MPS EBP HTTP/WEB/MOBILE
NOTIFY API

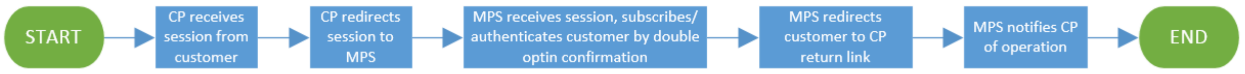
Date: 10 - 2019

1. Overview

1.1. Function description

- Step 1: Customer connects to CSP pre LP.
- Step 2: Customer clicks on “Go to Subscription LP”.
- Step 3: CP redirects customer to MPS Service LP (provided by BITEL) and adds TrackID (REQ) parameter if needed to track operation.
- Step 4: MPS will identify correctly encrypted requests, process them and require customer to double confirm subscription/authentication (by OTP or HE)
- Step 5: Customer will be redirected to configured Return Link (provided by CSP)
- Step 6: After completing transaction. MPS will attempt to notify CP of the operation (Whether subscription could not charge, could authenticate, could charge, etc).
- Step 7: CP will offer service contents to customer.

1.2. Sequence Diagram


START
CP receives session from customer
CP redirects session to MPS
MPS receives session, subscribes/ authenticates customer by double optin confirmation
MPS redirects customer to CP return link
MPS notifies CP of operation
END

2. Requirement

2.1. Web service WSDL or restful

URL: http://x.x.x.x:yyyy/notify?wsdl or any link from partner.

- CSP must implement to handle this link and return webservice description

- CSP needs to implement the described ***doNotify*** function as description below for MPS to call.

3. doNotify function:

- CSP develops the function that will receive the result of transactions from MPS system

4. Parameters description.

Parameter	Description
username	Username that CP provide to ***MPS***
Password	Password that CP provide to **MPS**
ServiceName	ServiceName provided by **MPS**
UserId	Userid, unique.
Req	request id from partner to track on response
params	response code 401: Subscription with not enough balance 0: Subscription with charge, check <amount> 408: Authentication (Customer already subscribed)
mode	For test function. - **CHECK**: check CP System with this

	content without any notify to user - **REAL**: get content for user
amount	Amount charged from operation
command (optional)	Register/cancel command
transactionId	transaction Id from MPS
startTime(optional)	Subscription time. Format yyyyMMddHHmmss
expireTime (optional)	next charge time, format yyyyMMddHHmmss
subStatus (optional)	subscriber status

Return format: **<CODE>|<Description>**
Return Code description table**Code Description Meaning**

- 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
- …

5. TPS, Time out & retry
- Time out: 5 seconds
- Retry: 1 time/hour, max 3 times

6. Input and output XML
6.1. Input XML (SMGW post to client)

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<doNotify xmlns="http://contentws/xsd">
<username>#USERNAME</username>
<password>#PASSWORD </password>
<serviceid>#SERVICE_NAME</serviceid>
<userid>#USERID</userid>
<cpRequestId>#REQ</cpRequestId>
<params>#PARAMS</params>
<amount>#AMOUNT</amount>
<command>#FULLSMS</command>
<transactionId>#TRANS_ID</transactionId>
<startTime>#SUB_START_DATE</startTime>
<expireTime>#SUB_END_DATE</expireTime>
<subStatus>#STATUS</subStatus>
</doNotify>
</S:Body>
</S:Envelope>
```

6.2. Return XML (Client response)

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

5. Input and output JSON

Request

```
{
"username":"#USERNAME",
"password":"#PASSWORD",
"serviceid":"#SERVICE_NAME",
"cpRequestId":"#REQ",
"userid":"#USERID",
"params":"#PARAMS",
"startTime":"#SUB_START_DATE",
"expireTime":"#SUB_END_DATE",
"subStatus":"#STATUS",
"command":"#FULLSMS",
"amount":"#AMOUNT",
"cpRequestId":"#REQ",
"transid":"#TRANSID"
}
```

Response

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

Examples for SOAP

Authentication:

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<doNotify xmlns="http://contentws/xsd">
<username>MPS</username>
<password>MPS</password>
<serviceid>JUEGOS_DAILY</serviceid>
<userid>66951294-9655-4a79-bba9-f08ed2d4b06d</userid>
<cpRequestId>#REQ</cpRequestId>
<params>408</params>
<amount>15900</amount>
<command>REGISTER</command>
<transactionId>1234</transactionId>
<startTime>#REGISTER_TIME</startTime>
</doNotify>
</S:Body>
</S:Envelope>
```

Registration with no charge (already charged: params=0 and amount=0)

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<doNotify xmlns="http://contentws/xsd">
<username>MPS</username>
<password>MPS</password>
<serviceid>JUEGOS_DAILY</serviceid>
<userid>66951294-9655-4a79-bba9-f08ed2d4b06d</userid>
<cpRequestId>#REQ</cpRequestId>
<params>0</params>
<amount>0</amount>
<command>REGISTER</command>
<transactionId>1234</transactionId>
<startTime>#REGISTER_TIME</startTime>
</doNotify>
</S:Body>
</S:Envelope>
```

Registration with charge (params=0 and amount!=0)

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<doNotify xmlns="http://contentws/xsd">
<username>MPS</username>
<password>MPS</password>
<serviceid> JUEGOS_DAILY </serviceid>
<userid>66951294-9655-4a79-bba9-f08ed2d4b06d</userid>
<cpRequestId>#REQ</cpRequestId>
<params>0</params>
<amount>12900</amount>
<command>REGISTER</command>
<transactionId>1234</transactionId>
<startTime>#REGISTER_TIME</startTime>
</doNotify>
</S:Body>
</S:Envelope>
```

Registration with not enough balance (params=401 and amount=0)

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<doNotify xmlns="http://contentws/xsd">
<username>MPS</username>
<password>MPS</password>
<serviceid>JUEGOS_DAILY</serviceid>
<userid>66951294-9655-4a79-bba9-f08ed2d4b06d</userid>
<cpRequestId>#REQ</cpRequestId>
<params>401</params>
<amount>0</amount>
<command>REGISTER</command>
<transactionId>1234</transactionId>
<startTime>#REGISTER_TIME</startTime>
</doNotify>
</S:Body>
</S:Envelope>
```