



Strex Payment Platform API Implementation Guide

Version 1.79

1 Revisions

Version	Description
0.1	First draft
0.2	Added message flows and examples
0.3	Added Java Sample code, definitions table and other corrections
0.4	Minor changes on status codes
0.41	Updated urls
0.5	Change in Sell parameters, added delivery notification and reverse + other minor changes
0.51	Added format table + other minor changes
0.52	Minor changes.
0.6	Added exceptions to service codes + other minor changes.
0.7	Updated Business Models + other minor changes
0.8	Updated structure of response codes + added new. Deleted chapter 9.3.1 and 9.4.1
0.9	Updated after changes in registration process, update of Service Codes and general changes.
0.91	Added chapter 8.4.1 retry schedule, changed description of trans_ext_reference and merchant_transid in sell request, changed type from decimal to int for amount in sell request.
0.92	Added information about DN for SendSMS function.
0.93	Removed information SMS's from Strex, added information about mno_ref parameter in sell
0.94	Changed 8.4.1 retry schedule, added description of age verification
0.95	Added information about usage limits and introducing new error codes 10005-10008
0.96	Added new error code 10009 and suggested feedback to end user for selected error codes.
0.961	Added service code update
0.962	Added capacity restrictions information
0.963	Added information about USSD authorization of payment transactions
0.97	Added result codes, new parameter in sell and information about new registration functionality as well as changes on DN to support new registration flow. DN for SendSMS is now supported for all operators.
0.971	Added service code update
0.972	Added information in preauthorize and authorization parameter in sell.
0.973	Added status code 4054
0.974	Update preauthorize input parameter Security Level
0.975	Updated appendix G.
0.976	Added API selfreg v3 description and error codes 4100, 4052.
0.977	Added new status codes for SOAP NameSpace in chapter 9.1.1.3, new functionality for prohibiting parallel transactions in chapter 8.2.1.1, changed trans_ext_reference to o_trans_ext_ref in 8.2.5
0.978	Changed description in response of Get User Information.
1.0	Added new functionality SMS confirm. Upgraded to version 1.
1.1	Added description of generateOTP functionality.
1.2	Added description for Payment Card functionality, chapter 6.3.6 and new error codes.
1.3	Updated preauth description regarding multiple tokens pr MSISDN pr Merchant Added result codes 4141-4166. Updated default confirm_channel.
1.4	Added new message parameters for preauthorization and generateOTP,

	added statuscodes 4169,4183 and 4184 and parameter payment_source_restr in sell. USSD will be decommissioned by the MNOs sometime in 2020.
1.5	Added getuserinfoV2 and confirmSMS request, added new request parameters for preauthorize.
1.6	Added: * information about content type in smsforward and delivery notification requests. * getuserinfoV3 request * New input parameters for getuserinfoV2 * new output parameters for getuserinfoV1 * Registration widget * Appendix H Teletorg Removed USSD flows.
1.7	Changed appendix H to English. Added new error codes for bankid Added option "4" for merchant_sell_option in Sell and Security_level=4 in preauthorize.
1.71	B-number (called party) in Sell invoice parameter for Teletorg, Appendix H Premium Rate Services (Teletorg)
1.72	Added text in teletorg chapter.
1.73	Clarified message to end user in teletorg chapter.
1.74	Changed description of service code 03011
1.75	Added timeout information for preauth confirm. Added chapter 7.6 Handling of error scenarios. Added additional sell parameters for Teletorg/Premium voice services. Added information about handling_company in Sell and getSellInfo
1.76	Updated information in chapter 6.3.5 Strex registration widget
1.77	Added list of regulated services for card payments and credit agreements, Appendix I - Regulated services for card and credit agreement payments Added status code
1.78	Added check same user parameter in Sell
1.79	Updated reference in Appendix 1, charity

2 Definitions

Attribute	Description
Agent	All users, merchants, subscribers etc. are referred to as agents in the Strex Payment Platform.
Easy USSD Registration	End User registration process using USSD for End User dialogue
Easy Web Registration	End User registration process using web for End User dialogue
End User	A person using a mobile handset with a subscription, a.k.a. subscriber or customer
Full USSD Registration	End User registration process using USSD for End User dialogue
Full Web Registration	End User registration process using web for End User dialogue
Initiator	The initiator of an action or request. This is normally the merchant identified by the merchant username.

MNO	Mobile Network Operator, e.g. Telenor
MO-SMS	Mobile Originated SMS. The message sent from the end users handset
MT-SMS	Mobile Terminated SMS. The message sent to the end users handset
OTP	One Time Password
Recipient	The recipient of an action or request triggered by the initiator. Normally the end user represented by the MSISDN.
Strex Direct Payment	Strex product name for billing of End Users. Previously known as direct charge
Strex SMS Payment	Strex product name for billing of End Users including SMS to the end user. Previously known as Premium SMS

3 Formats

Attribute	Description
MSISDN	Is countrycode+msisdn, i.e. '4798765432'. Only Norwegian msisdns supported.
Merchant user id	"mer_merchantname"

4 Contents

1	Revisions	2
2	Definitions	3
3	Formats.....	4
5	Introduction.....	7
5.1	Audience.....	7
6	Overview	7
6.1	Strex SMS Payment	8
6.2	Strex Direct Payment	9
6.3	Registration of End Users.....	10
6.3.1	Easy Web Registration.....	10
6.3.2	Full Web Registration.....	11
6.3.3	Easy Web Registration triggered by merchants.....	11
6.3.4	Registration when exceeding amount limits	11
6.3.5	Strex registration widget	12
6.4	Age and amount limits	12
7	Getting Started.....	12
7.1	Merchant Setup.....	12
7.2	SOAP Architecture.....	14
7.3	Session management.....	15
7.4	Session Usage	15
7.5	Testing.....	16

7.6	Handling of error scenarios.....	16
8	SOAP API Messages.....	17
8.1	Incoming Generic messages	17
8.1.1	Createsession.....	17
8.1.2	Login.....	18
8.1.3	Parameters – not in use.....	19
8.2	Incoming Business Messages.....	20
8.2.1	Parallell transactions.....	20
8.2.2	Sell	20
8.2.3	Reverse transaction	30
8.2.4	SendSMS.....	32
8.2.5	Get Sell Information	33
8.2.6	Get User Information.....	35
8.2.7	Get User Information v2	37
8.2.8	Get User Information v3.....	40
8.2.9	preAuthorize.....	43
8.2.10	selfregisterV3.....	48
8.2.11	generateOTP.....	49
8.2.12	confirmSMS.....	52
8.3	Outgoing Messages.....	54
8.3.1	Retry schedule.....	54
8.3.2	SMS Forward.....	54
8.3.3	Delivery Notification.....	56
9	Appendixes.....	59
9.1	Appendix A Result Codes.....	59
9.1.1	Strex Payment Platform API Result Codes	59
9.1.2	Delivery Notification Result Codes (Strex → Merchant)	75
9.2	Appendix B Pin calculation.....	77
9.3	Appendix C Service Codes.....	78
9.4	Appendix D Business Models.....	83
9.5	Appendix E WSDL.....	83
9.6	Appendix F Java Sample code	83
9.7	Appendix G SMS confirm screenshot examples	84
9.7.1	Single payments.....	84
9.7.2	Preauthorized and recurring payments	85
9.8	Appendix H Premium Rate Services (Teletorg)	86
9.8.1	Brief description of the api requests.....	86

9.8.2	Current amount limits (subject to change)	86
9.9	Appendix I - Regulated services for card and credit agreement payments	87

5 Introduction

The service enables the Merchant to charge the End User and send the End user a SMS message. Also available are forwarding of SMS messages sent from the mobile terminal of the End User (MO) to 4-digit short numbers of the Merchant/Aggregator as well as Delivery Notifications of SMS messages (MT) sent from 4-digit short numbers of the Merchant/Aggregator.

The service is made available through a single web service API for charging and send SMS, as well as an outgoing webservice API for forwarding SMS and Delivery Notifications. The outgoing API must be implemented on the Aggregators/Merchants end.

The implementation guide (this document) describes the necessary technical details to implement the functionality described in chapter 6.

5.1 Audience

This document is intended for developers and architects implementing the payment service, and contains the necessary technical details to integrate with the Strex payment service.

Please also refer to the Strex Merchant agreement including Product Description and Merchant Code of Conduct.

6 Overview

Strex payment service supports this functionality:

- Strex Direct Payment; Charge of end users ("Sell")
- Strex SMS Payment; Charge of end users with SMS sent to end users ("Sell")
- Sending mobile terminated SMS (MT) to end users
- Forwarding mobile originated SMS (MO) messages to the Strex Partner/merchant
- Reversal of transactions
- Delivery notifications to Strex Partner/merchant of all submitted MT messages (in sell and sendsms)
- Registration of End Users during Sell
- Preauthorization of Sell transactions
- API to get information about previously executed sell transactions
- API to get high-level information about End Users

Strex payment service supports a **total of 3 request pr second** pr Merchant (not including Forwarding mobile originated SMS and Delivery notifications). Capacity exceeding this threshold requires a specific agreement with Strex.

6.1 Strex SMS Payment

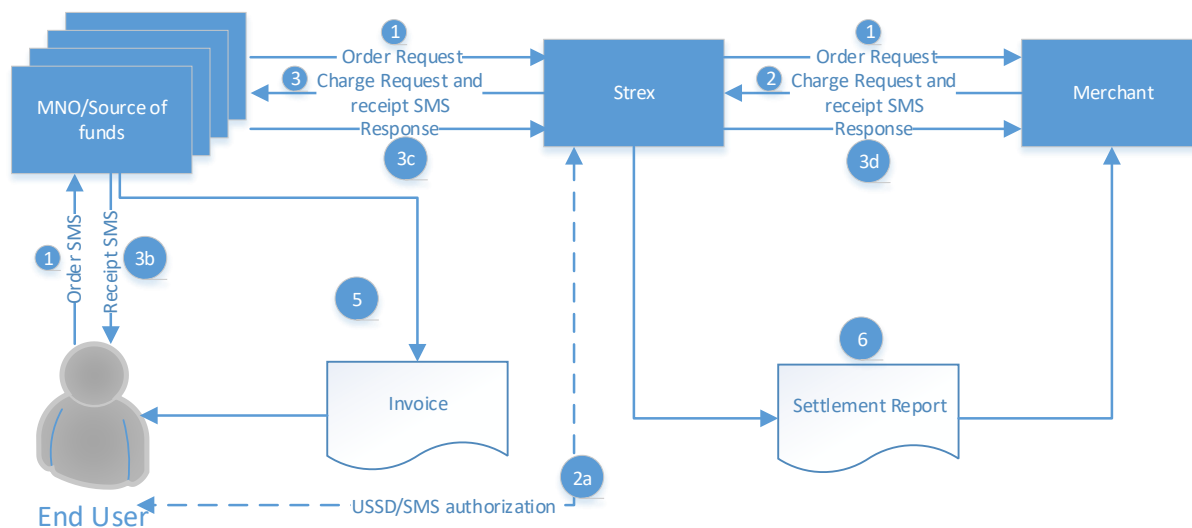


Figure 1 Strex SMS payment

The payment process:

1. The End user orders a product from the Merchant by sending an SMS to the four-digit short number of the Merchant. The SMS is forwarded to the merchant by the MNO and Strex.
2. The Merchant sends a Sell request including SMS Message to Strex (possibly via an Aggregator).
 - a. If applicable according to sell/End User settings, Strex sends a SMS authorization request to the mobile terminal of the End User.
 - b. If applicable, the End User authorizes the payment transactions entering "JA" or "OK" in the SMS application on his mobile terminal. The time out of the SMS authorization timeout is set to 1 minute.
 - c. The payment transaction can also be confirmed by using an OTP, please refer to [generateOTP](#) chapter
3. Strex executes the charge request with the End Users selected Source of Fund (Currently available MNO Bill, Strex invoice and Payment Card). The Source of Fund provider checks that the subscriber has sufficient funds for the payment, makes a charge, returns a result report to Strex which send the SMS message to the End user. Strex send a result report to the Merchant.
4. The Merchant approves the payment and delivers the product ordered by the End User. The product ordered may be included in the SMS message delivered to the End User or delivered separately in another channel, e.g. by post if physical goods.

6.2 Strex Direct Payment

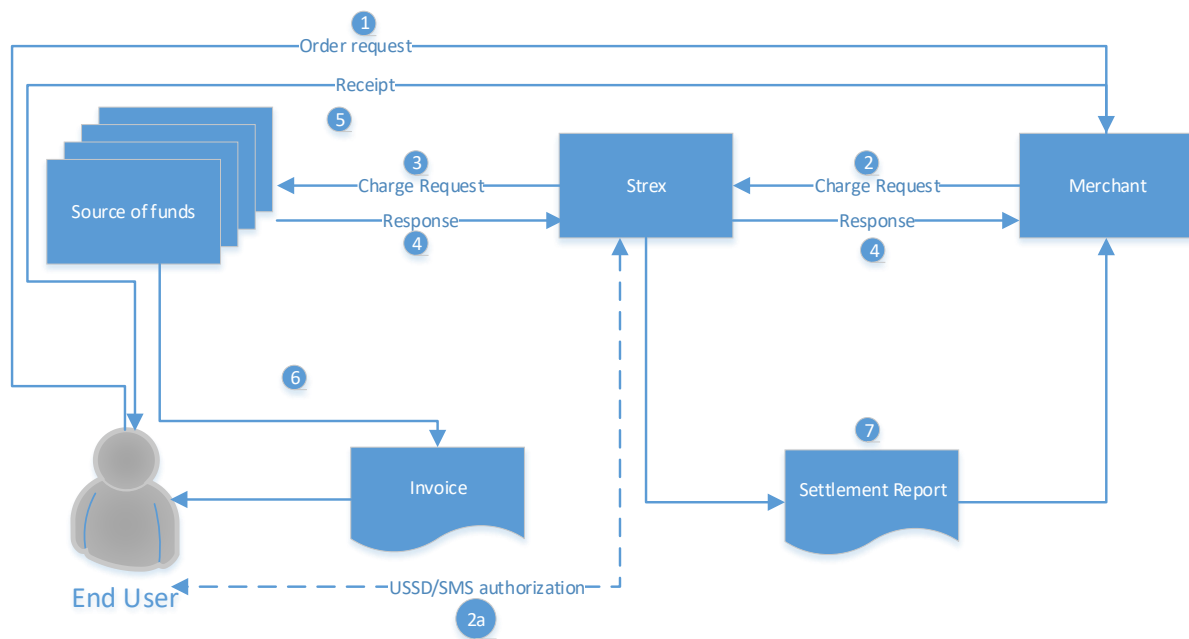


Figure 2 Strex Payment

The payment process:

1. The End user orders a product from the Merchant in a phone app or on a webpage or other means of communication.
2. The Merchant sends a Sell request to Strex. (Possibly via an Aggregator).
 - a. If applicable according to sell/End User settings, Strex sends a SMS authorization request to the mobile terminal of the End User.
 - b. If applicable, the End User authorizes the payment transactions entering "JA" in the SMS application on his mobile terminal, or with BankID on mobile. The time out of the SMS authorization timeout is to 1 minute.
 - c. The payment transaction can also be confirmed by using an OTP, please refer to [generateOTP](#) chapter
3. Strex executes charge request with the End Users selected Source of Fund (Currently available MNO Bill, Strex invoice and Payment Card). The Source of Fund provider checks that the End User has sufficient funds for the payment and makes a charge.
4. Source of Fund provider returns a result report to Strex. Strex returns a result report to the Merchant (Aggregator).
5. The Merchant processes the payment, and if successful, informs the End User about the Sell and delivers the product ordered to the End User.

6.3 Registration of End Users

Some payment transactions require that the End User is registered with Strex (i.e. “licensed purchases”), this is governed by the service code set in Sell request. All End Users can pay for good and services that don’t require a registration with Strex. Please refer to [Appendix C Service Codes](#) for an overview of the service codes and necessary registration.

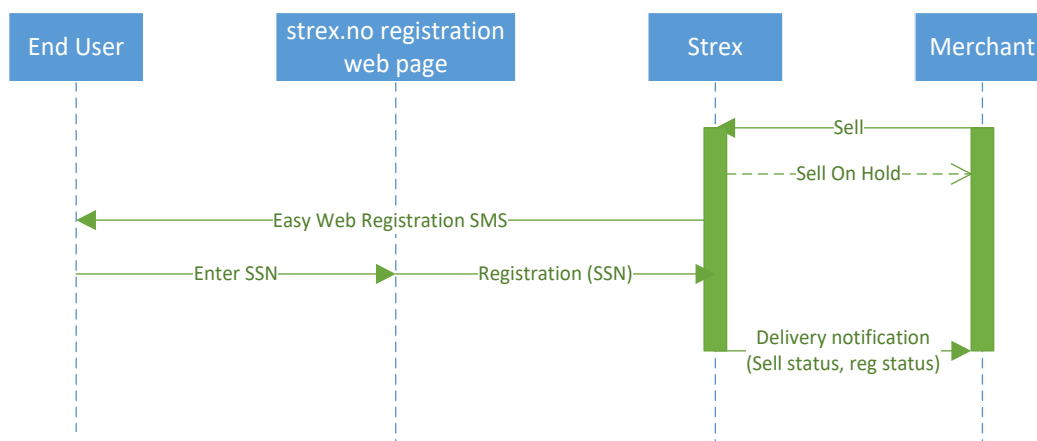
Several registration scenarios can be triggered by sell, depending on characteristics of the End User, Merchant, Sell transaction and other:

- Easy Web Registration on strex.no/min-side/
- Full Web Registration on strex.no/min-side/registrering

Easy Web Registration/Full Web Registration is the default method for registration.

6.3.1 Easy Web Registration

Easy web registration can be triggered when sufficient information on the End User is available.



1. Merchant initiates a sell transaction for a licensed service code
2. Strex determine that the End User is unregistered and that a Easy Web Registration SMS must be sent to the End User, and responds to the Sell transaction with statuscode=4092 “Sell put on hold”
 - a. Note: default timeout for the sell to be “on hold” is set to 5 min. Timeout can be overruled by setting the “timeout” parameter in sell request. Please refer to [Sell request chapter](#) for information about the valid parameter settings.
 - b. The merchant can change the text in the registration SMS by using the merchant_reg_smstext parameter in sell, see sell request chapter.
3. End User opens the link to the registration page in the SMS, accepts terms&conditions and enters the SSN.
4. Strex receive the registration request and respond to the merchant with status of the sell through a delivery notification request. If registration is completed within the timeout the sell is also completed successfully and the End User is charged for the purchase.

6.3.2 Full Web Registration

In case none of the previous registration scenario applies the unregistered End User can receive the following (or similar) SMS when initiating a payment:

«Hei! Vi beklager, men vi kunne ikke gjennomføre betalingen fordi vi mangler informasjon om deg. Du må bestille varen/tjenesten på nytt etter å ha registrert deg her: strex.no/reg. Beløpet vil belastes din mobilfaktura. Hilsen Strex – mobiloperatørenes betalingstjeneste.»

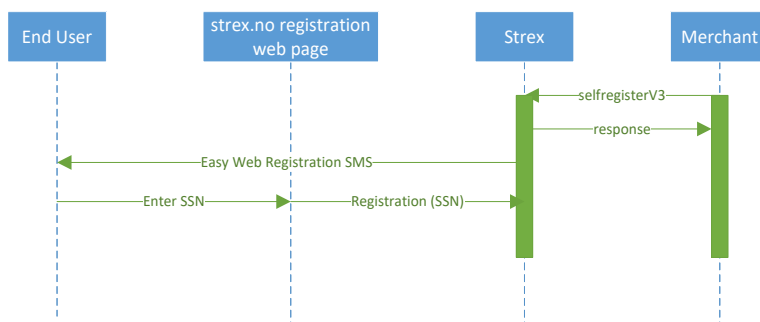
SMS text translated into English (for explanatory purposes only): “Hello, we are sorry, but we cannot process the payment because we lack some information about you. You must order the good or service once more after registering here: strex.no/min-side/registrering. The payment will be charged to your mobile phone bill. Best regards the mobile operators common payment service – Strex.”

