Short Message Peer to Peer
WSDL	Web Service Description Language
SOAP	Protocole de WebServices
HTTP	Hyper Text Transport Protocol
HTTPS	Secured Hyper Text Transport Protocol
API	Application Programmable Interface
SHORTCODE	Numéro court
OADC	Originator Address Code

1. Introduction

Ce document spécifie les caractéristiques techniques d'accès au service Free Pro SmarText fourni par Free Pro.
Ces spécifications décrivent :

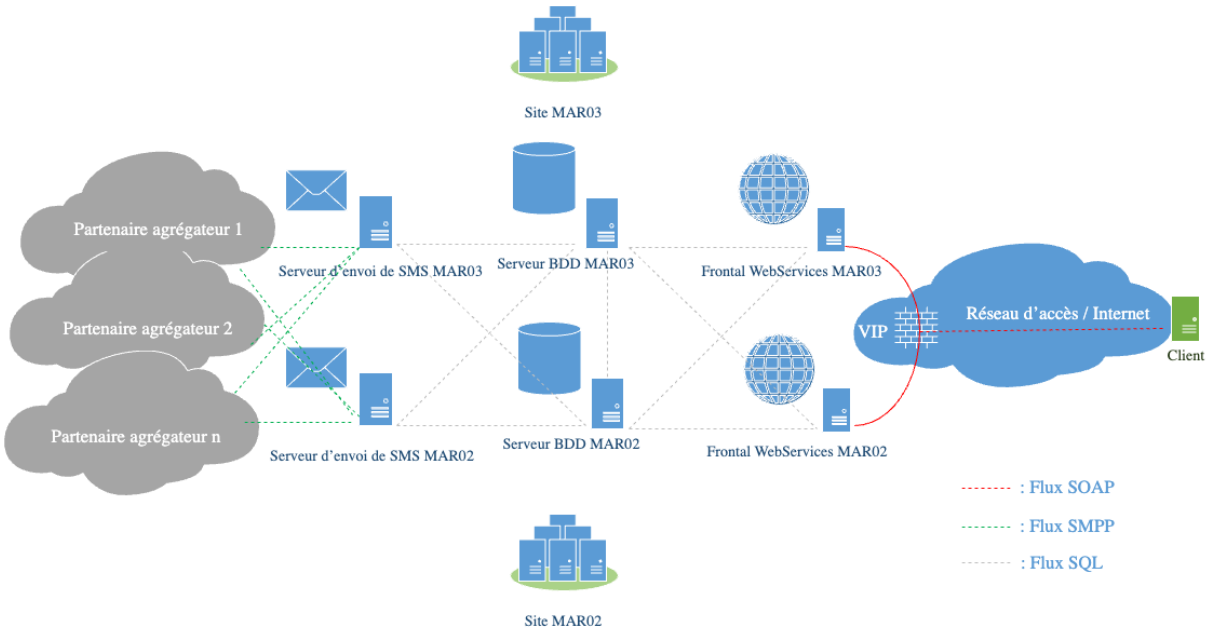
- Les caractéristiques du service
- Les modes d'accès au service
- Les caractéristiques techniques des Interfaces de Service
- Les différentes options possibles de mise en œuvre du Service

2. Présentation du service Free Pro SmarText

2.1. Architecture générale

Le service d'envoi de messages de type texte en masse par SMS propose une interface WebServices en protocole SOAP, disponible depuis le réseau Internet. Tout message soumis par le client via des requêtes WebServices est enregistré sur l'infrastructure de Free Pro pour être envoyé dans les délais les plus brefs sur les terminaux mobiles, via des partenaires de Free Pro.
Le service permet également la réception de réponses SMS.
L'infrastructure entièrement redondée est composée de trois types d'éléments distincts :

- Des serveurs Web hébergeant les WebServices. Toute requête donne lieu à des échanges avec la base de données du service.
- Des serveurs hébergeant la base de données. Cette dernière stocke tous les envois effectués et leur statut.
- Des passerelles d'interconnexion SMPP avec des partenaires. Ces passerelles sont alimentées par la base de données. Le statut de chaque message est maintenu à jour en temps réel dans la base de données.



