



MobyT PHP SDK

TABLE DES MATIERES

INTRODUCTION	2
CONFIGURATION DE LA LIBRAIRIE	2
ENVOI DE SMS	2
ÉTAT DES MESSAGES	4
HISTORIQUE DES MESSAGES.....	5
CRÉDIT SMS DISPONIBLE	6
GESTION DES SOUS-COMPTES	6
GESTION DU CRÉDIT DES SOUS-COMPTES	8
TELECHARGEMENT DE LA LIBRAIRIE	9



INTRODUCTION

Les serveurs MobyT mettent à disposition des développeurs des interfaces de communication HTTP pour l'envoi et la réception de messages SMS, permettant la vérification de la validité des numéros de téléphone et la récupération des données concernant l'état des messages et l'historique SMS.

CONFIGURATION DE LA LIBRAIRIE

Modifier le username ET le password dans config.php:

```
<?php

define('mobyT_HOSTNAME','https://api.mobyT.fr');
define('mobyT_USERNAME','your_mobyT_login');
define('mobyT_PASSWORD','your_mobyT_password');
define('mobyT_DEFAULT_PORT',80);
define('mobyT_PROXY','');
define('mobyT_PROXY_PORT',80);

?>
```

Si vous utilisez un proxy, il est nécessaire de configurer dans la même classe l'entrée PROXY et PROXY_PORT.

La librairie téléchargeable se trouve à cette adresse :
http://www.mobyT.fr/pdf/sdk/sms_gateway_php.zip

ENVOI DE SMS

La classe mobyT_SMS (file sendsms.php) contient toutes les données nécessaires pour envoyer un message SMS par le gateway mobyT.

Une fois créé un objet SMS, il est nécessaire de définir :

- type de SMS : SMSTYPE_TOP, SMSTYPE_DIRECT (attribut sms_type) ;
- destinataires : numéro de téléphone des destinataires du message en format international ; pour ajouter un numéro à la liste des destinataires, utiliser la méthode add_recipient(\$string)
- message : attribut message, corps du message à envoyer, maximum 160 caractères pour un seul SMS, maximum 1000 caractères pour un SMS long (concaténé).

Attention! Les caractères ^ { } \ [~] | € sont calculés comme 2 caractères ; il est possible d'utiliser la méthode `length()` pour connaître le nombre de caractères du SMS et la méthode `count_smss()` pour demander le nombre de SMS à envoyer ;

- expéditeur: attribut `SMSSender`, nécessaire pour les SMS avec expéditeur personnalisé, maximum 11 caractères pour les suites de caractères alphanumériques (Ex: "Nom2Societe") attribut `sender`.

Vous pouvez également définir en option :

- la date et l'heure d'un envoi programmé (méthode `set_scheduled_delivery($unix_timestamp)`) ;
- une chaîne alphanumérique à associer à l'envoi, indispensable pour obtenir ensuite les informations concernant l'état des messages (attribut `order_id`). La longueur maximum est de 32 caractères, tous les caractères excédents ne seront pas considérés.

Si le `order_id` n'est pas précisé, le serveur en génère un unique.

Le `order_id` peut être utilisé dans 30 jours suivants pour demander le status du message.

Exemple :

```
require('../sendsms.php');
```

```
$sms = new MobyT_SMS();
$sms->sms_type = SMSTYPE_TOP;
$sms->add_recipient('+336062223333');
$sms->add_recipient('+336078765432');
$sms->message = 'hello world!';
$sms->sender = 'test';
$sms->set_immediate(); // or sms->set_scheduled_delivery($unix_timestamp)
$sms->order_id = '999FFF111';
echo 'About to send a message '. $sms->count_smss(). ' SMSs long ';
echo 'to '. $sms->count_recipients(). ' recipients </br>';
if ($sms->validate()) {
    $res = $sms->send();
    if ($res['ok']) {
        echo $res['sentsmss']. ' SMS sent, order id is '. $res['order_id']. ' </br>';
    } else {
        echo 'Error sending SMS : '. $sms->problem(). ' </br>';
    }
    } else {
        echo 'invalid SMS : '. $sms->problem(). ' </br>';
    }
```