The payment will not be processed. A new payment can be initiated when the End User has registered with Strex.

Full web registration page is also available “offline” (i.e. not triggered by sell).

6.3.3 Easy Web Registration triggered by merchants

Easy web registration can be triggered by merchants by sending the selfregisterV3 request. The request will send a SMS to the End User with a link to the Strex Easy Web Registration site. If Easy Web Registration is not available for this End User, he will be redirected to the Full Web Registration site.



1. Merchant initiates the selfregisterV3 request
2. Strex responds with an ACK on the request, and send the registration SMS to the End User
3. The End User opens the link to the Strex Easy Web Registration site.
4. The End User is register by entering his National ID. If unsuccessful, the End User is redirected to the Full Web Reg site where he can enter all necessary information.
5. The End User is registered. Please note that there is no response to the merchant regarding registration status to the merchant who initiated the selfregisterV3 request. The Merchant can use Get User Information request to get the registration status of the End User.

See chapter [selfregisterV3](#) for details.

6.3.4 Registration when exceeding amount limits

Amount limits are described in chapter below. An End User may register alternative Source of Funds (Strex invoice or Payment Cards) if any of the transaction limits are exceeded. When the

sell transaction exceed the amount limit, the sell will fail and strex will send a registration SMS to the End User. Strex will use the Merchant Company Name as part of the SMS text, or a Company Brand Name if preferred. Please contact Strex to add a brand name to the merchant profile.

6.3.5 Strex registration widget

This is a JavaScript widget used for adding an iframe to your page where users can register at Strex to be able to pay for services that requires registration.

Documentation of this solution can be found here:

<https://github.com/Target365/docs/blob/main/StrexSelfRegistrationWidget.pdf>

6.4 Age and amount limits

An amount limit for MNO bill is enforced by Strex for End Users as follows:

Below 18 years: max 1000 NOK per month

Over 18 years: max 5000 NOK per month

7 Getting Started

Use of the service requires a Strex Merchant agreement and the technical information form must be filled in.

7.1 Merchant Setup

Setup of all Aggregators, merchants and short number on the Strex platform follows the logic described in Figure 3 Merchant setup example below. This also applies for single merchants without Strex Partner and Strex Partners with only one merchant/short number.

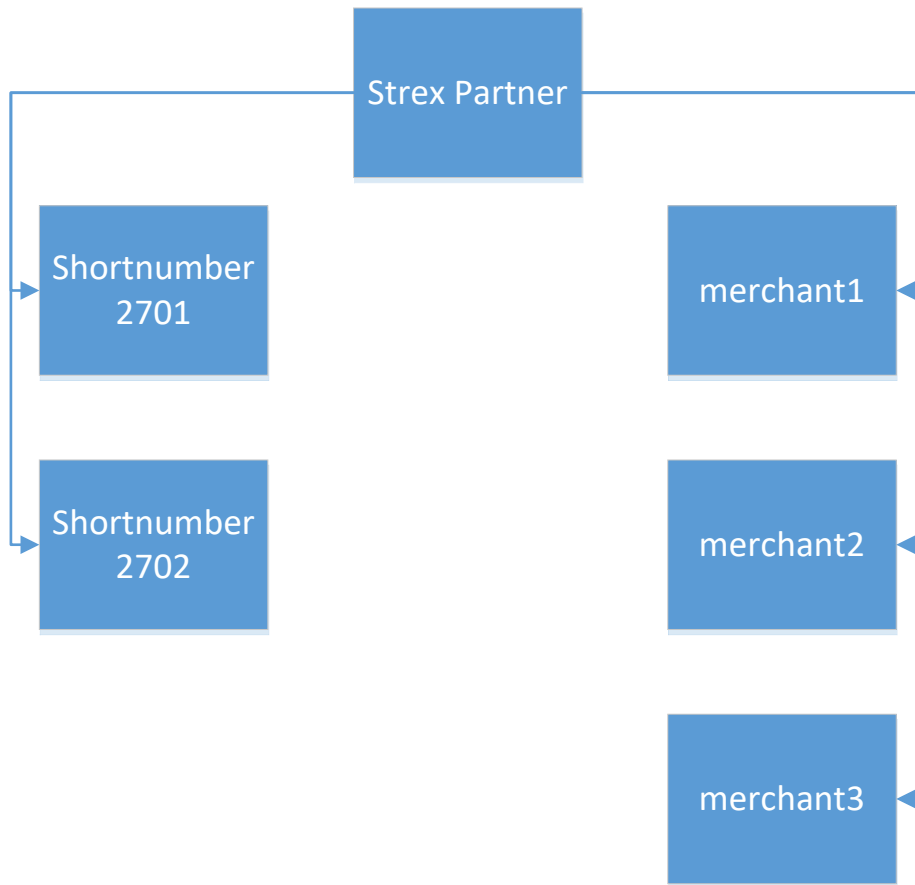


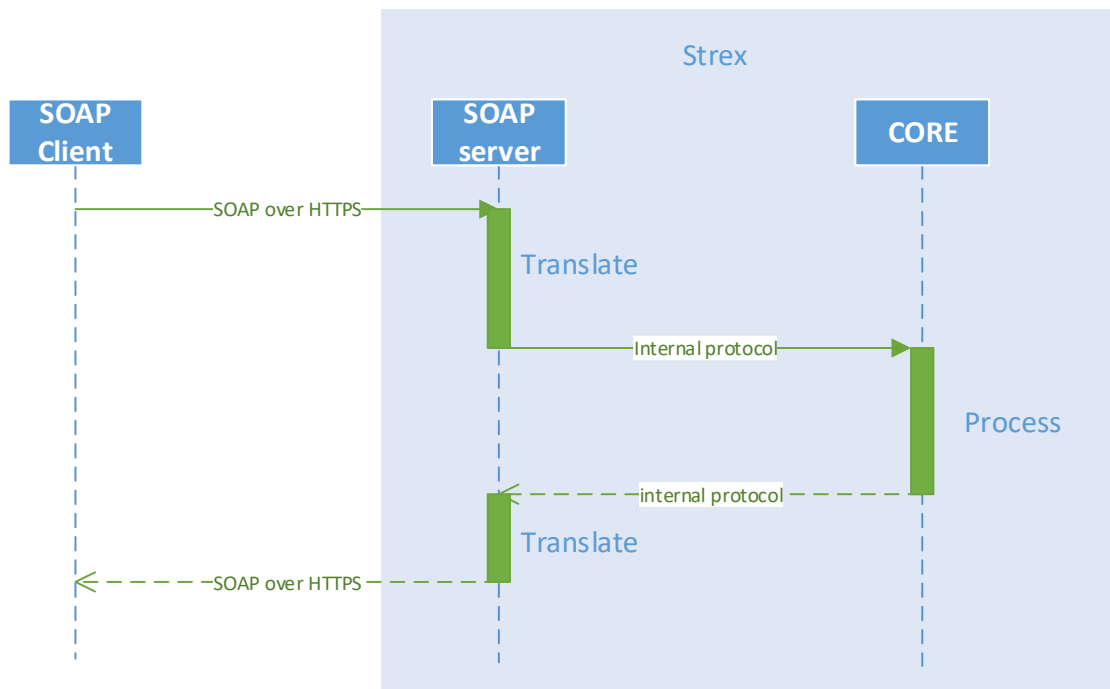
Figure 3 Merchant setup example

The following rules and restrictions apply for this setup:

- All Aggregators, Shortnumbers and Merchants are registered as separate users in the Strex platform.
- Merchants and shortnumbers are registered as “customers” of the Strex Partner
- Only Merchants can initiate transactions on the SOAP API. Aggregators and shortnumbers can not initiate any SOAP transactions.
- All merchants registered with an Strex Partner have access to all of the shortnumbers registered with the same Strex Partner. Merchants do not have access to shortnumbers registered with other Aggregators. Restrictions on the use of shortnumbers for the Strex Partner’s merchants must be enforced by the Strex Partner.
- SOAP transactions must be initiated with the username/password assigned to the merchant.
- The shortnumber reference must be included in the sell request (parameter <short_code>). This applies both for Strex SMS Payment and Strex Direct Payment. Please note that the short_code reference is used as SMS Sender Address and billing reference for Strex SMS Payment, whereas for Strex Direct Payment it is used as billing reference.

7.2 SOAP Architecture

The Web services architecture is illustrated below. Communication is SOAP over HTTPS over TCP Sockets. As is standard with the HTTPS protocol, socket 443 will be used.



The URI and host for the SOAP service are as specified below:

URI: urn:UMARKETWS

URL: the URL for accessing the soap servers:

Production:

- webservice: <https://ws.strex.no/services/umarketsc>
- wsdl: <https://ws.strex.no/services/umarketsc?wsdl>

Test:

- webservice: <http://ws-uat.strex.no/services/umarketsc>
- wsdl: <http://ws-uat.strex.no/services/umarketsc?wsdl>

Both environments can be used for end-to-end testing including SMS. However, the test environment is in use by Strex and will be unstable. Strex recommend that the test environment is used for integration testing only. Strex can provide temporary shortnumbers for both environments.

Strex use FW whitelisting of source Ips for access to the webservice endpoints. The source Ips for your production and test environment must be included in the technical order form to be whitelisted.

Wsdl is also available by contacting technicalsupport@strex.no

The SOAP server supports SOAP 1.1 and SOAP 1.2 specifications.

The soap interface broadly supports the following capabilities:

- Session and Login Management
- Access/Execute all agent management and transaction business functions

7.3 Session management

Communication shall be via SOAP over HTTPS over TCP Sockets. As is standard with the HTTPS protocol, socket 443 will be used.

HTTP Session Management encompasses web services interaction with the Strex system. Of principal importance is addressing security issues:

1. Minimizing data transmitted and shared between client and server.
2. Session theft from another Host/Server connected to the Internet.
3. Capturing and replaying of password token (i.e. eliminating password compromise, replay attack).

The Strex system will act as the session manager. Sessions shall have unique ID's generated by the manager. Each session tracks session information (i.e. user name, etc.) and shall be automatically deleted if idle (un-accessed) for a pre-defined period of time. SOAP clients shall create sessions, then authorize the session via a login. Session Creation and Login are two separate HTTP requests. Authorization will require the SOAP client to send a username and the password token. In spite of the utilization of HTTPS, the password token shall not be transmitted in the SOAP request in plaintext – the SOAP client must pass the password token through a SHA algorithm, prefixed (salted) with the session id. The net effect will be a doubly encrypted password, once using SHA digest, and again using on-the-wire SSL.

All requests that follow the initial session creation (Using the authorization after session creation) shall verify the client's IP address and any other details, whose change may represent a malicious third party attempting to defraud a session.

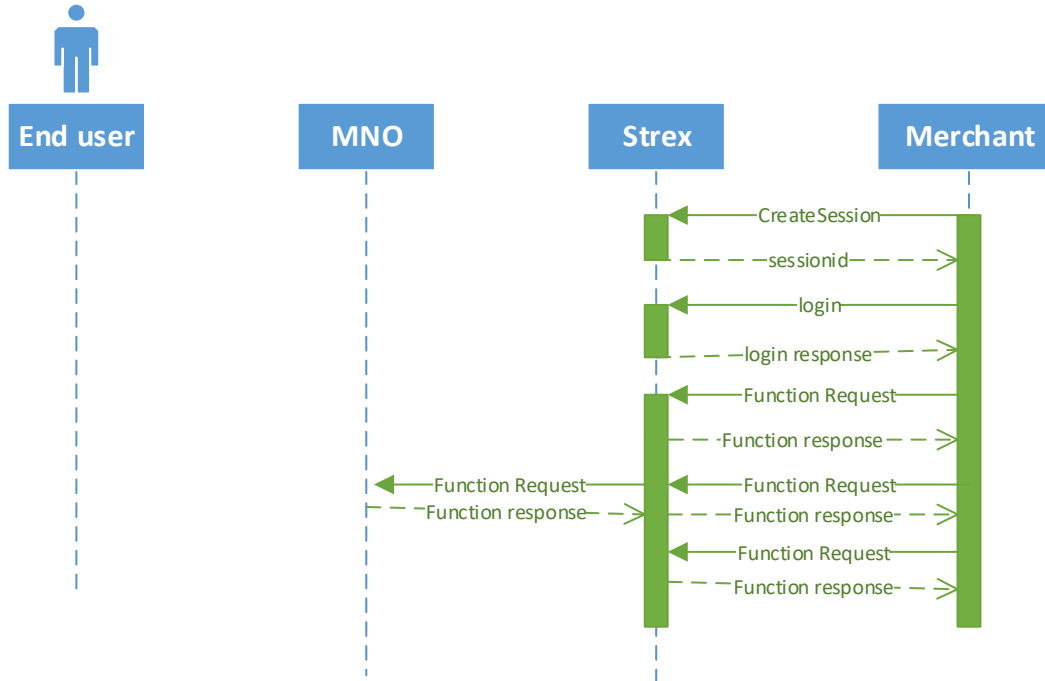
Sessions will automatically be invalidated after a preset period has passed with no activity on that session. The default value of this timeout period is 1700 seconds. There is no explicit 'logout' command. Sessions that are not used for longer than the timeout period will automatically be invalidated. There is no need for the user to explicitly log out of a session.

7.4 Session Usage

Some users may choose to generate a session and log in to it, then use that session for a large number of transactions. Others may choose to generate a fresh session and login for every single transaction. Either approach is legitimate, although the first approach is recommended for performance considerations.

Note: it is allowable to have multiple simultaneously-active sessions using the same login, this will not result in any problems.

An example message flow for one session is provided in the figure below.



7.5 Testing

Integration test should first be executed with the test environment described in [SOAP Architecture](#). Please note that test environment is used for internal purposes as well, so merchants should not expect that test environment is fully functional and bug-free. Test environment is merely an environment where your integration can be tested.

All merchant Sell transaction prior to launch of the commercial service should be reversed. Sell transactions that are not reversed can be subject to financial settlement between Strex and the Merchant.

Make sure that required functionality is tested for all mobile operators prior to launch:

- Createsession
- Login
- Sell (Strex Direct Payment, if applicable)
- Sell (Strex SMS Payment, if applicable)
- Reverse
- SendSMS (if applicable)
- forwardSMS (if applicable)
- Delivery Notification (if applicable)

7.6 Handling of error scenarios

Only transactions with resultcode=0 is regarded as successful. The most common error scenario is network problems which causes timeouts in the value chain. A timed out transaction, or a transaction where no response is given might be charged the subscriber. To avoid duplicate charging use the [Get Sell Information](#) request to check if the sell transaction was successful or

not before executing a new sell request. NB! Do also make sure result=0 is received on the Get Sell Information request as well before executing a new sell request.

8 SOAP API Messages

This chapter describes the relevant messages to use for the incoming (to Strex) and outgoing (from Strex) APIs. The examples in this chapter describe a subset of parameters for each given request for simplicity. For more detailed requests refer to the appendix.

8.1 Incoming Generic messages

8.1.1 Createsession

This command is used to create a new session. A valid session is required for all other commands.

Input Parameters

Parameter	Description	Type	Optional
-	-	-	-

Response Parameters

Parameter	Description
sessionID	Valid sessionid to be used for subsequent API requests
result	Result Code for the request. Refer to the appendices for a complete list
result_namespace	Namespace associated with the result code

Example request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:createsession/>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:createsessionResponse
xmlns="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:createsessionReturn>
        <ns3:sessionid>6GQWYC8OF8IG00D51HH</ns3:sessionid>
```

```

        <ns3:result>0</ns3:result>
        <ns3:result_namespace>session</ns3:result_namespace>
    </ns3:createsessionReturn>
</ns3:createsessionResponse>
</S:Body>
</S:Envelope>

```

8.1.2 Login

This command is used to authenticate the merchant. The biz functions require authentication in order to function. If a merchant is already logged in on the session a subsequent successful login will switch the session to the new login merchant.

Input Parameters

Parameter	Description	Optional
sessionid	Valid sessionid retrieved from createsession response	No
initiator	userID, Identifies the merchant.	No
pin	PIN or access code. String generated from password, userid and sessionid. See appendix 9.2 for details.	No
trans_ext_reference	Merchants own transaction reference	Yes
pin_type	Is set to pin_type=1.	No

Response Parameters

Parameter	Description	Optional
transid	Strex transaction id for this request.	No
trans_ext_reference	Merchant transaction reference as sent in the input parameters. Not present if not sent in the request.	Yes
result	Result Code for the request. Refer to the appendices for a complete list	No
result_namespace	Namespace associated with the result code	No

Example request

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:login>
            <urn:loginRequest>
                <!--Optional:-->
                <urn:sessionid>6GQWYC8OF8IG00D51HH</urn:sessionid>
                <urn:initiator>mer Strex as</urn:initiator>
            </urn:loginRequest>
        </urn:login>
    </soapenv:Body>
</soapenv:Envelope>

```

```

<urn:pin>87AE43FF4447EA78FCB541C618E37E900DE14D04</urn:pin>
  <urn:pin_type>1</urn:pin_type>
</urn:loginRequest>
</urn:login>
</soapenv:Body>
</soapenv:Envelope>