Infrastructure d'envoi de SMS en masse de Free Pro

Cette infrastructure est actuellement redondée sur 2 sites géographiques distincts de Marseille.
Nos infrastructures sont disponibles sur les sites Free Pro suivants :

- MAR02 : Marseille – Datacenter Free Pro – 71 avenue André Roussin - 13016 Marseille, France
- MAR03 : Marseille - Interxion MRS2 - 6 bassin Léon Gourret- 13015 Marseille, France

Le point d'entrée des connexion SOAP est unique via l'utilisation d'une IP virtuelle, portée par un load balancer. Tout requête est distribuée entre les deux serveurs hébergeant les WebServices. Ces deux serveurs fonctionnent dans un mode actif-actif. En cas de panne d'un serveur, celui-ci ne reçoit plus de requêtes du load balancer.
De plus, chaque serveur de la solution fonctionne sur une infrastructure virtualisée VMware. En cas de panne du serveur physique hébergeant le serveur virtualisé, ce dernier est déplacé à chaud sur un autre serveur physique sans interruption du service.

2.2. Éléments du Service Free Pro SmarText

Le service FREE PRO SmarText est une offre d'envoi en masse de messages SMS fournie par Free Pro au travers de partenaires interconnectés aux différents opérateurs de téléphonie mobile.
Les messages envoyés respectent les standards SMS.

2.2.1. Destinations

Les destinations supportées sont toutes les destinations nationales et internationales.
Important : Les DOMs sont considérés comme des destinations internationales. Tout numéro de mobile vers un DOM est automatiquement converti par la plateforme SmarText dans le format international correspondant (cf. Tableau ci-dessous)

Destination	Format national	Format international E164
Guadeloupe	0690PQMCDU 0691PQMCDU	+590690PQMCDU +590691PQMCDU
Martinique	0696PQMCDU 0697PQMCDU	+596696PQMCDU +597697PQMCDU
Guyane	0694PQMCDU	+594694PQMCDU
Réunion	0692PQMCDU 0693PQMCDU	+262692PQMCDU +262693PQMCDU
Mayotte	0639PQMCDU	+262639PQMCDU

2.2.2. Affichage du nom d'émetteur

Le service SmarText supporte la présentation d'un nom d'émetteur personnalisé au terminal de réception du message (personnalisation de l'OADC), en lieu et place du shortcode par défaut.
Ce service n'est pas garanti à l'international.
Le nom de l'émetteur est limité à 11 caractères ASCII. Les accents ou caractères spéciaux ne sont pas supportés. Les numéros sont interdits. Les chiffres sont autorisés si l'ensemble de la chaîne n'est pas un numéro.
Par exemple : « 0612345678 » est interdit mais « EmetteurN2 » est autorisé.
Si aucun nom d'émetteur n'est spécifié, le terminal de réception affichera comme émetteur le shortcode utilisé (numéro à 5 digits). Ce shortcode dépend de l'agrégateur via lequel le service SmarText émet le SMS.

2.2.3. Fragmentation des messages

Tout message envoyé dépassant une certaine taille est découpé en plusieurs SMS. Le terminal de réception se charge de reconstituer le message complet à partir des fragments reçus. Cette fragmentation est effectuée par la plateforme SmarText. Les critères de fragmentation dépendent du contenu du message. En effet, l'utilisation de caractères spéciaux

dans les messages peut nécessiter un changement de la table d'encodage utilisée dans tous les SMS d'un message.

2.2.4. Capacité du service / délai de transmission

La capacité d'envoi de l'infrastructure d'envoi de SMS en masse de Free Pro est de 360 000 SMS par heure.