ÉTAT DES MESSAGES

Après l'envoi des messages, il est possible d'en demander l'état en utilisant la fonction `moby_get_message_status($order_id)`. En précisant le id du SMS, il est possible de demander au serveur le résultat de l'envoi ; le serveur restitue une liste d'objets `mobyt_SMS_STATUS` contenant :

- le numéro de téléphone du destinataire (attribut `recipient`) ;
- l'état de l'envoi (attribut `status`) ;
- date et heure de réception du message, si le SMS a été reçu correctement par le destinataire. L'attribut `dt_received` contient la date/heure de réception dans le format d'une chaîne ; la méthode `get_sms_received_timestamp()` restitue la date/heure comme unix timestamp ; il est aussi possible d'utiliser la fonction `strtotime()` pour obtenir un array associatif, comme depuis [http ://it.php.net/manual/en/function strtotime.php](http://it.php.net/manual/en/function strtotime.php), le format utilisé est `MOBYT_DATE_TIME_FORMAT` (voir exemple).

Le nombre d'objets `MOBYT_SMS_STATUS` passés est contenu dans l'array. La chaîne de caractères est associée à l'array (voir l'exemple dans la page suivante).

L'attribut `status` peut prendre les valeurs suivantes :

- `SCHEDULED` // programmé, pas encore envoyé
- `SENT` // envoyé, il n'attend pas de delivery
- `DLVRD` // SMS reçu correctement
- `ERROR` // erreur dans l'envoi du SMS
- `TIMEOUT` // après 48h l'opérateur n'a pas fourni d'informations
- `TOOM4NUM` // trop de SMS pour le même destinataire dans les dernières 24 heures
- `TOOM4USER` // trop de SMS envoyés par l'utilisateur pendant les dernières 24 heures
- `UNKNPFX` // préfixe SMS invalide ou inconnu
- `UNKNRCPT` // numéro de téléphone du destinataire invalide ou inconnu
- `WAIT4DLVR` // message envoyé en attente de delivery
- `WAITING` // en attente, pas encore envoyé
- `UNKNOWN` // état inconnu

S'il n'est pas nécessaire de gérer en détails les éventuels erreurs d'envoi, la méthode `is_error()` renvoie `true` si le message est en erreur, autrement `false`. La méthode `str_status()` fournit une description textuelle de l'état.

Exemple :

```
require('../sms_status.php');

$status = mobyt_get_message_status('1F11DC4A77A8F');
if ($status['ok']) {
    for ($i=0;$i<$status['count'];$i++) {
        if ($status[$i]->is_error()) {
            echo 'Error sending sms to '.$status[$i]->recipient.' : ' ;
            echo $status[$i]->str_status() ;
        } else {
            echo 'Message to number '.$status[$i]->recipient.' : ' ;
            if ($status[$i]->status == 'DLVRD') {
                echo 'correctly delivered on ' ;
                echo date('l',$status[$i]->get_sms_received_timestamp()) ;
                echo ' at ' ;
                echo date('H:i',$status[$i]->get_sms_received_timestamp()) ;
            } else {
                echo $status[$i]->str_status() ;
            }
        }
        echo '<br/>' ;
    }
} else {
    echo 'Request failed : '.$status['errcode'].' - '.$status['errmsg'] ;
}
```

HISTORIQUE DES MESSAGES

La fonction `moby_get_sms_history($from,$to)` demande au serveur mobyt la liste des SMS envoyés dans un laps de temps spécifique ; un array d'objets `moby_SENT_SMS` est renvoyé. Les paramètres `$from` et `$to` sont unix timestamp.