```

Example response

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:loginResponse
xmlns="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:loginReturn>
        <ns3:transid>714806</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
      </ns3:loginReturn>
    </ns3:loginResponse>
  </S:Body>
</S:Envelope>

```

8.1.3 Parameters – not in use

The ws api has some default parameters that are not in use, described in the table below. These parameters must be omitted in the request.

Not in use Input Parameters

Parameter	Description	Type	Optional
extra_trans_data	Not in use	KeyValuePair	yes
wait	Not in use	boolean	yes
schedule	Not in use	String	yes
schedfreq	Not in use	int	yes
ext_usedefault	Not in use	String	yes
ext_source	Not in use	String	yes
ext_details	Not in use	String	yes
ext_src_method	Not in use	String	yes
ext_couponid	Not in use	String	yes
suppress_confirm	Not in use	Boolean	yes
suppress_notify_trans	Not in use	Boolean	yes
type	Not in use		Yes
recipientEncrypted	Not in use	Boolean	Yes
ip_address	Not in use	String(255)	Yes
ip_header	Not in use.	String(255)	Yes

Response parameters

Parameter	Description	Type	Optional
Alias	Not supported	String(255)	Yes
Encrypted IP address	Not supported.	String(255)	Yes
Decrypted IP address	Not supported.	String(255)	Yes

8.2 Incoming Business Messages

This chapter describes the business requests and responses and relevant business parameters.

Note: CreateSession and login can be omitted if the merchant is already logged in with an active session on the correct userid.

8.2.1 Parallell transactions

Parallel Sell and reverse transactions for the same MSISDN can create problems in transaction handling and will often result in increased response times for the merchant. A limitation of 2 seconds between each transaction per msisdn have been implemented to prevent this scenario (i.e. the second sell/reverse transaction must be sent at least 2 seconds after the first for one msisdn.) Transactions sent within the 2 sec limitation will be terminated and status code 70.

8.2.2 Sell

The Sell function represents a payment for goods or services and allows a merchant (via merchant userid and a ShortNumber) to charge the end users mobile subscription bill as payment. Two options are available:

- Strex Direct Payment; Charge end users mobile subscription bill
- Strex SMS Payment; Charge End users mobile subscription bill and send an MT SMS to the End User in the same operation

8.2.2.1 Sell confirmation

Several End User confirmation options are currently available and can be set in sell request:

- No confirmation
- SMS Confirmation
- One Time Password
- BankID on mobile

Input Parameters

Parameter	Description	Type	Optional
sessionid	Valid sessionid retrieved from createsession response	-	No
amount	The amount of this transaction. Amount should be in 1/100 NOK (e.g. 1 NOK = 100 Øre).	Int	No
recipient	The mobile subscriber (End User) who initiates	String(20)	No

	the payment.		
smsc_id	The SMSC that this request was received from.	String(255)	Yes
mno_ref	The MNO to which the End User belongs to. One of [<telenor><netcom><networknorway><ice>] or empty. Strex will do a lookup in NRDB for the msisdn if mmoref is empty/missing. Strex recommends to not use this parameter. If mno_ref is misspelled, or an incorrect mno_ref is supplied, the sell request will fail.	String(20)	Yes
service_code	The Service Code identifies the type of goods or services being paid for. They are categories like Parking, Directory Service, etc. See appendix for a complete list of service codes.	String(255)	No
business_model	Business model defines commercial agreed Strex payment fee for the item paid for. Refer to appendix for the list of all business models.	String(255)	No
confirm_channel	Selects the communication channel where End User confirms the purchase. Options SMS and USSD must be used in combination with merchant_sell_option = 2. Possible Values: - SMS: End User confirms purchase by replying “ok” or “yes” on a SMS. Only allowed for merchant_sell_option=2 - USSD: Not In use - OTP: End User confirms the purchase by OTP. OTP value must be provided in “one_time_pass” parameter and transid from generateOTP in otp_transid. Default setting if parameter is not provided or left blank is SMS. Also refer to short_code parameter description for information about the SMS confirm message sent to the end user.	String	Yes
one_time_pass	One Time Password entered by End User at merchant side.		
otp_transid	Strex transaction ID from the generateOTP request used to send the OTP to End User.		
age	The minimum age of the End User required for this payment.	Number	Yes

	If not set or present in the request, there is no age limit.		
trans_ext_reference	External transaction reference. Must be unique for each request.	String(40)	Yes
merchant_transid	The merchant's internal transaction ID for this request. Non-unique values accepted.	Int	Yes
sms_message	The SMS message to be sent if the transaction is successful. This may represent the content of a premium SMS to be sent to the End User when this payment is successfully completed. The parameter SMS_message decides if the transaction is charge or charge + SMS; - for Strex Direct Payment, do not enter a MT SMS message text; - for Strex SMS Payment, enter a MT SMS message text	String(1024)	Yes
delivery_notify	A flag to indicate whether the Delivery Notification for the SMS Message above is required from the system that delivers the SMS. This shall not have any effect if SMS Message parameter is empty as no SMS message will be sent upon successful Charge. Defaults to True. If Delivery Notification is not required, set this parameter as false. Note: The Delivery Notification Flag only applies to the SMS delivery notification, in the case of a Sell trx that get's put on hold for Secure Web Registration, the Delivery Notification will be sent.	String(20)	Yes
validity_period	A period of time in seconds for which the responsible entity (e.g. SMSC) should keep trying to send the SMS Message to End User when the MSISDN is unreachable. Defaults to 36 hours (129600)(configurable).	Number	Yes
invoice_text	The invoice text to be displayed to the end user on the Strex transaction report. (strex.no/minside)	String(255)	Yes
subnumber	Strex supports SubNumbers of 10 digits. Together with ShortNumber, they form Extended ShortNumber. Allowed length of the extended shortnumber is 14 digits.	String(20)	Yes
short_code	Short code to be used as sender address of the		No

	SMS and/or MNO billing reference. If SMS confirm function is used, the following will apply to the SMS confirm message only to the end user: Full sender address will consist of short_code+10 digit sequential number which is used to match response from End User with request. Example sender address: "22221234567890", where "2222" is the short_code and "1234567890" is the sequential number. If short_code parameter is in the direct payment short number range (4xxx) in the request, the default setting for sender address will be "2222+sequential number".		
merchant_sell_option	This parameter controls the use of the authorization method in terms of End User authorization is required. Possible values are: • 1 = «none» • 2 = «confirmation» • 3 = «pin» (not in use) • 4 = BankID confirmation The value 2 will trigger SMS confirm authorization on the mobile terminal of the End User depending on the settings in confirm_channel and/or End User settings. Merchant Sell Option overrides confirm_channel=OTP. Option 3 is currently not in use. Option 4 will trigger BankID on mobile if this is available for the End User.	String(20)	Yes
timeout	Timeout setting for sell transaction which triggers End User registration (licensed purchases). A value of 1-30 will put the sell on hold while registration is executed. Default value is set to 5 minutes (subject to change). Possible values are: 0: prohibits triggering of registration	integer	Yes

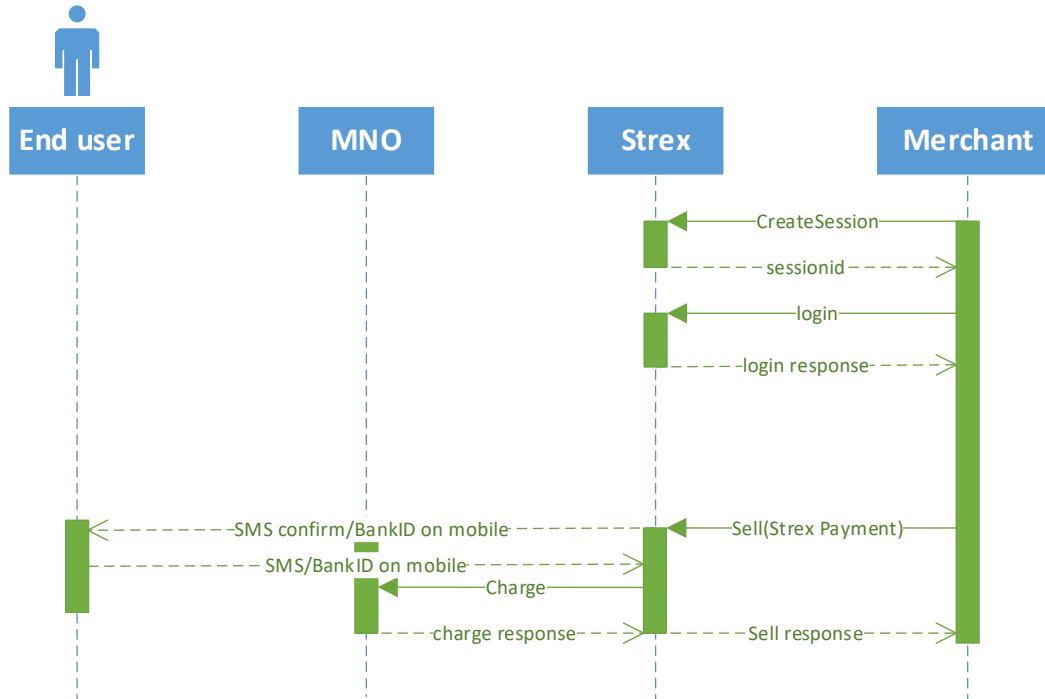
	1-30: number of minutes sell transaction is put on hold. Whole numbers only		
authorization	Authorization number to be used when sell is preauthorized by the End User. Request settings for one_time_pass and confirm_channel will be overruled if authorization parameter is set.	String	yes
merchant_reg_smstext	Merchant can change the text in the registration SMS using this parameter. The text will be prepended to strex registration link (example link: "strex.no/R/bYnN1j3-z5Jy0M/832/UM5")	string	yes
payment_source_restr	Some source of funds (e.g. payment cards and credit agreements) restricts payments for specific sectors, goods or services, as described in the Strex Merchant Agreement chapter 5.3. List of restricted sectors, goods or services is listed in Appendix I - Regulated services for card and credit agreement payments . If any of the items in the list applies, set payment_source_restr="true"	True/false	yes
change_date	Used to validate if the MSISDN is associated to the same recipient on transaction date/time compared to the change_date. Format: YYYY-MM-DDTHH:MM:SS. Example: <urn:change_date>2023-05-01T00:00:00</urn:change_date>	dateTime	yes
Teletorg section	Parameters in use for teletorg (premium voice services)		
b_number	The called teletorg number	string	yes
start_time	Start time of the charged teletorg voice call. Format: 'YYYY-MM-DD'T'hh:mm:ss'	dateTime	yes
duration	Charged voice call duration, in seconds	int	yes
start_price	Start price for the charged voice call. Start price should be in 1/100 NOK (e.g. 1 NOK = 100 Øre).	decimal	yes

Response Parameters

Parameter	Description	Type
result	The result code for this payment transaction.	Numeric
result_namespace	Namespace for the result code.	String
transid	The Strex transaction ID for this payment.	String(255)
trans_ext_reference	Merchant transaction reference as sent in the input parameters. Not present if not sent in the	String(255)

	request.	
merchant_transid	The Merchant's internal transaction ID. Not present if not sent in the request.	String(255)
recipient	The End User's MSISDN.	String(20)
postpaid_or_prepaid	Information about the End User's subscription type at the applicable MNO. Possible values are: • Postpaid • Prepaid Optional response Parameter.	String(255)
message_Type	Possible values are: • "Sell Direct Charge" • "Sell Charge SMS" This is an indicator for the merchant about how the original payment request arrived on merchant's system. The value to be set should be configurable based on the values of certain other response parameters above.	String(255)
handling_company	The Sell transaction has been handled by the Strex company indicated here, effective from 01.01.2023. Values: • "SxP" → Strex Payment AS • "SxMS" → Strex Mobile Services AS	String
handling_comp_descr	Description of handling company, effective from 01.01.2023. Values: • "Strex Payment AS" • "Strex Mobile Services AS"	String

8.2.2.2 Strex Direct Payment Request flow



Example request, Strex Direct Payment

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:sell>
      <urn:sellRequest>
        <urn:sessionid>6GQWYC8OF8IG00D51HH</urn:sessionid>
        <urn:amount>350</urn:amount>
        <urn:recipient>4799999999</urn:recipient>
        <urn:mno_ref>telenor</urn:mno_ref>
        <urn:service_code>01001</urn:service_code>
        <urn:short_code>4000</urn:short_code>
        <urn:business_model>STREX-PAYMENT</urn:business_model>
        <urn:invoice_text>Invoice text on Strex transaction
list</urn:invoice_text>
      </urn:sellRequest>
    </urn:sell>
  </soapenv:Body>
</soapenv:Envelope>
  
```

Example response, Strex Direct Payment

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:sellResponse xmlns="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="urn:UMARKETSCWS"
  
```

```

xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
  <ns3:sellReturn>
    <ns3:transid>714812</ns3:transid>
    <ns3:result>0</ns3:result>
    <ns3:result_namespace>umarket</ns3:result_namespace>
    <ns3:result_description>Success</ns3:result_description>
    <ns3:recipient>4799999999</ns3:recipient>
    <ns3:message_type>Sell Direct Charge</ns3:message_type>
  </ns3:sellReturn>
</ns3:sellResponse>
</S:Body>
</S:Envelope>

```

Example request, Strex Direct Payment with SMS confirm

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:sell>
      <urn:sellRequest>
        <urn:sessionid>6GQWYC8OF8IG00D51HH</urn:sessionid>
        <urn:amount>350</urn:amount>
        <urn:recipient>4799999999</urn:recipient>
        <urn:mno_ref>telenor</urn:mno_ref>
        <urn:service_code>01001</urn:service_code>
        <urn:short_code>4000</urn:short_code>
        <urn:business_model>STREX-PAYMENT</urn:business_model>
        <urn:confirm_channel>SMS</urn:confirm_channel>
        <urn:merchant_sell_option>2</urn:merchant_sell_option>
        <urn:invoice_text>Invoice text on Strex transaction
list</urn:invoice_text>
      </urn:sellRequest>
    </urn:sell>
  </soapenv:Body>
</soapenv:Envelope>

```

Example response, Strex Direct Payment with SMS confirm

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:sellResponse xmlns="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:sellReturn>
        <ns3:transid>714812</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:result_description>Success</ns3:result_description>
        <ns3:recipient>4799999999</ns3:recipient>
        <ns3:message_type>Sell Direct Charge</ns3:message_type>
      </ns3:sellReturn>
    </ns3:sellResponse>
  </S:Body>
</S:Envelope>

```

```
</S:Body>
</S:Envelope>
```

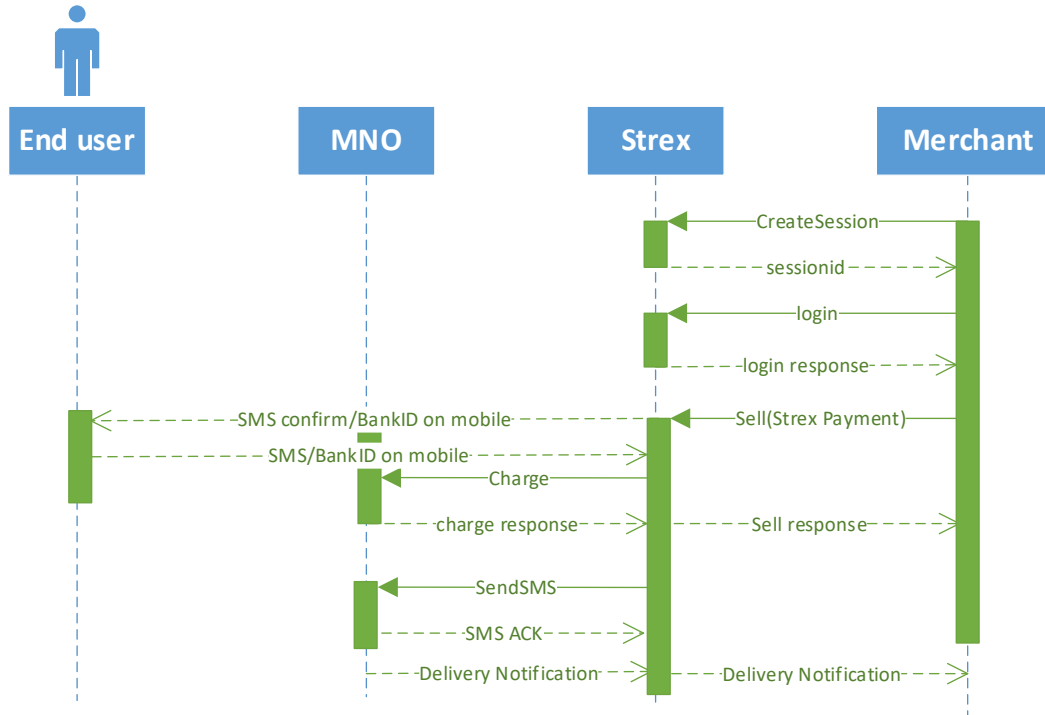
Example request, Strex Direct Payment with Teletorg parameters

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc"
xmlns:typ="http://www.utiba.com/delirium/ws/TypedQuery" >
  <soapenv:Header/>
  <soapenv:Body>
    <urn:sell>
      <urn:sellRequest>
        <urn:sessionid>6GQWYC8OF8IG00D51HH</urn:sessionid>
        <urn:amount>350</urn:amount>
        <urn:recipient>4799999999</urn:recipient>
        <urn:mno_ref>telenor</urn:mno_ref>
        <urn:service_code>13004</urn:service_code>
        <urn:short_code>4000</urn:short_code>
        <urn:business_model>TELETORG</urn:business_model>
        <urn:invoice_text>82900000
Teletorgtjenestester</urn:invoice_text>
        <urn:teletorgReq>
          <typ:b_number>82900000</typ:b_number>
          <typ:start_time>2022-12-22T22:22:22</typ:start_time>
          <typ:duration>120</typ:duration>
          <typ:start_price>100</typ:start_price>
        </urn:teletorgReq>
      </urn:sellRequest>
    </urn:sell>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response, Strex Direct Payment with Teletorg parameters

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:sellResponse xmlns="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:sellReturn>
        <ns3:transid>714812</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:result_description>Success</ns3:result_description>
        <ns3:recipient>4799999999</ns3:recipient>
        <ns3:message_type>Sell Direct Charge</ns3:message_type>
      </ns3:sellReturn>
    </ns3:sellResponse>
  </S:Body>
</S:Envelope>
```

8.2.2.3 Strex SMS Payment message flow



Example request, Strex SMS Payment

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:sell>
      <urn:sellRequest>
        <urn:sessionid>6GQWYC8OF8IG00D51HH</urn:sessionid>
        <urn:amount>350</urn:amount>
        <urn:recipient>4799999999</urn:recipient>
        <urn:mno_ref>telenor</urn:mno_ref>
        <urn:service_code>01001</urn:service_code>
        <urn:short_code>2222</urn:short_code>
        <urn:business_model>STREX-PAYMENT</urn:business_model>
        <urn:sms_message>This message will go to the End
User</urn:sms_message>
        <urn:age>18</urn:age>
        <urn:invoice_text>This is the text that will show on Strex'
transaction report</urn:invoice_text>
        <urn:delivery_notify>true</urn:delivery_notify>
        <urn:validity_period>180</urn:validity_period>
        <urn:subnumber>1234567890</urn:subnumber>
      </urn:sellRequest>
    </urn:sell>
  </soapenv:Body>
</soapenv:Envelope>
  
```

Example response, Strex SMS Payment

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:sellResponse xmlns="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:sellReturn>
        <ns3:transid>714812</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:result_description>Success</ns3:result_description>
        <ns3:recipient>4799999999</ns3:recipient>
        <ns3:message_type>Sell Charge SMS</ns3:message_type>
      </ns3:sellReturn>
    </ns3:sellResponse>
  </S:Body>
</S:Envelope>
```

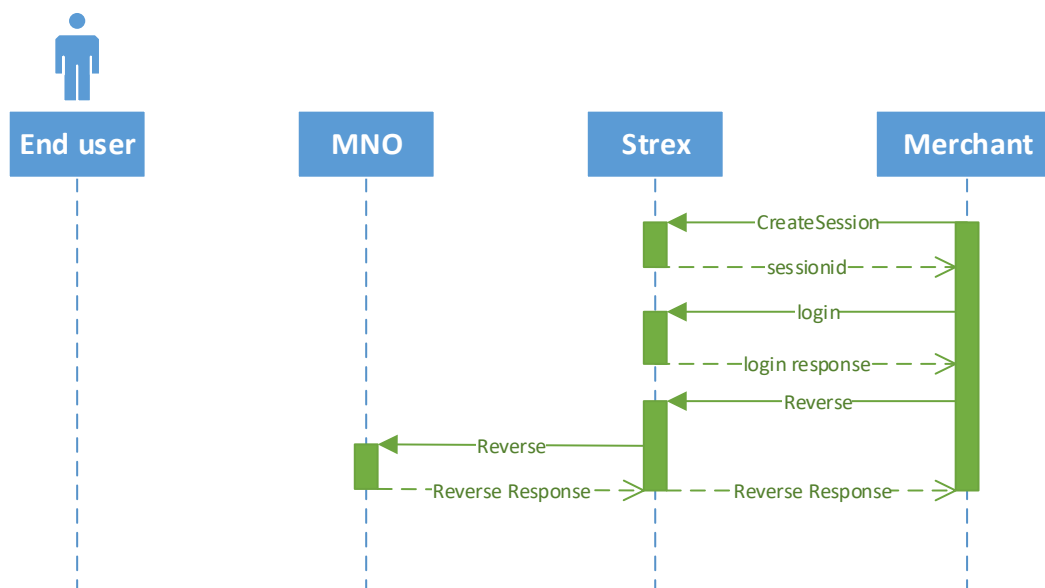
8.2.3 Reverse transaction

This function is used to reverse a previous successful payment transaction. Only reversal of the full amount is supported.

If the transaction id of the original Sell transaction is needed in the response, use one of the parameters fields for external/merchant reference as reference since the original sell transaction reference is not returned in the response by default.

Reversal is available up to 6 months (180 days) after the Sell transaction is executed.