2.2.5. Rapport d'émission et réception

Les requêtes WebServices permettent à tout moment de requêter les statuts d'émission et de réception des messages envoyés dans les derniers 90 jours. Les codes d'erreur éventuels peuvent être présents dans le rapport. Les erreurs possibles sont spécifiées en annexe. Il existe des erreurs temporaires et définitives. Toute erreur temporaire donne lieu à de nouvelles tentatives de livraison du message.
A noter que les date et heure de réception peuvent être imprécises, certains opérateurs n'envoyant pas cette information à nos partenaires. Les date et heure spécifiées dans le rapport correspondent alors à l'instant de la réception de l'accusé sur la plateforme du partenaire.
D'une manière générale, les dates et heure de réception sont exactes et précises pour les messages envoyés vers les réseaux Orange et Free mobile. Elles ne le sont pas vers les réseaux Bouygues Telecom et SFR, pour lesquels la précision de l'heure de réception ne descend pas sous la minute.

2.3. Prérequis nécessaires au Service FREE PRO SmarText

2.3.1. Accès à Internet

Un accès à Internet en IPv4 est requis pour bénéficier du service SmarText.

2.3.2. Bande passante

La bande passante symétrique nécessaire pour l'utilisation du service dans des conditions optimales est de 512 Kbits/s (valeur indicative, non vérifiée).

2.3.3. Adressage IP

L'adressage IP utilisé est de type public en IPv4 (IPv6 non supporté pour le moment).
L'adresse IP utilisée pour joindre le service est 78.153.232.3. Le FQDN associé est : gw01-mar-sms.jaguar-network.com

2.3.4. Protocole SOAP

Le protocole supporté entre la plateforme Free Pro et le client final est le SOAP version 1.2 selon la recommandation W3C.

Les WebServices sont décrits dans un fichier WSDL fourni au client par Free Pro. Ce fichier est présent en annexe de ce document.

2.3.4.1. Transport / Chiffrement

Les transactions de l'API sont transportées en HTTPS uniquement. Les versions de transport supportées sont le TLS 1.2 et 1.3. Les versions TLS 1.0, 1.1, ainsi que le SSL ne sont pas supportées pour des raisons de sécurité.

2.3.4.2. Authentification

Free Pro fournit un login et un mot de passe d'authentification au service.

Cette authentification est nécessaire sur toutes les requêtes SOAP.

La méthode d'authentification actuellement supportée est la méthode basique proposée par le protocole HTTPS, en mode préemptif.

2.3.4.3. Description des services

Les WebServices sont décrits dans un fichier WSDL fourni au client par Free Pro. Ce fichier est présent en annexe de ce document.

2.3.5. Redondance

Il est recommandé de disposer d'une redondance de l'accès à Internet utilisé vers la plateforme SmarText. La sécurisation de cet accès n'est pas traitée dans ce document.

2.3.6. Sécurité et confidentialité

La sécurité et la confidentialité sont assurée à plusieurs niveaux :

- Réseau/Transport : via l'utilisation protocole HTTPS pour un chiffrement des données échangées.
- Applicatif : via l'utilisation d'une authentification par couple login / mot de passe.

2.4. Fonctions des WebServices Free Pro SmartText

2.4.1. Envoi en masse d'un message vers une liste de numéros

Une requête SOAP « SendMessage » d'envoi de message d'un message consiste en l'émission d'un même message à une liste de destinataires fournie en paramètres dans la requête.
Pour des raisons techniques ; afin de ne pas alourdir les requêtes, chaque envoi se limite actuellement à une liste de 128 destinataires au maximum. Ainsi, l'envoi d'un même message à un plus grand nombre de destinataires doit se faire en plusieurs requêtes, avec un découpage par paquets de 128 destinataires.
Celle limite est susceptible d'évoluer.

Les paramètres à fournir dans la requête sont :

Paramètres d'entrée	**Type**	**Commentaires**
Nom de l'émetteur	Chaîne	Nom affiché en émetteur du SMS sur le téléphone mobile.
Texte du message	Chaîne	
Séquence	Type complexe SOAP (cf. WSDL)	Limité actuellement à 128 éléments au maximum.
Destinataire	Chaîne	Numéro au format national ou E164.

