Micro Payment

Version	Date	Author	Description
1.0	16-Apr-2018	DSC	Initial Version

APIs

Intended for the partners which we have VPN connection.

1. Subscriber Initialize

We initialize the charging process, if provided parameters with URL are true. In case of true we send the access code to the user which he/she use it in your portal and you must send that code back to us to get the charging initialized.

Request

Method	URL
GET	http://**host:port**/initialize?partnerId=partnerId&serviceId=serviceId&msisdn=msisdn&operatorId=operatorId&amount=amount

Host and **port** will be provided during integration

Request Params	Data Type	Is Mandatory
partnerId	string	yes
serviceId	string	yes
msisdn	string	yes
operatorId	byte	no
amount	int	yes

partnerId

This is the id of partner given by DSC

serviceId

This is the id of partner’s service given by DSC

msisdn

This is the number of subscriber in 994yyxxxxxxx format

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. However, 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, and 3-Azerfon

amount

This is the amount of charging in kopecks

An example request (with operator): http://host:port/initialize?partnerId=dscpartner&serviceId=dscservice&msisdn=994501234567&operatorId=1&amount=3

An example request (without operator): http://host:port/initialize?partnerId=dscpartner&serviceId=dscservice&msisdn=994501234567&amount=3

Response

Response will be a JSON object containing the Result Code, Result Description.

Response format:

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

An example response:

```
{
   "resultCode": 1,
   "resultDescription": "success"
}
```

Result Codes and Descriptions specified in the end of this section.

Result Codes and Descriptions

Result Code	Result Description
0	success
-1	general error
-2	invalid msisdn
-3	invalid operatorId
-4	invalid partner IP
-5	invalid partnerId
-6	invalid serviceId

-7	invalid amount

2. Subscriber Charge

Charging of the subscriber for specified amount. In this step, we check that service's type. If service is subscription-based then we save the user's information in our database and charge automatically in next subscription date. Charging depends on services' fee and applied automatically without any user interaction. Please bear the followings in mind.

- You must provide us the entire list of subscription-based services in advanced.
- Users are charged in charge request, it does not depend on service is subscription-based or not. If the user has already subscribed to that service then he/she won't be charged and appropriate message returned to the partner.

Request

Method	**URL**
GET	http://**host:port**/charge?partnerId=partnerId&serviceId=serviceId&msisdn=msisdn&operatorId=operatorId&amount=amount&accesscode=accesscode

Host and **port** will be provided during integration

Request Params	**Data Type**	**Is Mandatory**
partnerId	string	yes
serviceId	string	yes
msisdn	string	yes
operatorId	byte	no
amount	int	yes
accesscode	int	yes

partnerId

This is the id of partner given by DSC

serviceId

This is the id of partner's service given by DSC

msisdn

This is the number of subscriber in 994yyxxxxxxx format

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. Nevertheless, 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, and 3-Azerfon

amount

This is the amount of charging in kopecks

accesscode

This is the six digits code, which we send to user to ensure that charging is trusted. If access code is right then we charge the user.

An example request (with operator): http://host:port/charge?partnerId=dscpartner&serviceId=dscservice&msisdn=994501234567&operatorId=1&amount=3&accesscode=123456

An example request (without operator): http://host:port/charge?partnerId=dscpartner&serviceId=dscservice&msisdn=994501234567&amount=3&accesscode=123456

Response

Response will be a JSON object containing the Result Code, Result Description.

Response format:

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

An example response:

```
{
    "resultCode": 1,
    "resultDescription": "success"
}
```

Result Codes and Descriptions specified in the end of this section.

Result Codes and Descriptions

Result Code	Result Description
0	success
-10	insufficient balance
-1	general error
-2	invalid msisdn
-3	invalid operatorId
-4	invalid partner IP
-5	invalid partnerId
-6	invalid serviceId
-7	invalid amount
-8	invalid access code
-9	access code has expired
-11	already charged
-16	already subscribed

3. Stop Subscription

We also provide a stop subscription functionality and user can request to stop the subscription to the service he/she wants.

Request

Method	URL
GET	http://**host:port**/stopSubscription?partnerId=partnerId&serviceId=serviceId&msisdn=msisdn&operatorId=operatorId

Host and **port** will be provided during integration

Request Params	Data Type	Is Mandatory
partnerId	string	yes
serviceId	string	yes
msisdn	string	yes
operatorId	byte	no

partnerId

This is the id of partner given by DSC

serviceId

This is the id of partner's service given by DSC

msisdn

This is the number of subscriber in 994yyxxxxxxx format

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. However, 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, and 3-Azerfon

An example request (with operator): http://host:port/stopSubscription?partnerId=dscpartner&serviceId=dscservice&msisdn=994501234567&operatorId=1

An example request (without operator): http://host:port/stopSubscription?partnerId=dscpartner&serviceId=dscservice&msisdn=994501234567

Response

Response will be a JSON object containing the Result Code, Result Description.