Reverse transaction flow



Input Parameters

Parameter	Description	Type	Optional
sessionid	Valid sessionid retrieved from createsession response	-	No
transaction_ID	The transaction ID of the transaction to be reversed.	String	No
trans_ext_reference	The Merchant's reversal transaction ID.	String(255)	Yes
merchantTransID	Merchant transaction ID (duplicate)		Yes

Response Parameters

Parameter	Description	Type
Result	The result code for this payment transaction.	Numeric
Result_namespace	Namespace related to the result code	String(255)
transid	The Strex transaction ID for this payment.	String(255)
trans_ext_reference	external reversal transaction ID. (if present in the request)	String(255)
merchantTransID	Merchant transaction ID (if present in the request)	

Example request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:reverse>
      <urn:reverseRequest>
        <urn:sessionid>H2IMRWBID06ABH2LCTUU</urn:sessionid>
        <urn:transactionID>170093</urn:transactionID>
      </urn:reverseRequest>
    </urn:reverse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <reverseResponse xmlns="urn:UMARKETSCWS"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <reverseReturn>
```

```

        <transid>170094</transid>
        <result>0</result>
        <result_namespace>umarket</result_namespace>
    </reverseReturn>
</reverseResponse>
</S:Body>
</S:Envelope>

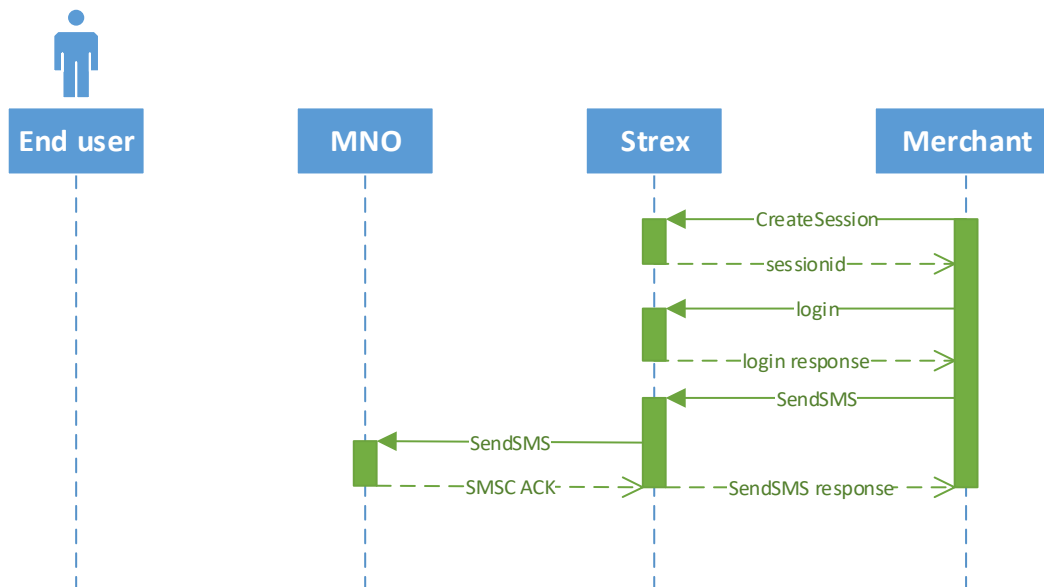
```

8.2.4 SendSMS

This function allows the merchant to send an SMS to a End User.

Please note:

- the parameter sms_originator must be set for all SendSMS requests. Norwegian characters in sms_originator is not supported for messages sent via NetCom SMSC.
- Delivery Notification for SendSMS is supported for all MNOs.



Input Parameters

Parameter	Description	Type	Optional
sessionid	Valid sessionid retrieved from createsession response	-	No
recipient	The End User who will receive the SMS. The end user is a End User with a MNO/service provider that have an agreement with Strex.	String(255)	No
sms_originator	The Address that the receiver will see this message has come from. Norwegian characters only supported for some MNOs.	String(11)	No
subnumber	Not supported.	String(20)	Yes

message	The SMS message. Max. 1024 characters	String(1024)	No
trans_ext_reference	Merchant transaction id for this request		

Response Parameters

Parameter	Description	Type
result	The result code for this payment transaction.	Numeric
result_Description	The textual descriptions of the Result Code above.	String(255)
trans_ID	The Strux transaction ID for this payment.	String(255)
fee	Returned in the response but not in use.	

Example request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:sendSMS>
      <urn:sendSMSRequest>
        <urn:sessionId>H2IMRWBID06ABH2LCTUU</urn:sessionId>
        <urn:message>This is the message to the end
user</urn:message>
<urn:sms_originator>2497</urn:sms_originator>
        <urn:recipient>4799999999</urn:recipient>
      </urn:sendSMSRequest>
    </urn:sendSMS>
  </soapenv:Body>
</soapenv:Envelope>
```

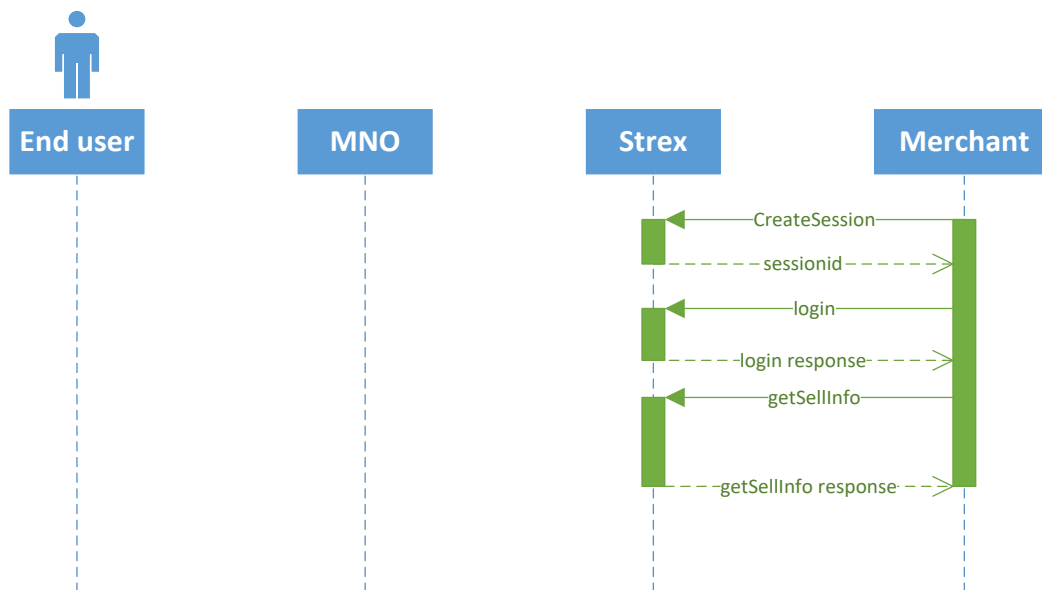
Example response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <sendSMSResponse xmlns="urn:UMARKETSCWS"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <sendSMSReturn>
        <transid>170212</transid>
        <result>0</result>
        <result_namespace>umarket</result_namespace>
        <fee>0.00</fee>
      </sendSMSReturn>
    </sendSMSResponse>
  </S:Body>
</S:Envelope>
```

8.2.5 Get Sell Information

The getSellInfo function returns the statusCode of a merchants previously executed Sell request. Accepted input parameters is minimum one or maximum all of Original Sell transaction id (o_Trans_ID), trans_ext_reference (o_trans_ext_ref) and merchant_transid.

If the Sell request is not completed, a specific status code will be given in the response (4100)



Input Parameters

Parameter	Description	Type	Optional
sessionid	Valid sessionid retrieved from createsession response	-	No
o_Trans ID	ID of the Original Sell transaction request		Yes
o_trans_ext_ref	Merchant Transaction ID (trans_ext_reference), of the original Sell request.		Yes
Merchant_transid	Merchant Transaction ID, of the original Sell request.		Yes

Response Parameters

Parameter	Description	Optional
o_Trans ID	ID of the Original Sell transaction request in U:market	Yes
result	Statuscode of the getsellinfo transaction	No
result_namespace	Namespace associated with the result.	Yes
o_trans_ext_ref	Merchant Transaction ID (trans_ext_reference), of the original Sell request.	Yes
merchant_transid	Merchant Transaction ID, of the original Sell request.	Yes
status_Code	Status codes are the same as the ones used on the Sell (API) response.	No
trans ID	ID of the GetSellinfo transaction	No
handling_company	The Sell transaction has been handled by the Strex company indicated here, effective from 01.01.2023. Values: “SxP” → Strex Payment AS	String

	• “SxMS” → Strex Mobile Services AS	
handling_comp_descr	Description of handling company, effective from 01.01.2023. Values: • “Strex Payment AS” • “Strex Mobile Services AS”	String

Example request

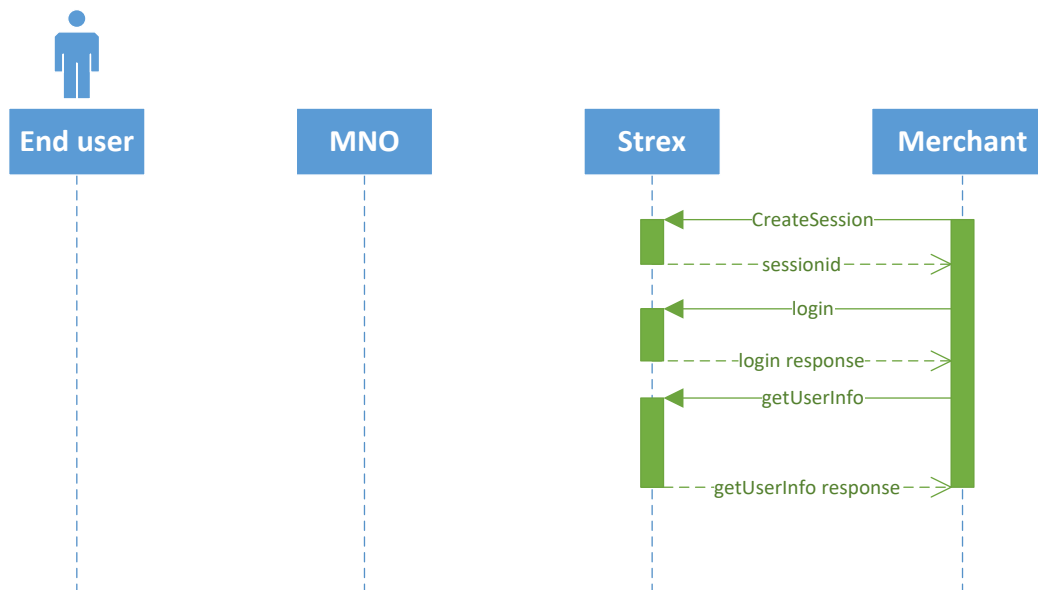
```
<soapenv:Envelope xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope/
xmlns:urn=urn:UMARKETSCWS
xmlns:std=http://www.utiba.com/delirium/ws/StdQuery
xmlns:misc=http://www.utiba.com/delirium/ws/Misc>
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getSellInfo>
      <urn:getSellInfoRequest>
        <urn:sessionId>ZZ11U2PQ6GN8LOOSYI6D</urn:sessionId>
        <urn:o_transid>33921330</urn:o_transid>
      </urn:getSellInfoRequest>
    </urn:getSellInfo>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:SellInfoResponse xmlns="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:SellInfoReturn>
        <ns3:transid>33922915</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:o_transid>33921330</ns3:o_transid>
        <ns3:status_code>0</ns3:status_code>
      </ns3:SellInfoReturn>
    </ns3:SellInfoResponse>
  </S:Body>
</S:Envelope>
```

8.2.6 Get User Information

The GetUserInformation function returns information about an End Users registration status on the Strex Payment Service and the latest registered terminal type (handset OS).



Input Parameters

Parameter	Description	Optional
sessionid	Valid sessionid retrieved from createsession response	No
msisdn	MSISDN of the End User 10 digits on format "4798654321"	No

Response Parameters

Parameter	Description	Optional
transid	Strex transid of the GetUserInformation transaction	No
result	Statuscode of the GetUserInformation transaction	No
result_namespace	Namespace associated with the result.	Yes
msisdn	MSISDN of the End User	Yes
Validity	Describes validity for licensed purchases as well as barring. 0 = not registered 1 = Registered, but not valid for licensed purchase. 2 = Valid for licensed purchase 3 = MNO billing barred 4 = Registered and BankID authenticated	No
terminal_type	Result is "Android" or "iPhone OS" or returns handset model.	Yes

Example request

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>

```

```

    <urn:UserInfo>
      <urn:getUserInfoRequest>
        <urn:sessionId>A6NP6YL3H6PA8VQP0CA4</urn:sessionId>
        <urn:msisdn>4799999999</urn:msisdn>
      </urn:getUserInfoRequest>
    </urn:UserInfo>
  </soapenv:Body>
</soapenv:Envelope>

```

Example response

```

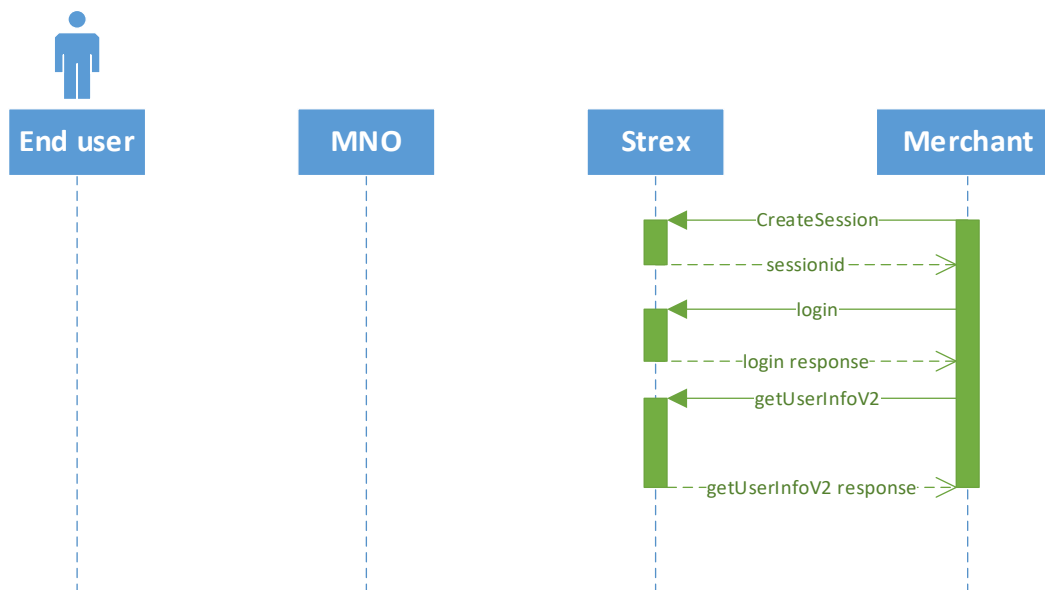
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:UserInfoResponse xmlns="http://www.utiba.com/delirium/ws/Misc"
      xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
      xmlns:ns3="urn:UMARKETSCWS"
      xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:UserInfoReturn>
        <ns3:transid>33922942</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:msisdn>4799999999</ns3:msisdn>
        <ns3:validity>2</ns3:validity>
        <ns3:terminal_type>Android</ns3:terminal_type>
      </ns3:UserInfoReturn>
    </ns3:UserInfoResponse>
  </S:Body>
</S:Envelope>

```

8.2.7 Get User Information v2

The `getuserinfo v2` function runs an extensive “sell validation check”. Based on the input parameters, the response will provide an **indicative** response on whether the subscriber can be charged with the given input parameters. The function checks:

- Subscriber blocked or suspended status
- MNO subscription status (active/inactive)
- MNO CPA (premium SMS billing) barring
- MNO agreement/integration with Strex
- Validation of PSD2 limits (single transactions and aggregated monthly amount)
- Validation of MNO and card billing limits (single transactions and aggregated monthly amount)
- Validity for licensed purchases
- Subscriber Age if present in request
- MNO prepaid balance (selected merchants)



Input Parameters

Parameter	Description	Optional
sessionid	Valid sessionid retrieved from createsession response	No
msisdn	MSISDN of the End User 10 digits on format "4798654321"	No
amount	The amount to charge the subscriber. Amount should be in 1/100 NOK (e.g. 1 NOK = 100 Øre).	No
service_code	The Service Code identifies the type of goods or services being paid for. They are categories like Parking, Directory Service, etc. See appendix for a complete list of service codes.	No
age	Subscriber age. If provided, age of subscriber is checked to be above the input age.	Yes

Response Parameters

Parameter	Description	Optional
transid	Strex transid of the GetUserInformationv2 transaction	No
result	Statuscode of the GetUserInformationv2 transaction	No
result_namespace	Namespace associated with the result.	Yes
fee	Not in use	yes

Example request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```

xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getUserInfoV2>
      <urn:getUserInfoV2Request>
        <urn:sessionId>A6NP6YL3H6PA8VQP0CA4</urn:sessionId>
        <urn:msisdn>4799999999</urn:msisdn>
        <urn:amount>3000</urn:amount>
        <urn:service_code>14001</urn:service_code>
      </urn:getUserInfoV2Request>
    </urn:getUserInfoV2>
  </soapenv:Body>
</soapenv:Envelope>

```

Example response

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:getUserInfoV2Response
xmlns="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:getUserInfoV2Return>
        <ns3:transid>35502697</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:fee>0.00</ns3:fee>
      </ns3:getUserInfoV2Return>
    </ns3:getUserInfoV2Response>
  </S:Body>
</S:Envelope>

```

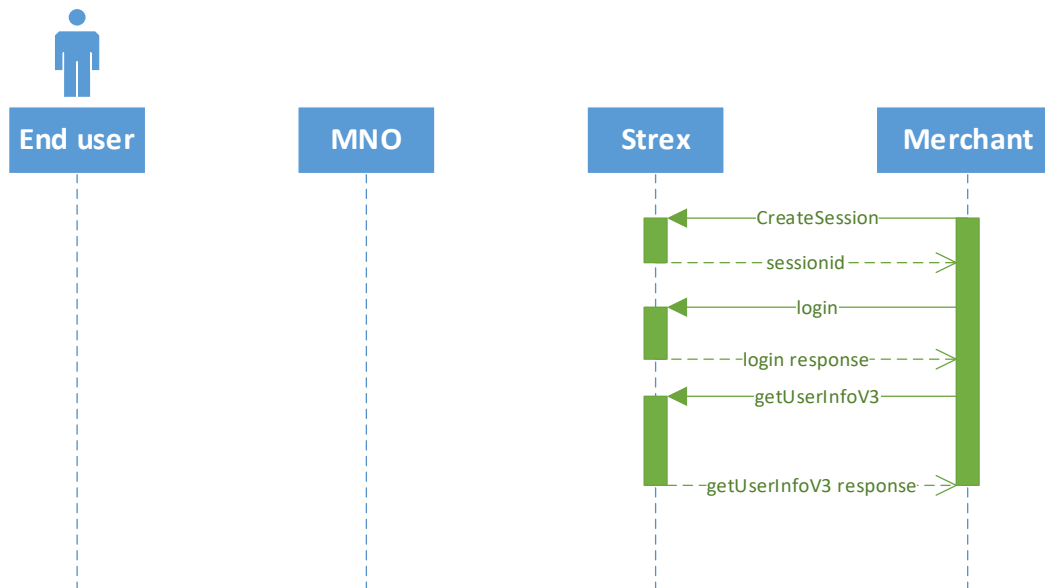
Valid Status Codes for getuserinfoV2 request:

Error code	Description	Explanation
3	AGENT_NOT_FOUND	The Agent attempting to access the system could not be found
4001	MNO_NOT_RESOLVED	The End Users MNO does not have an agreement with Strex/technical error, i.e. Strex could not identify which MNO the End User belongs to.
4070	MISSING_FIELD	Missing mandatory input parameter
4081	INVALID_REQUEST	INVALID_REQUEST
4116	SSN_BLACKLISTED	The End User is blacklisted
4120	Error during PSD2 validation	Error during PSD2

		validation
4169	FAILED_BY_PAYMENT_RESTRICTION_MATRIX_VALIDATION	End User has no available Payment Source (i.e. CPA barred, and alternative payment source is not registered)
4170	PSD2_SERVICE_CODE_IS_LICENSED	End User is unregistered and Service_code is licensed
4171	PSD2_IN_BYPASS_GROUP_BUT_SUSPENDED	End User is suspended
4172	PSD2_AMOUNT_VIOLATE_LIMIT	amount over PSD purchase limit
4173	PSD2_SUBSCRIBER_SUSPENDED	End User is suspended
4174	PSD2_CAPSET_EXCEEDED	amount over PSD monthly purchase limit
4176	SUBSCRIBER_NOT_ACTIVE	Subscriber does not have an active subscription
4183	IMEI_IS_BLACKLISTED_ACCESS_DENIED	Handset (IMEI) is blacklisted.
4218	Amount exceeds the MNO subscriber balance.	Amount exceeds subscriber prepaid balance
4219	Amount exceeds the MNO payment restrictions.	Amount exceeds the subscriber max monthly amount defined at MNO

8.2.8 Get User Information v3

The getuserinfo v3 function retrieves the current (snapshot at the time of the request) transaction limits for an End User. The transaction limits depends on End User registration level, age and preferred payment method (MNO bill or card payment). The response will provide an **indicative** response, i.e. there is no guarantee that the sell will pass. MNO status (e.g. prepaid balance, CPA block etc.) can be checked for selected merchants and some MNOs only (depending on implementation by the MNOs). Payment Card balance is not checked.



