SMS API General

Version	Date	Author	Description
1.0	28-Jul-2016	DSC	Initial Version
1.1	20-Apr-2017	DSC	MNP Support

APIs

1. MO (Mobile Originated) Transactions API

This API is used to receive the messages sent by mobile subscriber. API is a HTTP POST method call made by DSC platform to the partner with the following JSON parameters.

Request

Method	URL
POST	provided by partner during integration

Request Params	Data Type	Is Mandatory
requestType	string	yes
transactionId	long	yes
msisdn	string	yes
shortNumber	string	yes
operatorId	byte	yes
smsText	string	yes
smsDateTime	datetime	yes
publicKey	string	yes
hashedKey	string	yes

requestType

This is the type of request: Possible values are: "sms" or "report".
When *requestType* is "sms" that means this is a sms from subscriber, when *requestType* is "report" that means this is a sms delivery status request

transactionId

This is the transactionId of subscriber’s request. In case of "report" *requestType* this will be the transactionId of request sent by partner via SMS MT API.

msisdn

This is the number of subscriber in 994yyxxxxxxx format

shortNumber

This is a short number to which subscriber sent a SMS

operatorId

This is the id of subscriber’s operator. Possible values are: 1-Azercell, 2-Bakcell, 3-Azerfon

smsText

This is MO text message sent by subscriber. In case of "report" *requestType* this field will contain the sms delivery status and description. SMS text format is *statusCode: statusDescription* (f.e. 2: DELIVERED).
Possible SMPP Delivery Statuses and Descriptions is provided in the end of this section.

smsDateTime

This is DateTime when text message was received by DSC

publicKey

This is the randomly generated string Key by DSC for each request.

hashedKey

This is the key which generated with below scenario.

hashedKey=MD5(transactionId+publicKey+privateKey).

Here privateKey is the key of the partner given by DSC.

Request format:

```
{
  "requestType": requestType,
  "transactionId": transactionId,
  "msisdn": msisdn,
  "shortNumber": shortNumber,
  "operatorId": operatorId,
  "smsText": smsText,
  "smsDateTime": smsDateTime,
  "publicKey": publicKey,
  "hashedKey": hashedKey
}
```

The Password Hash gives this protocol the safety within the web. Being a one-time password impossible to reuse it or create new ones, this element will authenticate DSC into partner platform while assuring integrity of all the communication.

DSC waits response in JSON format, with “resultCode” and “resultDescription” parameters. Value **resultCode=0** will mean that SMS was successfully received by partner, otherwise SMS will be re-tried to be sent again.

If timeouts are detected by the system then process will be repeated until SMS will be successfully received and replied by partner.

An example request:

```
{
    "requestType": "sms",
    "transactionId": 987654321,
    "msisdn": 994501234567,
    "shortNumber": 1111,
    "operatorId": 1,
    "smsText": "test sms",
    "smsDateTime": "2016-07-22 10:00:00",
    "publicKey": "test1234",
    "hashedKey": MD5("987654321test1234"+privateKey)
}
```

Response

Response will be a JSON object containing the Result Code, Result Description fields. **resultCode=0** will mean that SMS was successfully received by partner, otherwise SMS will be re-tried to be sent again.

Response format:

```
{
    "resultCode": resultCode,
    "resultDescription": resultDescription,
}
```

An example response:

```
{
    "resultCode": 0,
    "resultDescription": "success"

}
```

That means SMS was successfully received by partner side. Any other response (timeout also) will be considered as SMS not received by Partner, and will be sent again.

SMPP Delivery Statuses and Descriptions

Status Code	Status Description
0	SCHEDULED
1	ENROUTE
2	DELIVERED
3	EXPIRED
4	DELETED
5	UNDELIVERABLE
6	ACCEPTED
7	UNKNOWN
8	REJECTED
9	SKIPPED

2. MT (Mobile Terminated) Transactions API

This API is used to send messages to the mobile subscriber. API is a HTTP POST method call made to the DSC platform by partner with the following JSON parameters. The API supports sending multiple MTs within a single HTTP method call.