Exemple :

```
require('../sms_history.php');

$history = mobyt_get_sms_history(time()-(7*24*60*60),time());
if ($history['ok']) {
    for ($i=0;$i<$history['count'];$i++) {
        echo 'SMS id '.$history[$i]->order_id.' sent from '.$history[$i]->sender ;
        echo ' to '.$history[$i]->recipients_count.' recipients<br/>' ;
    }
} else {
    echo 'Request failed : '.$history['errcode'].' - '.$history['errmsg'] ;
}
```

CRÉDIT SMS DISPONIBLE

La fonction `moby_get_credits()` demande au serveur mobyt la disponibilité de crédit ; un array d'objets `mobyt_CREDIT` est restitué. Chaque objet `mobyt_CREDIT` contient : le type, la nation (null si le crédit est international) et la quantité de crédits disponibles.

Exemple :

```
require('../credits.php');
```

```
$credits = mobyt_get_credits();  
if ($credits['ok']) {  
    for ($i=0 ; $i<$credits['count'] ; $i++) {  
        if (!$credits[$i]->is_international()) {  
            echo 'You can send ' . $credits[$i]->availability . ' smss';  
            echo ' of type ' . $credits[$i]->credit_type ;  
            echo ' in ' . $credits[$i]->nation . ' </br>';  
        } else {  
            if ($credits[$i]->credit_type == 'EE') {  
                echo 'You can send ' . $credits[$i]->availability ;  
                echo ' smss in foreign countries </br>';  
            }  
        }  
    }  
} else {  
    echo 'Request failed : ' . $status['errcode'] . ' - ' . $status['errmsg'];  
}
```

GESTION DES SOUS-COMPTES

Les utilisateurs qui disposent d'un super-compte peuvent utiliser une des méthodes de l'objet `SMSConnection` pour monitorer et gérer ses propres sous-comptes.

Plus précisément :

- `mobyt_getSubaccounts()`, renvoie une `List<Subaccount>` contenant tous les sous-comptes créés
- `mobyt_createSubaccount(Subaccount)` permet de créer un nouveau sous-compte ; le login et le password sont générés par mobyt et doivent être lus par l'objet `Subaccount` renvoyé
- `mobyt_lockSubaccount(Subaccount)` et `mobyt_unlockSubaccount(Subaccount)`, permettent d'activer/désactiver un sous-compte ; dans les deux cas un `Subaccount` est restitué.

Exemple :

```
require('../subaccounts.php');  
echo 'Listing subaccounts :</br>';  
$subs = mobyt_get_subaccounts();  
if ($subs['ok']) {  
    for ($i=0;$i<$subs['count'];$i++) {  
        echo 'Subaccount login : '.$subs[$i]->login.' active : '.$subs[$i]->active ;  
        echo ' type : '.$subs[$i]->credit_mode.' name : '.$subs[$i]->name.' '.$subs[$i]->surname.' <br/>';  
    }  
} else {  
    echo 'Request failed : '.$subs['errcode'].' - '.$subs['errmsg'];  
}  
echo 'Creating new subaccount :</br>';  
  
$subaccount = new mobyt_SUBACCOUNT();  
$subaccount->login = 'login_123';  
$subaccount->sub_password = 'xyz';  
$subaccount->credit_mode = 2 ;  
$subaccount->company_name = 'azienda prova';  
$subaccount->fiscal_code = 'XXXXXX80S11L111X';  
$subaccount->vat_number = '012345678901234';  
$subaccount->name = 'testname';  
$subaccount->surname = 'testsurname';  
$subaccount->email = 'testmail@your.it';  
$subaccount->address = 'testaddress';  
$subaccount->city = 'testcity';  
$subaccount->province = 'BO';  
$subaccount->zip = '38300';  
$subaccount->mobile = '3490000000';  
$subaccount->active = true ;  
$subaccount->createSubaccount();  
  
echo 'Subaccount login : '.$subaccount->login.' password : '.$subaccount->sub_password.'  
active : '.$subaccount->active.' <br/>';  
echo 'Disabling subaccount :</br>';  
$subaccount->lockSubaccount();  
echo 'Subaccount login : '.$subaccount->login.' password : '.$subaccount->sub_password.'  
active : '.$subaccount->active.' <br/>';  
echo 'Enabling subaccount :</br>';  
$subaccount->unlockSubaccount();  
echo 'Subaccount login : '.$subaccount->login.' password : '.$subaccount->sub_password.'  
active : '.$subaccount->active.' <br/>';
```

GESTION DU CRÉDIT DES SOUS-COMPTES

Les utilisateurs qui disposent d'un super-compte peuvent attribuer/enlever du crédit à ses propres sous-comptes.