Si la requête s'est correctement exécutée, la réponse contient les paramètres :

Paramètres de sortie	**Type**	**Commentaires**
Identifiant de l'envoi	Chaine	UUID de la requête. Cet identifiant doit être mémorisé pour toute requête ultérieure concernant cet envoi.

En cas d'erreur SOAP, la réponse contient un code d'erreur et une description de l'erreur.

2.4.2. Consultation des rapports d'émission

Une requête SOAP « GetSendingStatus » permet à tout moment de consulter les rapports d'émission pour un ou plusieurs numéros de destination d'un envoi donné.
Pour ce faire, il faut avoir mémorisé l'identifiant UUID de l'envoi.
Les paramètres à fournir dans la requête sont :

Paramètres d'entrée	**Type**	**Commentaires**
Identifiant de l'envoi	Chaîne	UUID, à récupérer de la requête d'envoi.
Liste de destinataires	Chaîne	Type complexe SOAP (cf. WSDL)

		Limité actuellement à 128 destinataires au maximum.

Si aucune liste de destinataires n'est fournie, la requête concerne tous les numéros de l'envoi.
Si la requête s'est correctement exécutée, la réponse contient les paramètres :

Paramètres de sortie	**Type**	**Commentaires**
Identifiant de l'envoi	Chaîne	UUID, répété.
Séquence	Type complexe SOAP (cf. WSDL)	Limité actuellement à 128 éléments au maximum.
Numéro	Chaîne	Numéro au format E164 (même si le numéro est au format national dans la requête)
Code d'état	Entier	Cf. tableau ci-dessous.
Date et heure d'envoi	Date et heure	Si message envoyé.
Date et heure de livraison	Date et heure	Si message livré.
Code d'erreur de l'opérateur	Chaîne	Code d'erreur SMPP (cf. Annexe).
Message d'erreur de l'opérateur	Chaîne	Traduction sous forme de message explicite du code d'erreur de l'opérateur.
Nombre de fragments	Entier	Indique le nombre de SMS nécessaires à l'envoi du message (selon la longueur et les caractères utilisés).

En cas d'erreur SOAP, la réponse contient un code d'erreur et une description de l'erreur.

Les codes d'état de livraison d'un SMS sont les suivants (liste non exhaustive, susceptible d'être enrichie) :

Code	**Signification**
0	Message non envoyé
1	Message envoyé
2	Message reçu
3	Message non reçu (statut définitif)

3. Annexes

3.1. Fichier WSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="https://gw01-mar-sms.jaguar-network.com/smartext/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"          name="smartext"
targetNamespace="https://gw01-mar-sms.jaguar-network.com/smartext/">
  <wsdl:types>
    <xsd:schema         targetNamespace="https://gw01-mar-sms.jaguar-
network.com/smartext/">
      <xsd:element name="RecipientList">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="1024">
           <xsd:element name="Number" type="xsd:string" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="SendMessageResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="GetSendingStatus">
           <xsd:complexType>
                <xsd:sequence>
                     <xsd:element                              name="in"
type="xsd:string"></xsd:element>
                </xsd:sequence>
           </xsd:complexType>
      </xsd:element>
      <xsd:element name="ReceptionReport">
           <xsd:complexType>
                <xsd:sequence minOccurs="0" maxOccurs="1024">
                     <xsd:element                          name="Number"
type="xsd:string"></xsd:element>
                     <xsd:element                    name="StatusCode"
type="xsd:int"></xsd:element>
                     <xsd:element                   name="SendingTime"
type="xsd:dateTime"></xsd:element>
                     <xsd:element name="DeliveryTime"
                          type="xsd:dateTime">
                     </xsd:element>
                     <xsd:element name="OperatorErrorCode"
                          type="xsd:string">
                     </xsd:element>
                     <xsd:element name="OperatorErrorString"
                          type="xsd:string">
```

```
                    </xsd:element>
                    <xsd:element                name="MessageParts"
type="xsd:int"></xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>