Input Parameters

Parameter	Description	Optional
sessionid	Valid sessionid retrieved from createsession response	No
msisdn	MSISDN of the End User 10 digits on format "4798654321"	No

Response Parameters

Parameter	Description	Optional
transid	Strex transid of the GetUserInformationv3 transaction	No
result	Statuscode of the GetUserInformationv3 transaction	No
result_namespace	Namespace associated with the result.	Yes
preferred_sof	Subscriber preferred source of funds: 1 – MNO bill 2 – (not in use) 3 – payment card	No
postpaid_or_prepaid	Subscriber MNO subscription type: - "prepaid" - "postpaid"	Yes
limit_trans	Maximum amount (In øre) which can be charged in a single sell transaction	Yes
remainder_month	Remaining monthly amount (in øre) which can be charged the subscriber	Yes

	before exceeding the monthly limit. "0" - limit already reached	
remainder_year	Remaining yearly amount (in øre) which can be charged the subscriber before exceeding the yearly limit. "0" - limit already reached "-1" - no limit applicable, subscriber valid for purchases above defined yearly limit	Yes

Example request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getUserInfoV3>
      <urn:getUserInfoV3Request>
        <urn:sessionId>8TLMR6M9RHU6C9N0K3F2</urn:sessionId>
        <urn:msisdn>4799999999</urn:msisdn>
      </urn:getUserInfoV3Request>
    </urn:getUserInfoV3>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:getUserInfoV3Response
xmlns="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:getUserInfoV3Return>
        <ns3:transid>35905721</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:preferred_sof>3</ns3:preferred_sof>
        <ns3:postpaid_or_prepaid>postpaid</ns3:postpaid_or_prepaid>
        <ns3:limit_trans>100000</ns3:limit_trans>
        <ns3:remainder_month>500000</ns3:remainder_month>
        <ns3:remainder_year>-1</ns3:remainder_year>
      </ns3:getUserInfoV3Return>
    </ns3:getUserInfoV3Response>
  </S:Body>
</S:Envelope>
```

8.2.9 preAuthorize

Preauthorized payments is recurring payments where the End User have confirmed that Sell transactions for a merchant can be executed without End User confirmation on each Sell.

Preauthorization request returns a token to the merchant to be used in subsequent sell transactions.

The token Issued by Strex does not expire after a given time period. Therefore, there is no time limit for Sell transactions referencing a pre-authorize token. However, the token may be made invalid by Strex in some situations. Please refer to the Code of Conduct document.

If required, please contact Strex Technical Support.

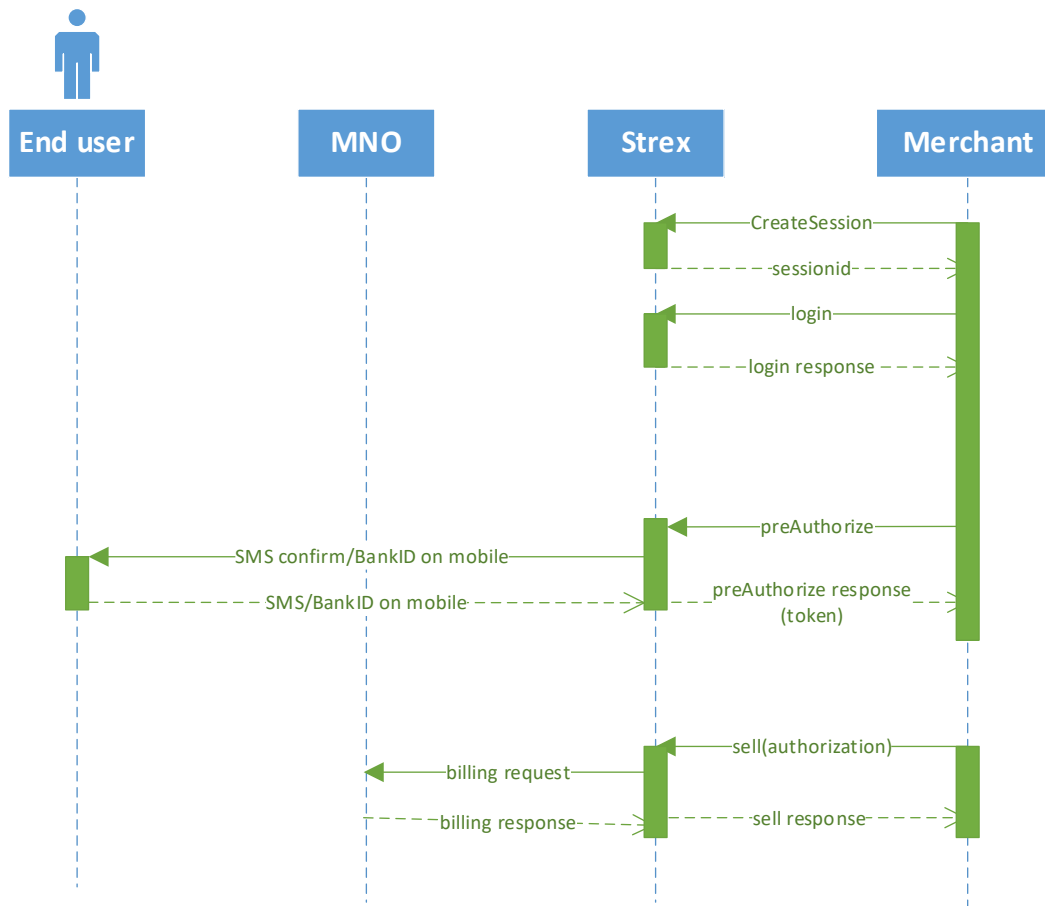
Please note that the End User needs to be a user on the Strex payment platform prior to preauthorize request to be successful, so the following process need to be followed:

Process using SMS confirm:

1. The End User purchase a service or good from the merchant by executing a sell request.
2. The merchant trigger a preauthorization request on Strex Payment Service
3. The End User confirms future payments in the SMS dialogue on the mobile handset
4. For each subsequent payment for the same End User, the merchant send the Sell request with «authorization» parameter and no SMS or authorization from End User is needed.

Preauthorization can also be confirmed by using an OTP, please refer to [generateOTP](#) chapter.

Preauthorization tokens can be shared between merchants, upon agreement with Strex.



Input Parameters

Parameter	Description	Type	Optional
sessionid	Valid sessionid retrieved from createsession response	-	No
trans_ext_reference	This is the Merchant's transaction ID.		Yes
msisdn	MSISDN of the End User		No
security_Level	This is the required Confirmation type of the preAuth: 1 = "none" (use of this setting requires an agreement with Strex) 2 = "Confirmation" 3 = "PIN" (not in use) 4 will trigger BankID on mobile If this Is available for this End User. Security level overrides OTP setting in confirm_channel		No
confirm_channel	Selects the communication channel where End User	String	Yes

	confirms the purchase. Must be used in combination with security_level = 2. Possible Values: • SMS: End User confirms preAuth request by replying “ok” or “yes” on a SMS. Only allowed for security_level = 2. Timeout for end user response is 90 sec. • USSD: not In use • OTP: End User confirms the preAuth by OTP. OTP value must be provided in “one_time_pass” parameter and transid from generateOTP in otp_transid. Default setting if parameter is not provided or left blank is SMS.		
one_time_pass	One Time Password entered by End User at merchant side.		
otp_transid	Strex transaction ID from the generateOTP request used to send the OTP to End User.		
short_code	short_code to be used as sender address of the SMS Confirm message. Full sender address will consist of short_code+10 digit sequential number which is used to match response from End User with request. Example sender address: “22221234567890”, where “2222” is the short_code and “1234567890” is the sequential number. If parameter is missing or left blank or in the direct payment short number range (4xxx) in the request, the default setting for sender address will be “2222”.		Yes
message_prefix	Text string which will be prepended to the standard Strex SMS message sent to the subscriber		
message_suffix	Text string which will be appended to the standard Strex SMS message sent to the subscriber		
token_description	Token description, i.e. information on what the token is used for.		
age	The minimum age of the End User required for this request. If not set or present in the request, there is no age limit.		

Response Parameters

Parameter	Description	Optional
transid	Strex transid of the getUserInfo transaction	No

result	Statuscode of the getsellinfo transaction	No
Result_namespace	Namespace associated with the result.	Yes
authorization	This is the Authorization token for this Pre-Authorization request.	Yes
trans_ext_reference	This is the Merchant's transaction ID of the PreAuthorization_req	Yes

Example request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:preAuthorization>
      <urn:preAuthorizationRequest>
        <urn:sessionid>VG9MHZFH9Q42S22OZUNJ</urn:sessionid>
        <urn:trans_ext_reference>12341236</urn:trans_ext_reference>
        <urn:msisdn>4799999999</urn:msisdn>
        <urn:security_level>2</urn:security_level>
      </urn:preAuthorizationRequest>
    </urn:preAuthorization>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:preAuthorizationResponse
xmlns="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:preAuthorizationReturn>
        <ns2:transid>33932384</ns2:transid>
        <ns2:result>0</ns2:result>
        <ns2:result_namespace>umarket</ns2:result_namespace>

      <ns3:authorization>33932366499998920161122095128</ns3:authorization>
        <ns3:trans_ext_reference>12341236</ns3:trans_ext_reference>
      </ns3:preAuthorizationReturn>
    </ns3:preAuthorizationResponse>
  </S:Body>
</S:Envelope>
```

Example request, preauthorization with SMS confirm

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
```

```

    <urn:preAuthorization>
      <urn:preAuthorizationRequest>
        <urn:sessionid>'VG9MHZFH9Q42S22OZUNJ'</urn:sessionid>
        <urn:trans_ext_reference>12341236</urn:trans_ext_reference>
        <urn:msisdn>4799999999</urn:msisdn>
        <urn:security_level>2</urn:security_level>
        <urn:confirm_channel>SMS</urn:security_level>
        <urn:short_code>4063</urn:security_level>
      </urn:preAuthorizationRequest>
    </urn:preAuthorization>
  </soapenv:Body>
</soapenv:Envelope>

```

Example response, preauthorization with SMS confirm

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:preAuthorizationResponse
xmlns="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:preAuthorizationReturn>
        <ns2:transid>33932384</ns2:transid>
        <ns2:result>0</ns2:result>
        <ns2:result_namespace>umarket</ns2:result_namespace>
      </ns3:preAuthorizationReturn>
      <ns3:authorization>33932366499998920161122095128</ns3:authorization>
      <ns3:trans_ext_reference>12341236</ns3:trans_ext_reference>
    </ns3:preAuthorizationResponse>
  </S:Body>
</S:Envelope>

```

8.2.10 selfregisterV3

Input Parameters

Parameter	Description	Type	Optional
sessionid	Valid sessionid retrieved from createsession response	-	No
ext_id	External transaction reference.	String	No
msisdn	The mobile subscriber (End User) who receives the registration SMS	String(20)	No
sms_text	SMS text to be added in the registration SMS. Example: "please register here" "strex.no/R/rIE1pkY-5Gv598/832/" will be added by Strex. If supplied by merchant, the campaign code will be appended (here "SX1") Complete SMS example: "Please register here: strex.no/R/rIE1pkY-5Gv598/832/SX1"	String	No
campaign_code	The campaign code will be stored on the user profile of the registered End User for statistical purposes. Example "SX1"	String	Yes

Response Parameters

Parameter	Description	Type
result	The result code for this payment transaction.	Numeric
result_namespace	Namespace for the result code.	String
transid	The Strex transaction ID for this payment.	String(255)

Example request, selfregisterV3

```
<?xml version="1.0"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:selfregisterV3>
      <urn:selfRegisterV3Request>
        <urn:sessionid>OYN2C2OASI5UWKB6UKV1</urn:sessionid>
        <urn:ext_id>1234587</urn:ext_id>
        <urn:msisdn>4799999999</urn:msisdn>
        <urn:sms_text>Please register here: </urn:sms_text>
        <!--Optional-->
        <urn:campaign_code>SX1</urn:campaign_code>
      </urn:selfRegisterV3Request>
    </urn:selfregisterV3>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</urn:selfRegisterV3Request>
</urn:selfregisterV3>
</soapenv:Body>
</soapenv:Envelope>
```

Example response, selfregisterV3

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:selfRegisterV3Response
xmlns="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:transid>34172072</ns3:transid>
      <ns3:result>0</ns3:result>
      <ns3:result_namespace>umarket</ns3:result_namespace>
    </ns3:selfRegisterV3Response>
  </S:Body>
</S:Envelope>
```

8.2.11 generateOTP

generateOTP will send a One Time Password (4 digit PIN) to the End User which can be used for confirmation of a subsequent sell or preauthorize transaction. The OTP is generated by Strex platform when generateOTP request is received. End User will enter the OTP on a merchant interface (web-page, app etc.) and the merchant can initiate a Sell or preauthorization with the OTP and transid of the generateOTP request included. The OTP/transid will then be verified by Strex platform.

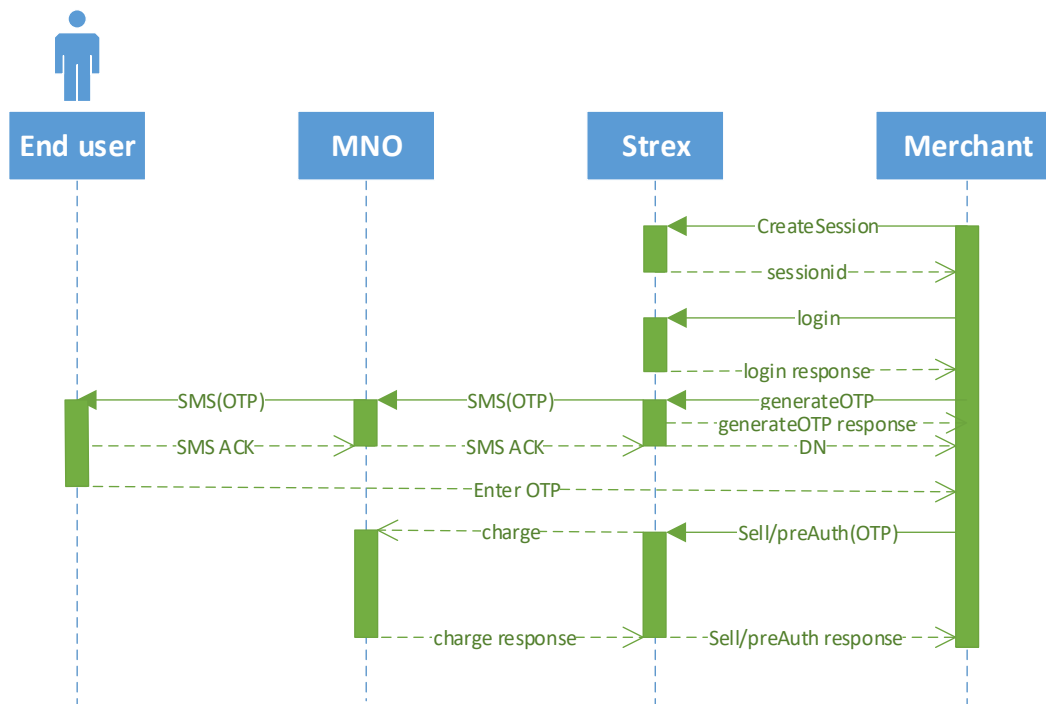
OTP is valid for one-time use only.

The process flow below describes the process of generating an OTP and using it in a subsequent sell or preauthorization transaction.

OTP SMS message format: <prefix_message><Strex text><Suffix message>

The total SMS message length must not exceed 1024 characters.

OTP SMS message example: «A prefix message here. 1234 er din engangskode for å inngå avtale om fremtidige belastninger til CompanyName. Suffix message here.»



Input Parameters

Parameter	Description	Type	Optional
sessionId	Valid sessionId retrieved from createsession response	-	No
recurring	True/false. Set as "false" for use in a sell transaction Set as "true" for use in preAuth transactions		No
recipient	The mobile subscriber (End User) who receives the OTPSMS	String(20)	No
message	SMS text defined by merchant to be added to the beginning of the SMS message. If not set in request, a default message will be sent. Default message example: "1234 er din engangskode for å bekrefte kjøp hos Merchant"	String	Yes
sms_originator	The Address that the receiver will see this message has come from. Norwegian characters only supported for some MNOs. A default sms_originator (Currently "Engangskode") will be used if not set in request.	String(11)	Yes
timeout	Validity period for OTP. Defined in seconds. Minimum 1, max 86400 (24 hours). If not set in request, default value of 300 (5 min) will be used.		Yes
delivery_notify	True/false. If set to "true", a delivery notification request will be sent to the merchant.		Yes

message_prefix	Text string which will be prepended to the standard Strex SMS message sent to the subscriber		Yes
message_suffix	Text string which will be appended to the standard Strex SMS message sent to the subscriber		Yes

Response Parameters

Parameter	Description	Type
Result	The result code for this payment transaction.	Numeric
Result_namespace	Namespace for the result code.	String
Transid	The Strex transaction ID for this payment.	String(255)

Example request, generateOTP

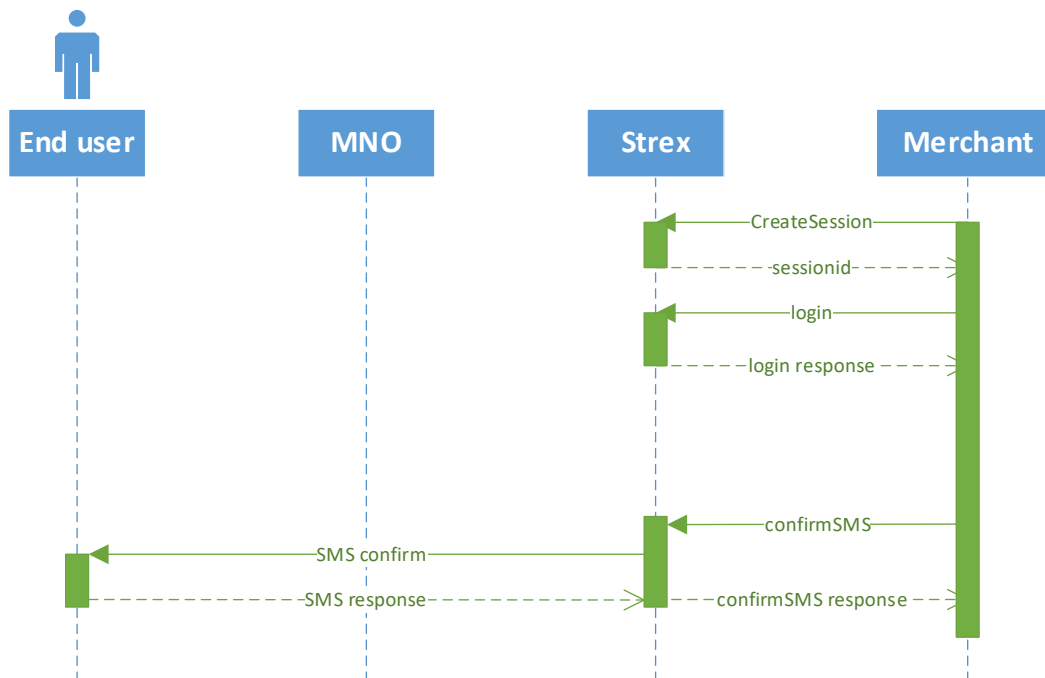
```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:generateOTP>
      <urn:generateOTPRequest>
        <urn:sessionid>IYK02E4NW864X8PQP753</urn:sessionid>
        <urn:recipient>4799999999</urn:recipient>
        <urn:recurring>false</urn:recurring>
      </urn:generateOTPRequest>
    </urn:generateOTP>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response, generateOTP