Différentes méthodes de gestion du crédit SMS sont disponibles en fonction des caractéristiques du sous-compte.

- Si le paramètre de gestion des crédits est USE_SUPER_CREDIT il n'est pas possible d'attribuer/enlever du crédit au sous-compte.
- Si le sous-compte HAS_CREDIT il est possible de lui attribuer/enlever des crédits mobyt_moveCredits(CreditMovement). Pour connaître le crédit du sous-compte, utiliser la méthode mobyt_getSubaccountCredits(Subaccount), qui renvoie une liste Credit.
- Si le sous-compte utilise le paramètre USE_BOTH_CREDITS il est possible de lui attribuer du crédit en utilisant la méthode mobyt_createPurchase(CreditMovement) et de lui l'enlever avec la méthode mobyt_deletePurchase(CreditMovement). Pour visualiser les transferts de crédits depuis le super-compte vers les sous-comptes, utiliser la méthode mobyt_getPurchases(Subaccount), qui renvoie une List<CreditMovement>.

Exemple :

```
require('../credit_movements.php');
require('../credits.php');

echo 'Giving credits to subaccount xxxS_004<br>';
$cm1= new mobyt_CREDIT_MOVEMENT();
$cm1->subaccount_login = 'xxxS_004';
$cm1->super_to_sub = 'y';
$cm1->amount=35;
$cm1->message_type='N';
$cm1->moveCredits();

echo 'Listing credits for subaccount xxxS_004<br>';
$credits = mobyt_get_subaccount_credits('xxxS_004');
if ($credits['ok']) {
    for ($i=0;$i<$credits['count'];$i++) {
        if (!$credits[$i]->is_international()) {
            echo 'You can send '.$credits[$i]->availability.' smss';
            echo ' of type '.$credits[$i]->credit_type;
            echo ' in '.$credits[$i]->nation.' </br>';
        } else {
            if ($credits[$i]->credit_type == 'EE') {
                echo 'You can send '.$credits[$i]->availability;
                echo ' smss in foreign countries </br>';
            }
        }
    }
}
```



```

    }
}
} else {
    echo 'Request failed : '.$credits['errcode'].' - '.$credits['errmsg'];
}

echo 'Creating purchase for subaccount xxxS_001<br>';
$cm2= new mobyt_CREDIT_MOVEMENT();
$cm2->subaccount_login = 'xxxS_001';
$cm2->price = '56.7';
$cm2->sms_types = explode(';', 'N;L');
$cm2->price_per_message = explode(';', '0.02;0.009');
$cm2->createPurchase();

echo 'Listing purchases for subaccount xxxS_001<br>';
$purchases = mobyt_get_subaccount_purchases('xxxS_001');
if ($purchases['ok']) {
    for ($i=0; $i<$purchases['count']; $i++) {
        echo ($purchases[$i]->super_to_sub ? '-->' : '<-- ');
        echo $purchases[$i]->amount;
        echo '(price '.$purchases[$i]->price.', available '.$purchases[$i]->available_amount.')
ppm : ';

        for ($j=0; $j<count($purchases[$i]->sms_types); $j++) {
            echo $purchases[$i]->sms_types[$j]. ' '.$purchases[$i]-
            >price_per_message[$j]. ', ';
        }
        echo '<br>';
    }
} else {
    echo 'Request failed : '.$purchases['errcode'].' - '.$purchases['errmsg'];
}
echo 'Deleting last purchase<br>';
$purchases[0]->deletePurchase();

```

TELECHARGEMENT DE LA LIBRAIRIE

Vous trouverez l'archive contenant les fichiers source du SDK à cette adresse :

http://www.mobyt.fr/pdf/sdk/sms_gateway_php.zip