Response format:

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

An example response:

```
{
   "resultCode": 1,
   "resultDescription": "success"
}
```

Result Codes and Descriptions specified in the end of this section.

Result Codes and Descriptions

Result Code	Result Description
0	success
-1	general error
-2	invalid msisdn
-3	invalid operatorId
-4	invalid partner IP
-5	invalid partnerId
-6	invalid serviceId
-12	user has no subscription to this service

4. Send message to Subscriber

You can send a message to any service's subscriber in order inform them about new features and changes.

Request

Method	URL
POST	http://**host:port**/sendMessage

Host and **port** will be provided during integration

```
{
   "partnerId":"partnerId",
   "serviceId":"serviceId",
   "msisdn":"msisdn",
   "message":"message",
   "operatorId": operatorId
}
```

partnerId

This is the id of partner given by DSC

serviceId

This is the id of partner’s service given by DSC

msisdn

This is the number of subscriber in 994yyxxxxxxx format

message

This is message text which you want to send and must not exceed 480 symbol.

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. However, 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, and 3-Azerfon

An example request (with operator):


```
{
   "partnerId":" dscpartner",
   "serviceId":" dscservice",
   "msisdn":"994501234567",
   "message":" hello world",
   "operatorId": 1
}
```


An example request (without operator):


```
{
   "partnerId":" dscpartner",
   "serviceId":" dscservice",
   "msisdn":"994501234567",
   "message":" hello world"
}
```


Response

Response will be a JSON object containing the Result Code, Result Description.

Response format:


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


An example response:


```
{
```

```
    "resultCode": 1,
    "resultDescription": "success"
}
```

Result Codes and Descriptions specified in the end of this section.

Result Codes and Descriptions

Result Code	Result Description
0	success
-1	general error
-2	invalid msisdn
-3	invalid operatorId
-4	invalid partner IP
-5	invalid partnerId
-6	invalid serviceId
-13	service is not subscription-based
-12	user has no subscription to this service
-15	message exceeds 480 character length

5. Sending Callback to Partner

This shows how partner gets notified through callback URL. The partner must provide the callback URL as following.

Request

Method	URL
GET	http://**host:port**/dscCallback?serviceId=serviceId&msisdn=msisdn&type=type&description=description

The partner will provide host and port. URL's pathname and query strings' name depend on partner. Nevertheless, count of the query strings must be same as mentioned above.

Request Params	Data Type	Is Mandatory
serviceId	string	yes
msisdn	string	yes
type	byte	yes
description	string	yes

serviceId

This is the id of partner's service given by DSC

msisdn

This is the number of subscriber in 994yyxxxxxxx format

type

This is the type of operation and it contains number, that each describes appropriate description.

- 0 = successful
- 1 = successful_renewal_subscription
- 2 = subscription_calcelled_by_user
- 3 = subscription_cancelled_by_dsc
- 4 = partial_debt_success
- 5 = debt_failure
- 6 = trial_period_ended
- 7 = OneShotSuccess
- 8 = IsNotSubscribedToMainService
- 9 = AlreadySubscribed
- 10 = WrongKeyword
- 11 = NotSubscribed
- 12 = NoMoney
- 13 = TrialSubscribed

If we cannot charge the user then prolong his/her subscription. This prolonging period is defined by DSC. During this period, we try to charge the user every day. In case of successful operation, we returned the partner “type 1”, otherwise “type 3”

description

This is the result of operation. For example, in case of renewal of subscription process, it contains specific message for that process.

http://host:port/getNotification?serviceId=dscservice&msisdn=994501234567&description=successful_renewal_subscription

6. Stop Subscription via SMS

User can stop his/her subscription to the particular services via SMS channel. When he/she does, we notify the partner that, what action he/she did.

Request

He / She sends a stop keyword to the short number. For example, He / She sends “stop{servicename}” to “XXXX” and we check if the user has the subscription then unsubscription process get initialized automatically.

- {servicename} is placeholder for the actual service name of partner.

7. Get Subscription Info

You can get current subscription infos via this method

Request

http://host:port/subscriptionInfo?partnerId=partner&serviceId=dscservice

partnerId

This is the id of partner given by DSC

serviceId

This is the id of partner’s service given by DSC

Response

Response will be a JSON object containing the Result Code, Result Description.

Response format:


```
{
   "resultCode": 0,
   "resultDescription": “success”,
   "activeSubscriptionsCount": 1,
   "stoppedSubscriptionsCount": 2,
   "confirmedPinMessagesCount": 4,
   "unconfirmedPinMessagesCount": 3,
}
```