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <generateOTPResponse xmlns="urn:UMARKETSCWS"
xmlns:ns2="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns3="http://www.utiba.com/delirium/ws/TypedQuery"
xmlns:ns4="http://www.utiba.com/delirium/ws/StdQuery">
      <generateOTPReturn>
        <transid>717895971</transid>
        <result>4110</result>
        <result_namespace>umarket</result_namespace>
      </generateOTPReturn>
    </generateOTPResponse>
  </S:Body>
</S:Envelope>
```

8.2.12 confirmSMS

End User confirmation by replying “ok” or “yes” on a SMS. The SMS is sent from a 14 digit sender address.



Input Parameters

Parameter	Description	Type	Optional
sessionid	Valid sessionid retrieved from createsession response	-	No
trans_ext_reference	This is the Merchant's transaction ID. Same id is returned in the response.		Yes
recipient	MSISDN of the End User		No
sms_originator	The four digit SMS shortnumber used by the merchant. Together with subnumber, they form the Extended ShortNumber which will be the sender of the confirmation SMS received on the subscriber handset. If not supplied in the request, Strex default number “2222” will be used as the shortnumber. Example sender address: “22221234567890” Please note that shortnumbers in the 4xxx series cannot be used for this request.		Yes

message	Optional SMS message text supplied by the merchant. Max. 1024 characters. A default text "Bekreft ved å svare OK på denne meldingen" will be used if the message text is not supplied by the merchant.	String(1024)	yes
transaction_ref	Merchant transaction id for this request. Not returned in response		yes

Response Parameters

Parameter	Description	Optional
transid	Strex transid of the confirmSmsRequest transaction	No
result	Statuscode of the confirmSmsRequest transaction	No
result_namespace	Namespace associated with the result.	Yes
fee	Not in use	No
trans_ext_reference	This is the Merchant's transaction ID sent in the request.	yes

Example request, confirmSMS

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:UMARKETSCWS"
xmlns:std="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:misc="http://www.utiba.com/delirium/ws/Misc">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:confirmSms>
      <urn:confirmSmsRequest>
        <urn:sessionid> IYK02E4NW864X8PQP753</urn:sessionid>
        <urn:trans_ext_reference>1590654176</urn:trans_ext_reference>
        <urn:recipient>4799999999</urn:recipient>
        <urn:sms_originator>2222</urn:sms_originator>
        <urn:message>Please confirm your free membership by replying
OK to this message</urn:message>
      </urn:confirmSmsRequest>
    </urn:confirmSms>
  </soapenv:Body>
</soapenv:Envelope>
```

Example response, confirmSMS:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:confirmSmsResponse
xmlns="http://www.utiba.com/delirium/ws/Misc"
xmlns:ns2="http://www.utiba.com/delirium/ws/StdQuery"
xmlns:ns3="urn:UMARKETSCWS"
xmlns:ns4="http://www.utiba.com/delirium/ws/TypedQuery">
      <ns3:confirmSmsReturn>
        <ns3:transid>1489590294</ns3:transid>
        <ns3:result>0</ns3:result>
        <ns3:result_namespace>umarket</ns3:result_namespace>
        <ns3:fee>0.00</ns3:fee>
        <urn:trans_ext_reference>1590654176</urn:trans_ext_reference>
```

```

    </ns3:confirmSmsReturn>
  </ns3:confirmSmsResponse>
</S:Body>
</S:Envelope>

```

8.3 Outgoing Messages

This chapter describes the requests and responses for the web services that must be implemented by the merchant.

8.3.1 Retry schedule

Strex platform will retry delivery of SMS forward and DN to the merchant if delivery is unsuccessful. The following retry schedule is implemented (subject to change):

Number of retries	Time since previous unsuccessful delivery	Retry number
1	15 sec	1
1	5 min	2

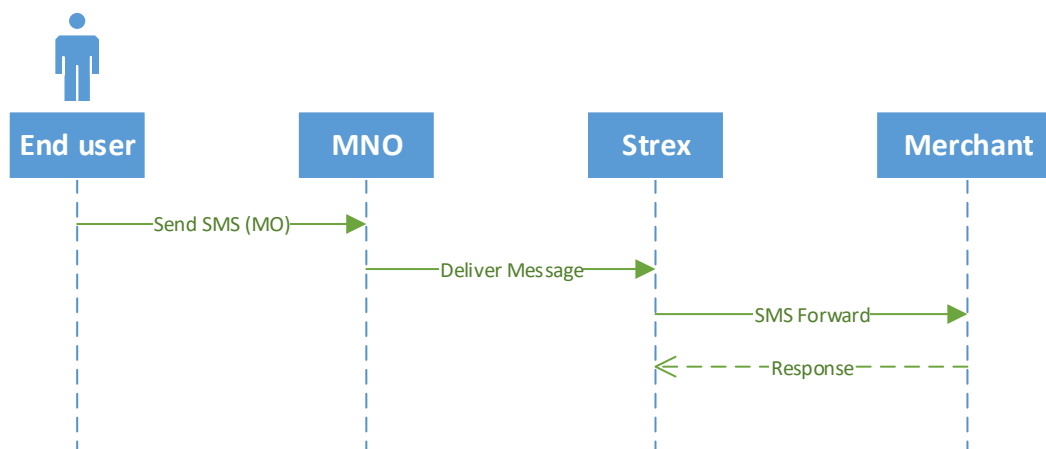
8.3.2 SMS Forward

The SMS forward function allows Merchants to receive SMS's sent to their shortnumber via Strex.

Please note that Strex supports subnumber of 10 digits also for MO SMS.

Also note that the namespace id's varies, please take this into account when implementing the server side.

In case of network issues the merchant might receive duplicate smsforward requests (multiple forwardSMS requests with same messageRequestId), make sure these are filtered to avoid duplicate charging requests.



Request Parameters

The only supported content type is "Content-Type:text/xml", add this to the header if needed.

Parameter	Description	Type	Optional
cpaHeader			
shortNumber	The shortnumber which the sms has been sent to	long(20)	No
messageRequestId	The unique message request ID	String(40)	No
smsHeader			
type	Hardcoded to "text"	String(10)	
requestBody			
message	The SMS message to forward to the Merchant	String(1024)	No
mno_ref	The MNO this message came from	String(20)	Yes
address	The End Users msisdn	String(16)	No

Response Parameters

Parameter	Description	Type	Optional
messageType	Always 'forwardSMS'.	String('forwardSMS', 'forwardMMS', 'deliveryNotification', 'refundNotification', 'chargeNotification', 'resend')	No
messageResponseId	Original request ID (messageRequestId)	string(40)	No
statusCode	statusCode describing status of delivery @merchant. "200" indicate success.	String(10)	No
statusDescription	Description for the statusCode	String(100)	No

Example request

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<forwardSMSNotification
xmlns="http://strex.no/MessageService/forwardSMSNotification/schema/v1"
xmlns:ns2="http://strex.no/MessageService/cpResponse/schema/v1">
<requestHeader>
  <cpaHeader>
    <messageRequestId>12345</messageRequestId>
    <shortNumber>2498</shortNumber>
```

```

        </cpaHeader>
        <smsHeader>
            <type>text</type>
        </smsHeader>
    </requestHeader>
    <requestBody>
        <address>4799999999</address>
        <mno_ref>netcom</mno_ref>
        <message>SMS text</message>
    </requestBody>
</forwardSMSNotification>
</S:Body>
</S:Envelope>

```

Example response

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-
ENV:Header/><S:Body><ns2:cpResponse
xmlns="http://strex.no/MessageService/forwardSMSNotification/schema/v1"
xmlns:ns2="http://strex.no/MessageService/cpResponse/schema/v1">
<ns2:message>
    <ns2:messageType>forwardSMS</ns2:messageType>
    <ns2:messageResponseId>12345</ns2:messageResponseId>
    <ns2:statusCode>200</ns2:statusCode>
    <ns2:statusDescription>OK</ns2:statusDescription>
</ns2:message>
</ns2:cpResponse></S:Body></S:Envelope>

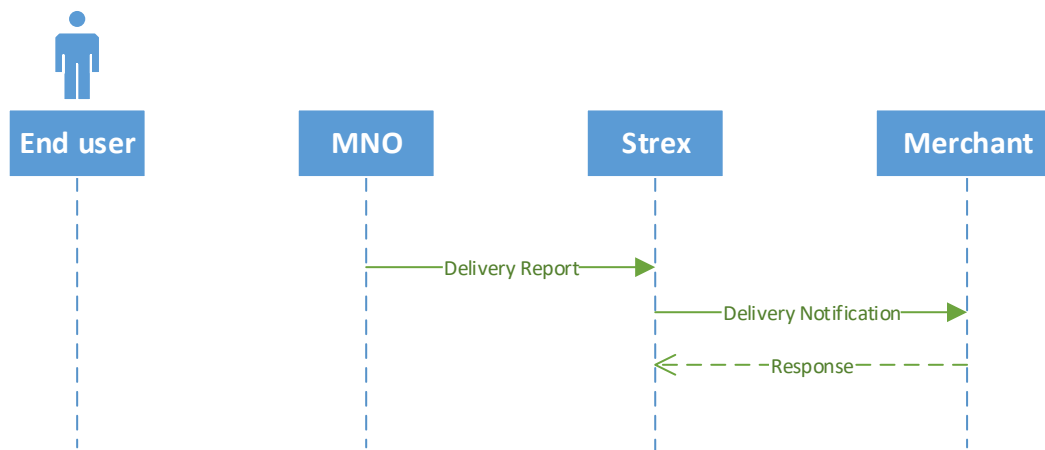
```

8.3.3 Delivery Notification

Delivery Notification informs Merchants about the delivery status of MT-SMS messages (SMS Payment and SendSMS), and status of the Sell transaction in case Easy Web Registration has been triggered.

Note: The Delivery Notification Flag in the Sell request only applies to the SMS delivery notification, in the case of a Sell trx that get's put on hold for Easy web registration, the Delivery Notification will always be sent.

Also note that the namespace id's varies, please take this into account when implementing the server side.



Request Parameters

The only supported content type is "Content-Type:text/xml", add this to the header if needed.

Parameter	Description	Type	Optional
requestHeader			
messageRequestId	The unique message request ID	String(40)	No
Initiator	The shortnumber which the sms has been sent to	long(20)	No
orgTrxID	Transaction ID from the originating Sell/Send SMS request initiated by Merchant.	String(40)	No
RequestBody			
address	The End Users msisdn	string(16)	No
mno_ref	The MNO this message came from	String(20)	Yes
statusCode	statusCode describing status of the SMS delivery.	String(10)	No
statusDescription	Description for the statusCode	String(100)	No

Response Parameters

Parameter	Description	Type	Optional
messageType	Always 'deliveryNotification'	String('forwardSMS', 'forwardMMS', 'deliveryNotification', 'refundNotification', 'chargeNotification', 'resend')	No
messageResponseId	Original request ID (OrgTrxId)	string(40)	No
statusCode	statusCode describing status of delivery @merchant. "200" indicates success.	String(10)	No
statusDescription	Description for the statusCode	String(100)	No

Example request

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
<SOAP-ENV:Header/>
<S:Body><deliverySMSNotification
xmlns="http://strex.no/MessageService/deliverySMSNotification/schema/v1"
xmlns:ns2="http://strex.no/MessageService/cpResponse/schema/v1">

<requestHeader>

    <messageRequestId>cre8-16a@-1264-id01</messageRequestId>

    <initiator>11110001</initiator>

    <orgTrxID>664111</orgTrxID>

</requestHeader>
<requestBody>

    <mno_ref>netcom</mno_ref>

    <statusCode>0</statusCode>

    <address>4799999999</address>

<statusDescription>Message Sent Successfully</statusDescription>
</requestBody>
</deliverySMSNotification>
</S:Body>
</S:Envelope>
```

Example response

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://strex.no/MessageService/cpResponse/schema/v1">
<SOAP-ENV:Header/>
<soapenv:Body>