        <xsd:element name="SendMessageFault">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="SendMessageFault"
                        type="xsd:string">
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="GetSendingStatusFault">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="GetSendingStatusFault"
                        type="xsd:string">
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
             <xsd:complexType name="ReceptionReport2">
                <xsd:sequence                           minOccurs="0"
maxOccurs="1024"></xsd:sequence>
                <xsd:attribute                          name="Number"
type="xsd:string"></xsd:attribute>
                <xsd:attribute                            name="Code"
type="xsd:string"></xsd:attribute>
            </xsd:complexType>
            <xsd:complexType name="ReceptionReportList">
                <xsd:sequence>
                    <xsd:element            name="ReceptionReport"
type="tns:ReceptionReport"                        maxOccurs="1024"
minOccurs="0"></xsd:element>
                </xsd:sequence>
            </xsd:complexType>

            <xsd:complexType name="ReceptionReport">
                <xsd:sequence>
                    <xsd:element                        name="Number"
type="xsd:string"></xsd:element>
                    <xsd:element                    name="StatusCode"
type="xsd:int"></xsd:element>
                    <xsd:element name="SendingTime"
                        type="xsd:dateTime">
                    </xsd:element>
                    <xsd:element name="DeliveryTime"
                        type="xsd:dateTime">
                    </xsd:element>
                    <xsd:element name="OperatorErrorCode"
                        type="xsd:string">
```

```
                    </xsd:element>
                    <xsd:element name="OperatorErrorString"
                         type="xsd:string">
                    </xsd:element>
                    <xsd:element                name="MessageParts"
type="xsd:int"></xsd:element>
                </xsd:sequence>
            </xsd:complexType>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="SendMessageRequest">
    <wsdl:part element="tns:RecipientList" name="RecipientList"/>
    <wsdl:part name="SenderName" type="xsd:string"></wsdl:part>
    <wsdl:part name="Message" type="xsd:string"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="SendMessageResponse">
    <wsdl:part name="SendingID" type="xsd:string"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetSendingStatusRequest">
        <wsdl:part name="SendingID" type="xsd:string"></wsdl:part>
        <wsdl:part                              name="RecipientList"
element="tns:RecipientList"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetSendingStatusResponse">
      <wsdl:part name="SendingID" type="xsd:string"></wsdl:part>

      <wsdl:part                          name="ReceptionReportList"
type="tns:ReceptionReportList"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="SendMessageFault">
      <wsdl:part                                    name="ErrorCode"
element="tns:SendMessageFault"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetSendingStatusFault">
      <wsdl:part                                    name="ErrorCode"
element="tns:GetSendingStatusFault"></wsdl:part>
  </wsdl:message>
  <wsdl:portType name="smartext">
    <wsdl:operation name="SendMessage">
      <wsdl:input message="tns:SendMessageRequest"/>
      <wsdl:output message="tns:SendMessageResponse"/>
            <wsdl:fault                                 name="fault"
message="tns:SendMessageFault"></wsdl:fault>
        </wsdl:operation>
    <wsdl:operation name="GetSendingStatus">
      <wsdl:input message="tns:GetSendingStatusRequest"></wsdl:input>
      <wsdl:output
message="tns:GetSendingStatusResponse"></wsdl:output>
            <wsdl:fault                                 name="fault"
message="tns:GetSendingStatusFault"></wsdl:fault>
        </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="smartextSOAP" type="tns:smartext">
      <soap:binding style="document"
          transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="SendMessage">
```

```
            <soap:operation
                  soapAction="https://gw01-mar-sms.jaguar-
network.com/smartext/SendMessage" />
            <wsdl:input>
                  <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                  <soap:body use="literal" />
            </wsdl:output>
            <wsdl:fault name="fault">
                  <soap:fault use="literal" name="fault" />
            </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="GetSendingStatus">
            <soap:operation
                  soapAction="https://gw01-mar-sms.jaguar-
network.com/smartext/GetSendingStatus" />
            <wsdl:input>
                  <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                  <soap:body use="literal" />
            </wsdl:output>
            <wsdl:fault name="fault">
                  <soap:fault use="literal" name="fault" />
            </wsdl:fault>
      </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="smartext">
    <wsdl:port binding="tns:smartextSOAP" name="smartextSOAP">
      <soap:address            location="https://gw01-mar-sms.jaguar-
network.com/smartext/"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