Request

Method	URL
POST	provided by DSC during integration

Request Params	Data Type	Is Mandatory

transactionId	long	yes
requestId	long	yes
msisdn	string	yes
shortNumber	string	yes
operatorId	byte	no
priority	int	yes
smsText	string	yes
publicKey	string	yes
hashedKey	string	yes

transactionId

This is unique numeric Id identifying each transaction, should not exceed big int limit.

requestId

If requested MT message is response for subscriber's MO message, then this field will contain transcationId of MO message which was sent by MO API. If it is not response message but subscription message then this field will contain "-1" value.

msisdn

This is the number of subscriber in 994yyxxxxxxx format

shortNumber

This is the short number to which subscriber sent a SMS

operatorId

This is the id of subscriber's operator and is optional. When not provided we will check whether this number is MNP or not and identify the operator of provided number. But this might increase your request/response duration. Therefore it is suggested to provide it if you know the operator of number. If provided, possible values are: 1-Azercell, 2-Bakcell, 3-Azerfon

priority

This is for ordering SMSs by their priorities, informative SMS which has highest priorities (0) will be sent immediately (if there is no waiting informative message in queue), the rest type (priority>0) of SMSs will be sent after all informative SMSs were sent. There is no time limit for informative SMSs (for example: subscriber can request some service at 04:53 and will be informed immediately with some informative message), the rest types of SMS have time limitation between 09:00 – 21:00 (SMS that must be sent due to subscription will be sent only between this time interval).

smsText

This is the text message which have to be sent to subscriber. Should not be exceeded 480 symbols limit.

publicKey

This is the publicKey of partner and can change for each request.

hashedKey

This is the hashedKey of partner which generated with below scenario.

hashedKey=MD5(transactionId+publicKey+privateKey).

Here privateKey is the key of the partner given by DSC.

Request format:

```
{
    "smsRecords": [smsRecord, smsRecord, smsRecord,… etc.]
}
```

smsRecord format:

```
{
    "transactionId": transactionId,
    "requestId": requestId,
    "msisdn": msisdn,
    "shortNumber": shortNumber,
    "operatorId": operatorId,
    "smsText": smsText,
    "priority": priority,
    "publicKey": publicKey,
    "hashedKey": hashedKey
}
```

An example request 1 (with operator):

```
{
    "smsRecords":
    [
        {
           "transactionId": 987654321,
            "requestId": -1,
            "msisdn": 994501234567,
            "shortNumber": 1111,
            "operatorId": 1,
            "smsText": "test sms",
            "priority": 0,
            "publicKey": "test1234",
            "hashedKey": MD5("987654321test1234"+privateKey)
        }
    ]
}
```

An example request 2 (one with operator, one without):

```
{
    "smsRecords":
    [
        {
           "transactionId": 987654321,
            "requestId": -1,
            "msisdn": 994501234567,
            "shortNumber": 1111,
            "operatorId": 1,
            "smsText": "test sms1",
            "priority": 0,
            "publicKey": "test1234",
            "hashedKey": MD5("987654321test1234"+privateKey)
        },
        {
           "transactionId": 649731258,
            "requestId": -1,
            "msisdn": 9945017654321,
            "shortNumber": 1111,
            "smsText": "test sms2",
            "priority": 1,
            "publicKey": "test1235",
            "hashedKey": MD5("649731258test1235"+privateKey)
        }
    ]
}
```

Response

Response will be a JSON object containing the Transaction Id, Result Code and Result Description fields. Possible Result Codes and Result Descriptions is provided in the end of this section.

Response format:

```
{
    "responses":
    [
        {
            "transactionId": transactionId,
            "resultCode": resultCode,
            "resultDescription": resultDescription
        }
    ]
}
```

An example response 1:

```
{
    "responses":
    [
        {
            "transactionId": 987654321,
             "resultCode": 0,
            "resultDescription": “success”
        }
    ]
}
```

An example response 2:

```
{
    "responses":
    [
        {
            "transactionId": 987654321,
             "resultCode": 0,
            "resultDescription": “success”
        },
        {
            "transactionId": 649731258,
            "resultCode": -13,
            "resultDescription": “invalid sms text”
        }
    ]
}
```

Result Codes and Descriptions

Result Code	Result Description
0	success
-1	general error
-2	invalid msisdn
-3	invalid operatorId
-4	invalid partner IP
-7	invalid requestId
-9	invalid json body
-10	invalid key
-11	invalid transactionId
-12	invalid short number
-13	invalid sms text
-14	invalid priority
-25	duplicate transactionId
-26	sms limit for partner exceeded