

# MPS INTERFACE

## SUBSCRIBE/UNSUBCRICE SERVICE

## INTERFACE

Date: 01 - 2019

### 1. Overview

#### 1.1. Function description

This function is defined for MPS to call when customer register (or unregister) a service. There're 2 case:

- User send SMS to MPS
- MPS cancel service for user after debit some days (depend on service) , MPS's renew process will send to partner

MPS support also: soap and json.

#### 1.2. Sequence Diagram

### 2. Requirement

#### 2.1. Web service WSDL or restful

URL: <http://x.x.x.x:yyyy/subscribe?wsdl> or any link from partner.

- o Client must implement to handle this link and return webservice description
- o The Client need to implement **subRequest** function as description below for MPS to call.

#### 3. subRequest function:

##### 3.1. subRequest function

This function will be called when an user subscribe (or unsubscribe) a service

Prototype:

Parameter	Description
username	Username that CP provide to <b>MPS</b>
Password	Password that CP provide to <b>MPS</b>
ServiceID	ServiceID provided by <b>MPS</b>
Msisdn	Phone number
chargetime	Subscribed Time, format yyyyMMddHHmmss
params	Parameter(s) for function 0: Subscribe 1: Unsubscribe 2: pending 3: Restore service
mode	For test function. - <b>CHECK</b> : check CP System with this

	content without any notify to user - <b>REAL</b> : get content for user
amount (optional)	Register fee which MPS has charged user
command (optional)	Register command
transactionId	transaction Id from MPS
sub_new (optional)	only for REGISTER case. true, false or #SUB_NEW (undefined). The first time register service.
promotionCode (optional)	promotion code if user got promotion.not empty mean no promotion.

Return format: <CODE>|<Description>

**Return Code description** table

0 Content sent to user Success  
1 Content sent to user Failed  
300 (empty) Invalid Parameter(s)  
301 (empty) Invalid username or password  
302 (empty) Server too busy

*Return might be:*

- **0** -> Success  
- **1** -> Failed  
- ...

**4. TPS, Time out & retry:**

- Time out: 5 seconds  
- Retry: 1 time/hour, max 3 times

## 5. Input and output XML

### 5.1. Input XML (MPS post to client)

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<subRequest xmlns="http://contentws/xsd">
<username>xxxxxx</username>
<password>yyyyyy</password>
<serviceid>ABCD1234</serviceid>
<msisdn>983456789</msisdn>
<chargetime>20130605131401</chargetime>
<params>0</params>
<mode>REAL</mode>
<amount>5000</amount>
<command>ON</command>
<transactionId>0300012345745</transactionId>
<subNew>true</subNew>
<promotionCode>ABCD1234_FIRST_REGISTRE_EXAMPLE</promotionCode>
</subRequest>
</S:Body>
</S:Envelope>
```

**Or, resumed version:**

#### 5.1.1 Register:

```
<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<subRequest xmlns="http://contentws/xsd">
<params>0</params> <!-- '0' means registration-->
<msisdn>MSISDN</msisdn><!-- MSISDN without 51-->
<serviceId>SUB_SERVICE_NAME</serviceId>
<amount>18000</amount> <!--Amount * 10000 charged if any-->
<chargetime>20190410100411</chargetime> <!--Date of transaction, all together-->
<transactionId>0700000320190410100409637</transactionId><!--Transaction ID-->
</subRequest>
</S:Body>
</S:Envelope>
```

### 5.1.2 Cancel:

```
<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<subRequest xmlns="http://contentws/xsd">
<params>1</params> <!-- '1' means cancellation-->
<msisdn>MSISDN</msisdn> <!-- MSISDN without 51-->
<serviceId>SUB_SERVICE_NAME</serviceId>
<amount>0</amount> <!-- Amount charged if any-->
<chargetime>20190410203100</chargetime> <!--Date of transaction, all together-->
<transactionId>0700000320190410203054899</transactionId> <!--Transaction ID-->
</subRequest>
</S:Body>
</S:Envelope>
```

### 5.2. Return XML (Client response)

```
<?xml version="1.0"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
<S:Body>
<subRequestResponse xmlns="http://contentws/xsd">
<return>0</return>
</subRequestResponse>
</S:Body>
</S:Envelope>
```

## 5. Input and output JSON

### request

```
{
  "username": "#USERNAME",
  "password": "#PASSWORD",
  "serviceid": "#SERVICE_NAME",
  "msisdn": "#MSISDN",
  "params": "#PARAMS",
  "content": "#CONTENT",
  "chargetime": "#CHARGE_TIME",
  "command": "#FULLSMS",
  "mode": "#MODE",
  "amount": "#AMOUNT",
  "transid": "#TRANSID"
}
```

Response

```
{  
  "return": "0"  
}
```