    <v1:cpResponse>

        <v1:message>

            <v1:messageType>deliveryNotification</v1:messageType>

            <v1:messageResponseId>22101990</v1:messageResponseId>

            <v1:statusCode>200</v1:statusCode>

            <v1:statusDescription>Message
delivered</v1:statusDescription>

        </v1:message>

    </v1:cpResponse>

</soapenv:Body>
</soapenv:Envelope>
```

9 Appendixes

9.1 Appendix A Result Codes

9.1.1 Strex Payment Platform API Result Codes

Result Codes has been split in two sections to make implementation easier, Common result Codes and other resultcodes.

9.1.1.1 Common Result Codes

Result Codes in this section which have been used by Strex (only showing errors in the list, result code =0 indicate success for all requests). Result codes are valid for the corresponding namespace/table heading. The actual result codes received by the merchant will vary depending on their use of the API.

TRANSACTION TYPE	RESULT_CODE	EXPLANATION
generateOTP	10002	Subscriber does not have a valid subscription @ the MNO
generateOTP	66	Connection Offline
generateOTP	1008	Bad request to the core system, the request has failed. Please attempt the transaction again.
generateOTP	4110	OTP generated Successfully
generateOTP	1002	Transaction Recovered
generateOTP	4001	MNO NOT RESOLVED
login	3	The Agent attempting to access the system could not be found/
login	4	The Agent attempting to access the system must be registered before they can use any functionality
login	1012	Password error retry exceeded
login	1013	Password expired
login	1014	Password invalid
login	4001	MNO_NOT_RESOLVED
preauthorization	4	Agent Not Registered
preauthorization	581	One Time Password mismatch
preauthorization	1002	Transaction Recovered
preauthorization	4001	MNO NOT RESOLVED
preauthorization	4020	USER_CHANGED
preauthorization	4043	Rejected by subscriber
preauthorization	4044	Reached Max retries for USSD confirmation
preauthorization	4052	umarket_4052
preauthorization	4072	USSD_TECHNICAL_ERROR_RECEIVED_FROM_MNO
preauthorization	4074	USSD_ABSENT_SUBSCRIBER
preauthorization	4108	PreAuth not available for non-recurring OTP
sell	1	Bad Format
sell	7	Access Denied

sell	16	Target Suspended
sell	17	Invalid Amount
sell	20	Target Blacklisted
sell	24	Amount Too Big
sell	66	Connection Offline
sell	67	Connection Error
sell	70	Agent Has Transactions
sell	581	umarket_581
sell	1001	Insufficient Funds
sell	1002	Transaction Recovered (Transaction timeout, normally due to timeout between Strex platform and external party)
sell	1006	System error, the request has failed. Please attempt the transaction again.
sell	4001	MNO_NOT_RESOLVED
sell	4005	REGISTRATION_REQUIRED
sell	4006	CONFIRMATION_REQUIRED
sell	4009	AGENT_BARRED
sell	4013	AGE_UNKNOWN
sell	4014	AGE_NOT_ALLOW
sell	4020	USER_CHANGED (USSD confirmation)
sell	4040	Duplicate trans_ext_reference
sell	4041	Transaction not allowed for this amount
sell	4042	Monthly transaction limit exceeded
sell	4043	Rejected by End User
sell	4044	Reached Max retries for Sell USSD confirmation
sell	4045	Invalid UserName or Password for billing
sell	4046	Technical Error from MNO
sell	4047	SELL_REJECTED_DUE_TO_MAX_RETRY_PIN
sell	4051	Purchase with Insufficient Security Level Used in sell confirmation.
sell	4052	umarket_4052
sell	4053	Monthly transaction limit exceeded
sell	4062	umarket_4062
sell	4064	PRE_AUTHORIZE_NOT_MATCH_SELL
sell	4072	USSD_TECHNICAL_ERROR_RECEIVED_FROM_MNO
sell	4074	USSD_ABSENT_SUBSCRIBER
sell	4080	TRANSACTION_NOT_PURCHASABLE
sell	4118	STREX INVOICE ACCEPTANCE REQUIRED
sell	4129	Suppress Use Payment Card
sell	4133	No Preferred Card Linked to Subscriber
sell	10002	User Doesn't Exist @ MNO
sell	10004	End User reserved against charging @ MNO
sell	10005	MNO usage limit exceeded
sell	10006	Illegal operator for MSISDN
sell	10007	Insufficient credit @MNO

sell	10008	MNO was unable to process charge
sell	10009	MNO was unable to process charge
preauthorization	4	The Agent attempting to access the system must be registered before they can use any functionality
preauthorization	7	Access Denied
preauthorization	1002	Transaction Recovered (Transaction timeout, normally due to timeout between Strex platform and external party)
preauthorization	4063	ALREADY_PRE_AUTHORIZE_EXIST
reverse	1	Bad Format
reverse	17	Invalid Amount
reverse	36	Transaction is too old to complete
reverse	66	Connection Offline
reverse	75	Reversal Failed
reverse	4020	USER_CHANGED
reverse	4045	Invalid UserName or Password for billing
reverse	10008	MNO was unable to process charge
reverse	10018	user not found in external system
selfregisterV3	4001	MNO_NOT_RESOLVED
sendsms	1008	Bad request
sendsms	4001	MNO_NOT_RESOLVED
sendsms	1	Bad Format
sendsms	20	Target Blacklisted
sendsms	66	Connection Offline
sendsms	10002	User Doesn't Exist @ MNO
trans_info	1	Bad Format
trans_info	5010	SC2_TRANS_INFO_O_TRANS_NOT_FOUND
trans_info	5011	SC2_TRANS_INFO_INVALID_DATA

9.1.1.2 Common Functional result codes with explanation

Error code	Description	Explanation
580	One Time Password expired	One Time Password expired
581	One Time Password mismatch	One Time Password mismatch
4001	MNO_NOT_RESOLVED	The End Users MNO does not have an agreement with Strex/technical error, i.e. Strex could not identify which MNO the End User belongs to.
4002	MNO_CHANGED	The transaction could not be executed since the End Users subscription has been moved to another MNO. Mostly in use with reversal transactions.
4003	MSISDN_NOT_RESOLVED	Strex could not identify which MNO the End User belongs to
4005	REGISTRATION_REQUIRED	The End User must be registered with strex to successfully execute this payment
4006	CONFIRMATION_REQUIRED	This Payment requires USSD confirmation
4008		Technical error
4009	AGENT_BARRED	End User is reserved against charging ("CPA BARRED"). The End User must remove this barring at the MNO to execute payment successfully Suggested message to end user in Norwegian: Betalingen kunne ikke gjennomføres fordi ditt abonnement er sperret for mobilbetaling. Kontakt din mobiloperatør for å fjerne sperren." Message in English: «The Payment could not be completed because your mobile subscription is barred for payment. Please contact your Mobile Network Operator to remove the barring.»
4010		Not in use
4011		Not in use
4012		Not in use
4013	AGE_UNKNOWN	The End Users age is unknown. The End User must register at strex.no/minside to successfully execute this payment. Suggested message to End User: "Betalingen kunne ikke gjennomføres fordi alder ikke er registrert på ditt abonnement. Registrer deg på strex.no/reg eller kontakt din mobiloperatør for å registrere alder."

		Message in English: «The Payment could not be completed because your age is not registered on your mobile subscription. Please contact your Mobile Network Operator to register your age.»
4014	AGE_NOT_ALLOW	End User too young to execute payment successfully. Suggested message to End User: “Betalingen kunne ikke gjennomføres fordi alder registrert på ditt abonnement er for lav i forhold til aldersgrense for tjenesten som er kjøpt.” Message in English: «The Payment could not be completed because the age registered on your mobile subscription is below the required age for this payment.
4020	USER_CHANGED	USER_CHANGED (USSD confirmation) i.e. USSD confirmation comes from an other msisdn.
4040	Duplicate trans_ext_reference	The external reference trans_ext_reference must be unique for every request.
4041	Transaction not allowed for this amount	Invalid amount for this transaction/End User
4042	Monthly transaction limit exceeded	The End Users monthly payment transaction limit is exceeded.
4043	Rejected by subscriber	Purchase authorization from End User is either declined or absent.
4044	Confirmation_max_retries	Reached Max retries for Sell confirmation
4045	Invalid UserName or Password for billing	Error message «invalid username or password for billing” received form MNO. Usually means just that, or is returned in case of other unspecified MNO error.
4046	Technical Error from MNO	Technical error received from MNO on billing request.
4047	Technical error	SELL_REJECTED_DUE_TO_MAX_RETRY_PIN
4050	User requires Lower Security Level	User requires Lower Security Level
4051	Invalid combination of security settings	Invalid combination of security settings
4052	Confirmation disabled for ICE	ICE MNO does not support USSD. Error message returned when End User exceeds the monthly transaction limit and USSD registration is normally triggered.
4053	Monthly transaction limit exceeded	The End Users monthly payment transaction limit is exceeded.
4054	Monthly transaction limit exceeded under 18	The End Users monthly payment transaction limit is exceeded for End Users below age 18.

4055	STREX_INVOICE_DISABLED_FOR_ICE	ICE MNO does not support USSD. Error message returned when End User exceeds the monthly transaction limit and USSD registration is normally triggered.
4056-4061		internal use
4062	TRANSACTION_NOT_ALLOWED_BY_ARVATO	Strex invoice transaction is not allowed by vendor.
4063	ALREADY_PRE_AUTHORIZE_EXIST	Currently only one preauth id per merchant is allowed. Will be changed shortly.
4064	PRE_AUTHORIZE_NOT_MATCH_SELL	Preauth code supplied in sell does not match code stored on End User profile
4070	MISSING_FIELD	Missing mandatory input parameter
4071	MNO_UNABLE_TO_SEND_USSD_MESSAGE	MNO was unable to send USSD message
4072	USSD_TECHNICAL_ERROR_RECEIVED_FROM_MNO	End User pressed "cancel" in USSD dialogue (Telenor MNO) or MNO was unable to send USSD message due to a technical error
4073	USSD_BAD_REQUEST	MNO was unable to send USSD message due to syntax error in the request
4074	USSD_ABSENT_SUBSCRIBER	MNO was unable to send SSD message due to Absent End User, i.e. not registered in HLR (phone turned off or out of reach)
4075-4079		Not in use.
4080	TRANSACTION_NOT_PURCHASABLE	Strex invoice transaction failed at vendor.
4081	INVALID_REQUEST	INVALID_REQUEST
4088		Not in use
4089	TRIGG_FULL_USSD	Full ussd registration triggered, sell rolled back.
4090	AGE_VALIDATION_FAILED	End User too young to execute payment successfully. Suggested message to End User: "Betalingen kunne ikke gjennomføres fordi alder registrert på ditt abonnement er for lav i forhold til aldersgrense for tjenesten som er kjøpt." Message in English: «The Payment could not be completed because the age registered on your mobile subscription is below the required age for this payment.
4091	NEW_REG_FLOW_ROLLBACK	Registration rolled back. The End User must be registered at strex.no/minside to successfully execute this payment
4092	SELL_TRANS_IN_PROGRESS	Sell is put on hold and Easy Web

		Registration triggered.
4093-4098		Not in use.
4099	DSF_LOOKUP_FAILED	Sell response after triggering Full USSD registration, DSF lookup fails
4100	Sell trans in progress	Returned on getsellinfo request when the queried sell transaction is not completed.
4101	SUBSCRIBER_AGE_AT_MNO_BELOW_AGE_REQUIREMENT	Applies for sell transactions with age requirement. The MNO subscriber is below required age.
4103	Invalid Confirmation Channel	Invalid Confirmation Channel
4104	Invalid merchant_security_level or subscriber security level	Invalid merchant_security_level or subscriber security level
4105	Sell Already completed (SMS Confirmations)	Sell Already completed (SMS Confirmations)
4106	Invalid response from subscriber (SMS Confirmations)	Invalid response from End User (SMS Confirmations)
4107	Invalid OTP transaction ID	Invalid OTP transaction ID
4108	PreAuth not available for non-recurring OTP	PreAuth not available for non-recurring OTP
4109	Timeout is Invalid	Timeout is Invalid
4110	OTP generated successfully	OTP generated successfully
4111	Sell not available for recurring OTP	Sell not available for recurring OTP
4112	Invalid OTP	Invalid OTP
4116	SSN_BLACKLISTED	The End User is blacklisted
4117	SUPPRESS USE STREX INVOICE	Source of Fund Strex invoice is not allowed for this merchant
4118	STREX INVOICE ACCEPTANCE REQUIRED	Exceeded monthly limit set for MNO bill purchases or not allowed, End User need to register an alternative source of fund
4120	Error during PSD2 validation	Error during PSD2 validation
4129	Suppress use Payment Card	Source of Fund Payment card is not allowed for this merchant
4131	Service Code barred for Card Payments	Source of Fund Payment card is not allowed for this service code
4133	No Preferred Card Linked to Subscriber	Source of Fund Payment card is selected by End User, but no preferred card registered.
4134	Bad Request to PSP	Bad Request to PSP
4135	Subscriber registration required	Exceeded monthly limit set for MNO bill purchases or not allowed, End User need to register an alternative source of fund
4136	inputerror – received from card provider	PSP technical error
4137	forbidden – received from card provider	PSP technical error
4138	notfound – received from card provider	PSP technical error
4139	systemerror – received from card provider	PSP system error
4140		Exceeded monthly limit set for this source

		of fund
4141	Yearly Limit for Purchases Exceeded	PSD2 Yearly limit has been exceeded. Customer needs to be verified with BankID.
4142	Registration not allowed	Registration not allowed
4143	Subscriber BankID verification required	Customer needs to be verified with BankID.
4147	Rejected by acquirer - Invalid Amount	Rejected by Card Payment Acquirer.
4148	Rejected by acquirer - Format Error	Rejected by Card Payment Acquirer.
4149	Rejected by acquirer	Rejected by Card Payment Acquirer.
4150	Rejected by acquirer	Rejected by Card Payment Acquirer.
4151	Rejected by acquirer - Card Expired	Rejected by Card Payment Acquirer.
4152	Rejected by acquirer	Rejected by Card Payment Acquirer.
4153	Rejected by acquirer	Rejected by Card Payment Acquirer.
4154	Rejected by Payex	Rejected by Card Payment Acquirer.
4155	Acquirer Host Offline	Card Payment Acquirer offline
4156	PSP was unable to Process the request	Technical error returned from Card Payment PSP.
4157	Rejected by acquirer - Invalid Amount	Reverse transaction - Rejected by Card Payment Acquirer.
4158	Rejected by acquirer - Format Error	Reverse transaction - Rejected by Card Payment Acquirer.
4159	Rejected by acquirer	Reverse transaction - Rejected by Card Payment Acquirer.
4160	Rejected by acquirer	Reverse transaction - Rejected by Card Payment Acquirer.
4161	Rejected by acquirer - Card Expired	Reverse transaction - Rejected by Card Payment Acquirer.
4162	Rejected by acquirer	Reverse transaction - Rejected by Card Payment Acquirer.
4163	Rejected by acquirer	Reverse transaction - Rejected by Card Payment Acquirer.
4164	Rejected by Payex	Reverse transaction - Rejected by Card Payment Acquirer.
4165	Acquirer Host Offline	Reverse transaction - Card Payment Acquirer offline
4166	PSP was unable to Process the request	Reverse transaction - Technical error returned from Card Payment PSP.
4169	FAILED_BY_PAYMENT_RESTRICTION_MATRIX_VALIDATION	End User has no available Payment Source for this purchase (i.e. CPA barred, and alternative payment source is not registered)
4170	PSD2_SERVICE_CODE_IS_LICENSED	GetUserInfov2, End User is unregistered and Service_code is licensed
4171	PSD2_IN_BYPASS_GROUP_BUT_SUSPENDED	GetUserInfov2, End User is suspended
4172	PSD2_AMOUNT_VIOLATE_LIMIT	GetUserInfov2, amount over PSD purchase limit
4173	PSD2_SUBSCRIBER_SUSPENDED	GetUserInfov2, End User is suspended

4174	PSD2_CAPSET_EXCEEDED	GetUserInfo2, amount over PSD monthly purchase limit
4183	IMEI_IS_BLACKLISTED_ACCESS_DENIED	Handset (IMEI) is blacklisted.
4184	SECURITY_LEVEL_NOT_ALLOWED	The security_level setting in incoming request is not allowed
4214	BankID error	BankId returned an error
4215	The end-user entered an incorrect mobile number or date of birth	There is no BankID on mobile registered for the given MSISDN/date of birth combination
5000		Not in use.
5001		Not in use
5002	GETOK_REJECTED_BY_SUBSCRIBER	Sell response, Easy USSD registration triggered, End User rejects registration confirmation
5003	GETOK_REJECTED_DUE_TO_MAX_RETRY_PIN	Sell response, Easy USSD registration triggered, max retry count exceeded for End User registration confirmation
5004	GET_OK_FULL_CONFIRM_REJECTED_DUE_TO_BLANK_RESP	Not in use
5005	GET_OK_FULL_REJECTED_BY_SUBSCRIBER	Sell response, Full USSD registration triggered, End User rejects registration confirmation
5006	GET_OK_FULL_REJECTED_DUE_TO_MAX_RETRY_PIN	Sell response, Full USSD registration triggered, max retry count exceeded for End User registration confirmation
5007	USSD_DISABLED	Sell response, USSD disabled (or no support) for MNO
5008	GETSSN_REJECTED_DUE_TO_MAX_RETRY_PIN	Sell response, Easy USSD registration triggered, max retry count exceeded for End User SSN input
5009	SELF_REGISTRATION_V3_SUCCESS_RESPONSE	Not in use
5010	SC2_TRANS_INFO_O_TRANS_NOT_FOUND	Getsellinfo response, supplied sell transid not found
5011	SC2_TRANS_INFO_INVALID_DATA	Getsellinfo response, more than one reference is supplied and there is a mismatch. For example transid and merchant_transid does not refer to the same transaction
5012	SC2_USER_INFO_INVALID_DATA	Getuserinfo response, mandatory parameters missing in the request.
5013	INITIATOR_NOT_MATCHED	Getsellinfo response, Initiator of sell is not the same as initiator of getsellinfo
5016	SUBSCRIBER_IS_NOTREGISTERED	
10002	User Doesn't Exist @ MNO	End User does not have a valid subscription @ the MNO
10003	System error received from MNO	System error received from MNO on billing

		request.
10004	Subscriber reserved against charging @ MNO	The MNO billing request have failed for this subscriber. Some possible reasons: • Subscription is barred @MNO • Other reservations or barring The subscriber can contact his MNO to resolve this issue. Suggested message to End User: «Betalingen kunne ikke gjennomføres fordi ditt abonnement er sperret for mobilbetaling. Kontakt din mobiloperatør for å fjerne sperren.» Message in English: «The Payment could not be completed because your mobile subscription is barred for payment. Please contact your Mobile Network Operator to remove the barring.»
10005	MNO usage limit exceeded	The End Users usage limit @MNO has been exceeded. Subscriber need to contact MNO to remove usage limit or wait until usage limit is reset. Suggested message to End User: “Betalingen kunne ikke gjennomføres på grunn av at beløpsgrense for mobilbetaling på ditt abonnement er nådd. Kontakt din mobiloperatør.” Message in English: «The Payment could not be completed because the maximum amount limit on your mobile subscription has been reached. Please contact your Mobile Network Operator.»
10006	Illegal operator for MSISDN	Wrong network operator registered for this subscriber
10007	Insufficient credit @MNO	Pre-paid card does not have sufficient credit balance to complete the purchase. Suggested message to End User: “Betalingen kunne ikke gjennomføres på grunn av begrensninger på ditt abonnement. Kontakt din mobiloperatør. “ Message in English: «The Payment could not be completed because of limitations on your mobile subscription. Please contact your Mobile Network Operator.”

10008	MNO was unable to process charge	Default error message for transactions failing @ the MNO.
10009	MNO was unable to process charge	MNO returned the error message "MNO was unable to process charge"
10010	Transaction already credited	MNO returned "transaction already credited" on reverse request
10011	Unknown transaction id	MNO returned "unknown transaction id" on reverse request
10050	LIMIT_REACHED	Not in use

9.1.1.3 Result Codes

Reference list of Result Codes defined on the API. Result codes are valid for the corresponding namespace/table heading. Please note that some result codes are valid for several namespaces (such as 1002 – Transaction recovered (timeout))

Session

Code	Description
1	Session Required
2	Session Invalid (client details mismatch - such as clients ip address)
3	Session Not found/Session Expired

SOAP

Result Codes	Description
1	Session Required. When no "session id" is found in the incoming HTTP request
2	Client Details Mismatch (source IP address, client's user agent or clients transport protocol changes). For example, login for a session ID "XYZ" was done from an IP address 10.192.168.3 but a sell request came from a different IP address.
3	Session Not found/Session Expired. When the session ID in request is not found
100	Operation Not Supported by Strex
101	No Result Generated. Strex internal timeout
102	Bad Result.
1002	Transaction recovered

CORE

Result Codes	Description
--------------	-------------

0	Success
1001	Insufficient funds
1002	Transaction recovered
1003	Wallet balance exceeded
1004	Wallet cap exceeded
1005	Request expired
1006	System error
1007	Database error
1008	Bad request
1009	Database constraint error
1010	Record not found
1011	Password or agent not found or invalid
1012	Password error retry exceeded
1013	Password expired
1014	Password invalid
1015	Change password - password is the same
1016	Change password - password has been used previously
1022	Parameter Invalid: Too Long

UMARKET

Note that the result codes in Core (from 1000 to 1999) can also occur for UMarket result namespace. Most of them are not in use.

Result Codes	Description
0	Success
1	Badly formatted request.
3	The Agent attempting to access the system could not be found
4	The Agent attempting to access the system must be registered before they can use any functionality
5	The Agent is inactive and needs to contact Customer Service
6	The Agent is suspended and needs to contact Customer Service
7	Access is denied for this function to the requesting Agent
8	Agent supplied an incorrect password
9	Agent's password has expired. They must change to a new one
10	Agent's new password has been previously used and has been rejected
11	Agent has made too many attempts to log in with a bad password. They need to contact Customer Service to get it reset
12	The new password supplied is the same as the old one and has been rejected
13	The target Agent in a transaction does not exist
14	The target Agent in a transaction is not registered as yet
15	The target Agent in a transaction is inactive
16	The target Agent in a transaction is suspended
17	The amount requested is invalid
18	Agent is already registered
19	Requesting Agent has been blacklisted. Please contact Customer Service
20	Target Agent has been blacklisted. Please contact Customer Service

22	The target Agent is the same as the requesting Agent
23	Amount requested is too small
24	Amount requested is too large
25	The wrong amount was supplied in a confirm command
26	No target Agent was supplied
27	The requesting Agent has expired
28	The target Agent has expired
29	User supplied an invalid password
30	Agent is already active
32	The original request has expired due to a timeout
33	The transaction in question has been reversed
34	Transaction is not a two party transaction
35	Requested Agent status change is invalid
36	Transaction is too old to complete
37	Transaction was not completed
38	Agent has been stopped
39	Target Agent has been stopped
40	Agent is already started
41	IN/Topup system returned a system error
42	IN/Topup system returned rejected the target MSISDN
43	IN/Topup system was unavailable
44	IN/Topup system returned a bad vouchererror
45	IN/Topup system timed out the request
46	Amount requested was not within a valid range
47	Target Agent is not a special Agent
48	The supplied Billpay account is not valid
49	The Bank is currently offline
50	The Bank returned an error
51	Synchronization is not required at this time
52	The requested device is already active
53	No configuration update is required
54	The requesting Agent is not the owner of the device
55	An alias of this name already exists
56	Alias supplied was not found
57	Agent is already suspended
58	Agent is already stopped
59	Agent cannot use the requested wallet type
60	Schedule has been disallowed
61	Schedule supplied was invalid
62	Schedule requested was not found
63	Requested confirmation would potentially match multiple transactions
64	Requesting Agent is not a Special Agent
66	Connection is offline/network issue. An external party could not be reached.
67	Connection Error/network issue. An external party could not be reached.
68	Operation not supported
69	Agent has been misconfigured
70	Agent still has transactions outstanding
71	Agent is deleted

72	Target is deleted
73	Agent still has a non-zero balance
74	Invalid password mismatch
75	Reversal Failed
76	Group name already exists
77	Group not empty
78	Confirmer same as initiator
103	Confirmation was rejected
112	Technical error
250	Confirm transfer
251	Confirm cashout
252	Confirm sell
253	Confirm Billpay
255	Confirm link
256	Confirm join parent
257	Confirm join child
312	Modify wallet ok
395	There are no pending confirmation transactions for this agent.
400	Requested transaction cannot be found
401	Transfer confirm rejected
415	Trace transaction was not found
423	Transaction has already been committed
424	Transaction has already been rolled back
425	Message was too long
501	Cannot link to requested Sub-Agent
502	Cannot link to yourself
503	The linked Agent has been deleted
504	Cannot join to self
505	Agents already joined
506	Join agent is deleted
507	Agents not joined
508	Purchase order number not unique
509	Rebate amount not exact
510	Rebate amount not multiple
511	Rebate amount too small
520	Voucher recipient not found
521	Voucher expiry date passed
522	Voucher does not exist
523	Non-deleted voucher id or provider id exists
524	Voucher is not in a deletable state
525	Matching voucher not found
530	Map agent group listener failed
540	Coupon write failed
541	Coupon in progress
542	Coupon does not exists
543	Invalid coupon user
544	Coupon funds temporarily unavailable
545	Coupon cancelled

546	Coupon ID exists
547	Multiple coupons exist
548	Coupon password retry exceeded
549	Coupon not blocked
550	Initiator attribute is missing
551	Debtor attribute is missing
552	Creditor attribute is missing
560	Generic retry
574	Coupon password mismatch
575	Coupon fail attempts mismatch
576	Coupon shared secret missing
577	Coupon shared secret mismatch
580	One Time Password expired
581	One Time Password mismatch
600	Target is not invocable
700	External insufficient funds
701	External excess funds
850	Rule access denied
900	Invalid post date
901	Undefined wallet type
902	No account product
903	Must be batch administrator
904	Interest calculation failed
905	Next posting date undefined
1002	Transaction recovered
3011	Invalid secret length
3012	Invalid secret word
5001	Account does not exist
5002	Account product does not exist
5003	Account product cannot be deleted
5004	Product name does not exist
5005	Interest balance type does not exist
5006	Interest balance type bad format
5007	Interest accrual interval does not exist
5008	Interest accrual units does not exist
5009	Interest posting interval does not exist
5010	Interest accrual units bad format
5011	Interest posting units does not exist
5012	Interest posting units bad format
5013	Interest source agent does not exist
5014	Low balance rate does not exist
5015	Low interest rate does not exist
5016	Transfer rate general error
5017	Range rate error
5018	Account with product
5019	Product accounts query error
5020	High balance is less than low balance
5021	Accrual date is larger than posting date error

6054	Pickup is rejected
6071	Accept not enrolled
6200	Target is not invocable
6201	Remittance Hub confirmation
6202	Agent data object ID exists
6203	Agent data object ID does not exist
6204	Agent data properties does not exist
6205	Agent data properties exists
6206	Address book is empty
6207	Nickname does not exist
6208	Invalid nickname length
6211	Maximum nickname exceeded
6270	Transaction not found
6271	Additional data required
6272	Transaction required
6273	Transaction rejected
6274	Transaction already paid out
6275	Amount requested in transaction is more than maximum
6276	Daily limit exceeded
6277	Monthly limit exceeded
6278	Mobile pickup locked
6279	Mobile pickup is not ready
6280	Transaction status unknown
6281	Transaction on hold
6282	Missing request data
6283	XML error
6284	Remittance Hub error
6285	Invalid promotion code
6286	Invalid country code
6287	Invalid phone number
6288	Invalid data
6289	Incomplete validation
6290	Money currently locked
6291	Task manager error
6292	Invalid phone service
6293	Mobile carrier not registered
6294	Service fee not available
6298	Transaction cannot be validated
6300	Pickup not available
6301	Invalid sent date
6302	Invalid expected payout amount
6303	Invalid country ISO
6400	MTCN not found
6401	Inconsistent external transaction
6402	No longer required
6403	Invalid MTCN
6405	Invalid response
6406	Invalid Umarket transaction ID

6407	Not Remittance Hub transaction
6408	Enrollment has expired
6501	Next reminder required
6502	Final reminder
6600	Agent not enrolled
6602	Reject not enrolled
6603	Insufficient enrolment details
6700	Empty transaction ID
6701	Empty amount
6702	Invalid fee
6703	Invalid currency code
6704	Missing temporary transaction ID
6705	Invalid originating payment
6706	Invalid destination payment
7000	Invalid feature name
8000	Exchange initiation failed
8100	Exchange execution failed
8200	Exchange connection failed
21000	Start external errors

9.1.2 Delivery Notification Result Codes (Strex → Merchant)

Most common statuscode are highlighted in **bold**.

Status Code	Description	Explanation
0	Success	Successful delivery of MT-SMS message. This is the only success case for delivery of MT-SMS.
10000	Charge Pending	Applies for Direct Payment and Telenor prepaid only. Not in use.
10001	Charge Pending, Message Sent	Applies for SMS Payment and Telenor prepaid only. Not in use.
10002	Charge Successful.	Applies for Direct Payment and Telenor prepaid only. Not in use.
10003	Charge Successful; Message Sent.	Applies for SMS Payment and Telenor prepaid only. Not in use.
10004	Message Sent Successfully.	Duplicate. Not in use
10005	Refund Pending.	Applies for reverse transactions and Telenor only
10006	Refund Successful.	Applies for reverse transactions and Telenor only
10007	Charge Successful, Partial Charging	Applies for Direct Payment and Telenor prepaid only. Not in use.
10008	Charge Successful, Full charge	Applies for Direct Payment and Telenor prepaid only. Not in use.
10009	Charge Uncertain, Message Sent	Applies for SMS Payment and Telenor prepaid only. Not in use.
10010	Charge Uncertain	Applies for Direct Payment and Telenor prepaid only. Not in use.

10020	Sell In Progress	Sent after secure web registration process is initiated during sell, and the End User have been registered.
10021	Sell Rolled Back	Sent after secure web registration process is initiated during sell, and Sell have failed (timeout).
10022	Sell Completed	Sent after secure web registration process is initiated during sell, and the End User have been registered, and the Sell transaction is successfully completed.
11000	InvalidDestinationAddress	MT-SMS not delivered to handset. Destination Address (receiver) set in request is invalid.
11001	InvalidOriginatingAddress	MT-SMS not delivered to handset. Originating Address (sender) set in request is invalid.
11002	InvalidLargeAccount	MT-SMS not delivered to handset. Large Account set in request is invalid.
11003	InvalidNumberSeries	MT-SMS not delivered to handset. The request failed because the number does not belong to a recognized number series as assigned by NPT (Norwegian Post and Telecom Authority).
11004	IllegalOperator	MT-SMS not delivered to handset. The request failed because the subscriber belongs to another MNO.
11005	InvalidPayload	MT-SMS not delivered to handset. Message is rejected by SMSC
11006	Unspecified error '[error description]'	MT-SMS not delivered to handset due to error in Description column. System error received from MNO
11007	Deleted	MT-SMS not delivered to handset due to error in Description column.
11008	Expired	MT-SMS not delivered to handset within validity time.

9.2 Appendix B Pin calculation

Pin used in the login request is calculated using SHA1.

Example: Pin= sha1 (sessionid + sha1(userid + password))

Where:

sessionid is the sessionid returned in createsession response

userid is your Strex userid provided by Strex

password is your password provided by Strex

The result of sha1(userid + password) must be kept in lowercase before calculating the second sha, but the pin itself should be converted to uppercase before it is sent to Strex.

The script below can be used on a unix/linux machine to verify the calculation of pin for login request.