3.2. Exemples de requêtes et réponses

3.2.1. Envoi d'un message

Requête :

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:smar="https://gw01-mar-sms.jaguar-network.com/smartext/">
   <soapenv:Header/>
   <soapenv:Body>
      <smar:RecipientList>
         <Number>0673454447</Number>
      </smar:RecipientList>
      <SenderName>Emetteur</SenderName>
      <Message>Message de test</Message>
   </soapenv:Body>
</soapenv:Envelope>
```

Réponse :

```
<SOAP-ENV:Envelope                                          xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Body>
      <SendingID>7c3aa3eb-e6e7-491c-bfc6-82c4016ec7af</SendingID>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.2. Consultation du rapport d'émission

Requête :

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:smar="https://gw01-mar-sms.jaguar-network.com/smartext/">
   <soapenv:Header/>
   <soapenv:Body>
      <SendingID>7c3aa3eb-e6e7-491c-bfc6-82c4016ec7af</SendingID>
      <smar:RecipientList>
           <Number>0673454447</Number>
      </smar:RecipientList>
   </soapenv:Body>
</soapenv:Envelope>
```

Réponse :

```
<SOAP-ENV:Envelope                                          xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Body>
      <SendingID>7c3aa3eb-e6e7-491c-bfc6-82c4016ec7af</SendingID>
      <ReceptionReportList>
         <ReceptionReport>
```

```
            <Number>+33673454447</Number>
            <StatusCode>2</StatusCode>
            <SendingTime>2016-10-07 16:01:12</SendingTime>
            <DeliveryTime>2016-10-07 16:01:14</DeliveryTime>
            <OperatorErrorCode/>
            <OperatorErrorString/>
            <MessageParts>1</MessageParts>
         </ReceptionReport>
      </ReceptionReportList>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.3. Liste et signification des erreurs SMPP

3.3.1. Erreurs temporaires

```
122 : Data corrupt while sending to operator
922 : Data corrupt while sending to application
123 : Rate limited by operator
823 : Rate limited by router
923 : Rate limited by application
126 : Message timed out in operator buffer
826 : Message timed out in router
926 : Message timed out in application
D26 : Application session timed out
129 : Operator network error
929 : Application network error
13D : Operator buffer full
33D : SIM card memory full
83D : Buffer full
93D : Application buffer full
33F : Mobile phone unreachable
8FC : Simulated retryable error
```

3.3.2. Erreurs définitives

```
120 : Unknown error received from operator
420 : Unknown billing error
820 : Unknown routing error
920 : Unknown error received from application
121 : Operator protocol error
921 : Application protocol error
124 : Encoding error while communicating with operator
824 : Encoding error in the router
924 : Encoding error while communicating with application
125 : Operator detected a duplicate message
825 : Router detected a duplicate message
925 : Application detected a duplicate message
426 : Billing session timed out
127 : Text too long for this operator
```

927 : Text too long for this application
D27 : Application session field too long
128 : Unauthorized access to operator network
928 : Unauthorized access to application network
130 : Invalid operator ID
830 : No route found (portability module did not find operator / routing did not find a suitable shortcode)
930 : Invalid application ID
331 : Invalid phone number
132 : Shortcode unknown by the operator
932 : Shortcode unknown by application
133 : Sender ID refused by operator connector
833 : Sender ID refused by router
933 : Invalid sender ID sent by application connector
534 : Invalid operator session
834 : Invalid routing session
D34 : Invalid application session
135 : Blacklisted by operator
835 : Blacklisted in router
336 : Ported MSISDN
137 : Previous message not found by operator
837 : Previous message not found in router
138 : Capped by operator
838 : Capped by router
439 : Not enough credits
43A : User not subscribed
13B : Operator rejected the message
43B : User refused billing
93B : Application rejected the message
13C : Operation not supported by the operator
43C : Billing mode not supported
93C : Operation not supported by the application
43E : User already subscribed
8FC : Simulated final error