```
#!/bin/sh
if [ $# -ne 3 ]
then
    echo "Usage: $1 <sessionid> <user_id> <password>"
else
    printf $1$(printf $2$3|shasum|awk '{print $1}')|shasum|tr a-z
A-Z|awk '{print $1}'
fi
```

9.3 Appendix C Service Codes

Service Codes apply both for production and test environments.

Service Code	Strex registration required	Description	Example	Invoice Text
01001	No	Mobile Internet and media (not overcharged)		Mobilt internett og media
02001	No	SMS information services	Information service sent to or from the End User via SMS or other electronic information channel (including e-mail, web page or app).	Informasjonstjeneste
02002	No	MMS information services	Information services sent to or from the End User. For example a number directory service, traffic information, dentist appointment notification, weather information, information about cancelling of services	MMS
02003	No	Instant messaging		Mobil IM
02004	No	e-mail		Mobil epost
03001	No	Ringtone		Ringetone
03002	No	Audio download	For example music or audio book	Musikk
03003	No	Game for download or used in web browser		Spill
03004	No	Picture for download		Bilder
03005	No	Audio streaming - NOT IN USE		Lyd streaming
03006	No	Not in use		Lyd streaming
03007	No	Video for download		Video
03008	No	Video streaming - NOT IN USE		Video streaming
03009	No	Not in use		Video streaming
03010	No	Digital chat service		Chat
03011	No	Community		Community
03012	No	Competition based on digital platforms	For example quiz services like IQ test.	Konkurranse
04001	No	Digital directory services	For example SMS number directory like 1881 or 1880.	Katalogtjenester

04002	No	Tv/radio interaction	For example voting in TV shows, interaction with radio host, Idol, påskesnøtter and traffic tips sent in by listener.	TV/radio interaksjon
04003	No	Automatic number directory services - NOT IN USE		Automatisk nummeroppslag
05001	No	Theatre ticket distributed and available on digital device		Teaterbillett
05002	No	Cinema ticket distributed and available on digital device		Kinobillett
05003	No	Concert ticket distributed and available on digital device		Konsertbillett
05004	Yes	Physical book		Bok
05005	Yes	Audio book on CD		Lydbok - CD
05006	Yes	Film dvd and blue ray disc		Film - DVD
05007	Yes	Music CD		Musikk - CD
05008	Yes	Newspaper on paper		Avis
05009	Yes	Magazine on paper		Magasin
05010	N	Event ticket distributed and available on digital device		Arrangement sbillett
06001	No	Ticket distributed and available on digital device		Billett
06002	No	Bus ticket distributed and available on digital device		Bussbillett
06003	No	Metro/tram/train ticket distributed and available on digital device		T-bane/ trikkebillett/ togbillett

06004	Case specific, contact Strex Partner Support	Taxi fare	Payment for personal taxi transport	Taxi
06005	No	Parking ticket distributed and available on digital device		Parkering
06006	No	Ferry ticket distributed and available on digital device		Fergebillett
06007	N	Public road charges ticket distributed and available on digital device	For example piggdekkavgift, toll road ticket	Veiavgift
06008	Yes	Car operation and maintenance services	For example car wash service, car maintenance service	Bil drift og vedlikehold
06009	N	Airplane ticket distributed and available on digital device	For example Ryan Air ticket	Flybillett
07001	Yes	Healthservice, non-digital	For example physician, physical therapist service, chiropractor service, medical prescription service.	Helsetjeneste
07002	Yes	Medicines	For example paracetamol tablets.	Medisiner
07003	Yes	Public fees - medical treatment	For example patient charge.	Off. avgifter medisinsk behandling
08001	No	Classified ads published on digital platform	For example finn.no classified ad	Annonser
08002	Yes	Classified ads published on non-digital platform	Aftenposten on paper,	Annonser
09001	Yes	Gift certificate	For example Universal Presentkort	Gavekort
09002	Yes	Insurance	For example travel insurance.	Forsikringstjeneste
09003	Yes	Peer to peer payment		Betaling
10001	Yes	Food and drink	For example Coca cola bottle	Mat og drikke
10002	Yes	Food and drink - Physical goods -		Mat og drikke

		NOT IN USE		
11001	Yes	Food and drink - restaurants	For example serving in restaurant	Servering
12001	Yes	Lodging	For example hotel room.	Overnatting
13001	N	Public fishing licence - "Fiskeravgift" distributed and available on digital device	Fiskeravgift service	Fiskeravgift
13002	N	Private fishing licence distributed and available on digital device	For example fishing licence paid to Lågen Grunneierlag	Fiskekort
13003	N	Facility area access fee distributed and available on digital device	For example airport lounge, tennis court rent, solarium, public toilet etc.	Tilgangsavgift
13004	No	Consulting/Coaching, digital or voice telephony based	For example consulting by lawyer, physician, IT support	Rådgiving
13005	No	Access fee television services - NOT IN USE		Tilgangsavgift TV
13006	Yes	Manual services	Hairdresser service, carpenter service, plumber service, nail service	Servicetjeneste
13007	Yes	No show manual services	No-show for Manual service	Ikke fremmøte
13008	Yes	Consulting/Coaching, non-digital	For example consulting by lawyer, physician, IT support	Rådgiving
14001	No	Membership Fee distributed and available on digital device, non-commercial	Fee for membership for non-commercial organisations, for example Røde Kors, Arbeiderpartiet, sports clubs under Norsk Idrettsforbund (NIF)	Medlemsavgift
14002	No	Donation - Donation	Payment to non-commercial organisations, no return service of significant value.	Givertjeneste
14003	No	Donation - NOT IN USE (use 14002)	NOT IN USE	Givertjeneste

14004	Yes	Membership Fee distributed and available on digital device, commercial	Fee for membership in commercial organisations, for example SATS, NAF.	Medlemsavgift
15001	Yes	Physical goods - non food		Fysiske varer
15002	Yes	Physical goods - non food - NOT IN USE		Fysiske varer
15003	Yes	E-commerce - STREX INTERNAL CODE		Netthandel
16001	No	Audio streaming distributed and available on digital device	For example Spotify service	Audio streaming
16002	No	Video streaming distributed and available on digital device	For example Netflix service	Video streaming
16003	No	e-book distributed and available on digital device	For example eReadz, Kindle	e-bok
16004	No	e-magazine distributed and available on digital device	For example flipp.no, vio.no	e-magasin
16005	No	e-newspaper distributed and available on digital device		e-avis
16006	No	Internet music - NOT IN USE		Nett-musikk
16007	No	Digital education services distributed and available on digital device	For example Teoritentamen, prøve.no.	Digital undervisnings tjeneste
16008	No	Automatic number directory services	Services that automatically looks up information (such as name) connected to the originating (A) number during an incoming telephone call on a mobile terminal.	Automatisk nummeroppslag
89001	No	Zero charged Information Message		Nulltaksert informasjons melding

9.4 Appendix D Business Models

Please refer to the Merchant Agreement front page for the Business Model(s) that apply for your Merchant. Business Models apply for both test and prod environment.

Aggregators can find a list of business models in the Appendix 3 of the Strex partner Agreement.

9.5 Appendix E WSDL

The WSDL file is too large to append to this document. WSDL for incoming and outgoing requests is available upon request to technicalsupport@strex.no.

9.6 Appendix F Java Sample code

Sample code, using Java, is provided as follows. Note that this code is based upon the Apache AXIS SOAP libraries. See <http://ws.apache.org/axis/> for more information.

```
import mPaymentWS.*;
import java.net.*;
import java.security.MessageDigest;

public class Example
{
    public static void msin(String args[]) throws Exception
    {
        Example e = new Example();
        e.run(args[0],args[1],args[2],args[3]);
    }
    public void run(String url,String [merchant_reference],String
salt,String pin) throws Exception
    {
        URL url = new java.net.URL(url);
        mPaymentSOAPStub stub = new mPaymentSOAPStub(url,null);

        CreatesessionResponseType crt = stub.createSession();
        if (crt.getResult()!=0) throw new Exception("Bad Result");

        LoginRequestType l_req = new LoginRequestType();
        l_req.setSessionid(crt.getSessionid());
        l_req.setInitiator(msisdn);
        l_req.setPin(getSHA(crt.getSessionid() +
getSHA(salt+pin).toLowerCase()));

        LoginResponseType l_resp = stub.login(l_req);
        if (l_resp.getResult()!=0) throw new Exception("Bad Result");
    }

    private static char[] hexits="0123456789ABCDEF".toCharArray();

    public String getSHA(String data) throws Exception
    {
        MessageDigest md = null;
        byte [] ba = null;
        md = MessageDigest.getInstance("SHA");
        md.update(data.getBytes("UTF-8"));
        ba = md.digest();

        StringBuffer sb = new StringBuffer(ba.length*2);
        for (int i = 0; i < ba.length; i++) {
```

```
        sb.append(hexits[(((int)ba[i] & 0xFF) / 16) & 0x0F]);
        sb.append(hexits[(((int)ba[i] & 0xFF) % 16)]);
    }
    return sb.toString();
}
```

9.7 Appendix G SMS confirm screenshot examples

Please note that the texts in screenshots are examples and are subject to change.

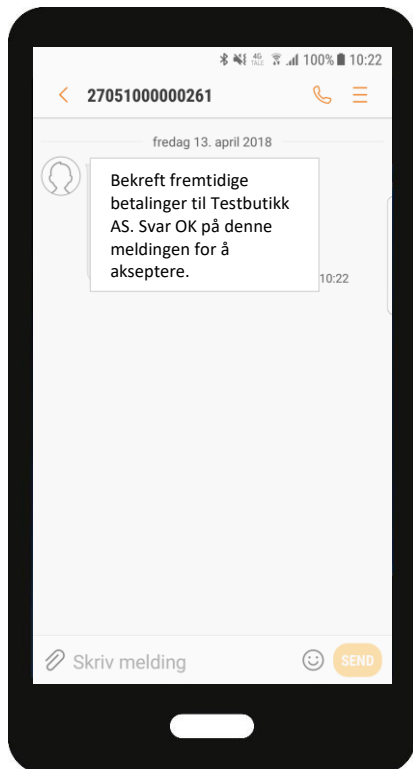
9.7.1 Single payments

SMS message translated to English: "Complete the purchase at Test Merchant AS. Answer OK on this message to accept."



9.7.2 Preauthorized and recurring payments

SMS message translated to English: "Confirm future payments at Test Merchant AS. Answer OK on this message to accept."



9.8 Appendix H Premium Rate Services (Teletorg)

Premium Rate Services can be charged with Strex, this chapter provides a brief description of the solution.

If a transaction for Premium Rate Services exceeds the permitted single transaction amount, a voice message must be presented in the call notifying the end user that the amount limit for a single transaction has been reached and that a new transaction will be started. The end user will receive a receipt from Strex on each transaction.

The voice message must be: "Du har nådd grensen for engangskjøp på kr. [amount*]. Ved å fortsette samtalen starter du et nytt engangskjøp. Du blir kun belastet for faktisk samtaletid."

* [Amount] from «[current amount limits](#)», depending on customer status

Strex recommended process:

1. Check end-user amount limits with `getuserinfov3`.
2. Check if the amount can be charged the end user with `getuserinfov2`.
3. If there is a need for the end user to be registered (when selling licensed goods or higher amount) the registration level can be checked further with `getuserinfo (v1)`.
 - a. Registration can be started by using `self-registerV3`
4. Perform Sell request

9.8.1 Brief description of the api requests

For details, see the respective chapters in the implementation guide.

[GetUserInfov3](#)

- Retrieves the end user's amount limits from Strex at the time the request arrives. The amount limits depend on the customer's registration level with Strex and age (over / under 18 year)
 - Single transaction
 - Aggregated monthly
 - Aggregated annually
- Also provides information on whether the subscription is postpaid or prepaid and end user preferred payment method (MNO invoice or payment card)
- Prepaid balance and content amount limits with MNO is checked for selected merchants
- Balance on payment card is not checked

9.8.2 Current amount limits (subject to change).

All amounts in NOK:

	Unregistered end user		Registered end user		Under 18 unregistered		15-18 year registered	
	MNO invoice	Payment card	MNO invoice	Payment card	MNO invoice	Payment card	MNO invoice	Payment card
Single transaction	500	-	1000	5000	500	-	-	1000

Aggregated monthly amount	3000	-	5000	10000	1000	-	-	1000
---------------------------	------	---	------	-------	------	---	---	------

[GetUserInfo2](#)

- Checks whether a transaction can be charged to the end user based on input (amount, service code and age). NB! Gives only an **indication** of the result of sell with the same input. No charge is done against MNO.
- The response will be "OK" or an error message that provides information about why the customer cannot be charged (e.g. CPA block, customer registration required)

[GetUserInfo](#) (v1)

- Checks if an end user is registered with Strex. The end user must be registered to be able to make licensed purchases (physical goods or amounts above a certain amount)
- Only registered end users can add a payment card as a means of payment

[SelfregisterV3](#)

- Sends an SMS to the end user with a link to Strex registration. The merchant can add an explanatory text in the message.

[Sell](#)

- Parameter invoice_text in the Sell request must contain the b-number at the beginning of the parameter value.
- Teletorg section must be filled in with b_number, start_time, duration and start_price

9.9 Appendix I - Regulated services for card and credit agreement payments

If a payment card is used as payment method some sales are subject to restrictions set by the acquirer (bank). If your company belongs to an industry with such restrictions the parameter "payment_source_restr" in the Sell request must be set to value "true". The following service/product/industry is subject to restrictions as mentioned above, thus value must be set to "true":

Narcotic substances, guns, tobacco, "snus" etc.
Merchandise with regulated sales restrictions, e-commerce for Health preparations, medicines, alcohol etc.
Travel agencies
Airplane tickets
Gambling and lottery
Digital wallets
Discount/membership clubs with subscription models

Pornography
Pyramid scheme or multi-level marketing
Charity and help organisations which does not meet the requirements in «Forskrift om elektronisk kommunikasjonsnett og elektronisk kommunikasjonstjeneste §4-6(6)»