Configuration and Application Programming Interface (API) Guide for Web OTP Module for 2-factor authentication process

Prepared by

TalariaX Pte Ltd
76 Playfair Road
#08-01 LHK2
Singapore 367996
Tel: 65-62802881
Fax: 65-62806882

CONFIGURATION AND APPLICATION PROGRAMMING INTERFACE (API) GUIDE FOR WEB OTP MODULE FOR 2-FACTOR AUTHENTICATION PROCESS

1. INTRODUCTION

The sendQuick Web OTP Module is designed for all applications who wish to have the two factor authentiction (2FA) capability without the hassle of developing the codes for OTP generation, authentication and verifications. The Web OTP consists of a pair (OTP Generation and OTP Verification) of Application Programming Interface (API) that can be utilised for any applications for any types of transactions.

The Web OTP module is an add-on function and can be added to sendQuick Alert Plus, sendQuick Entera and sendQuick Conexa.

2. WEB OTP CONFIGURATION

The Web OTP need to be configured for it to accept and authenticate the individual applications. These applications are called Client. Login to the Web OTP administration URL (http://sendQuickIP/webotp/). Login as 'otpadmin' and password using 'admin123'.

Once you have access, create a new Client (as shown in Figure 1 below). Complete the form as shown in Figure 2 and the configuration will complete.

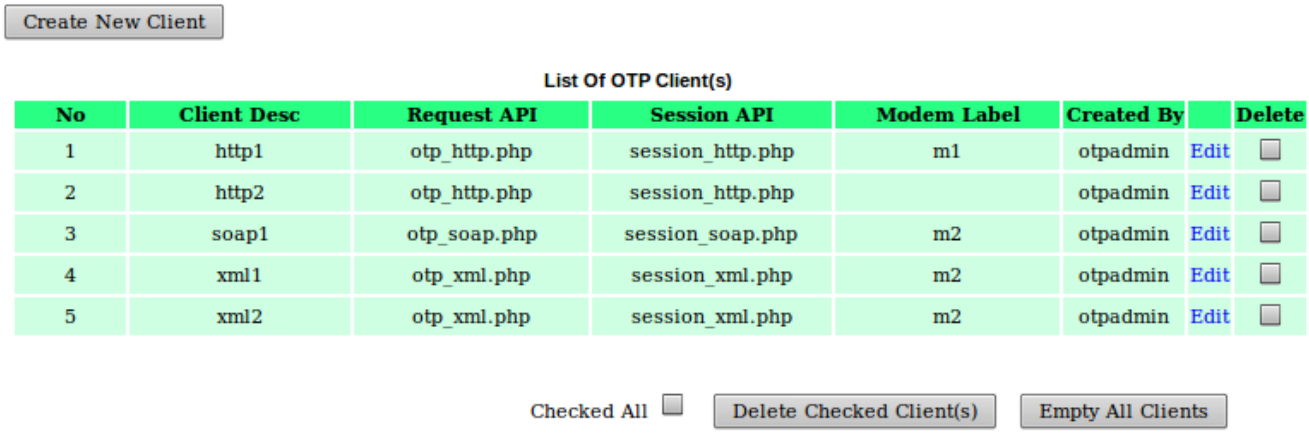


Figure 1: Web OTP Clients Configuration

2.1 Add New Web OTP client

Configure the new Web OTP client using the explanation as provided below and as shown in Figure 2. Each client is an application that will utilise the Web OTP function. Once the configuration is completed, the client is ready to use Web OTP with the chosen API as explained in Section 4 in this document.

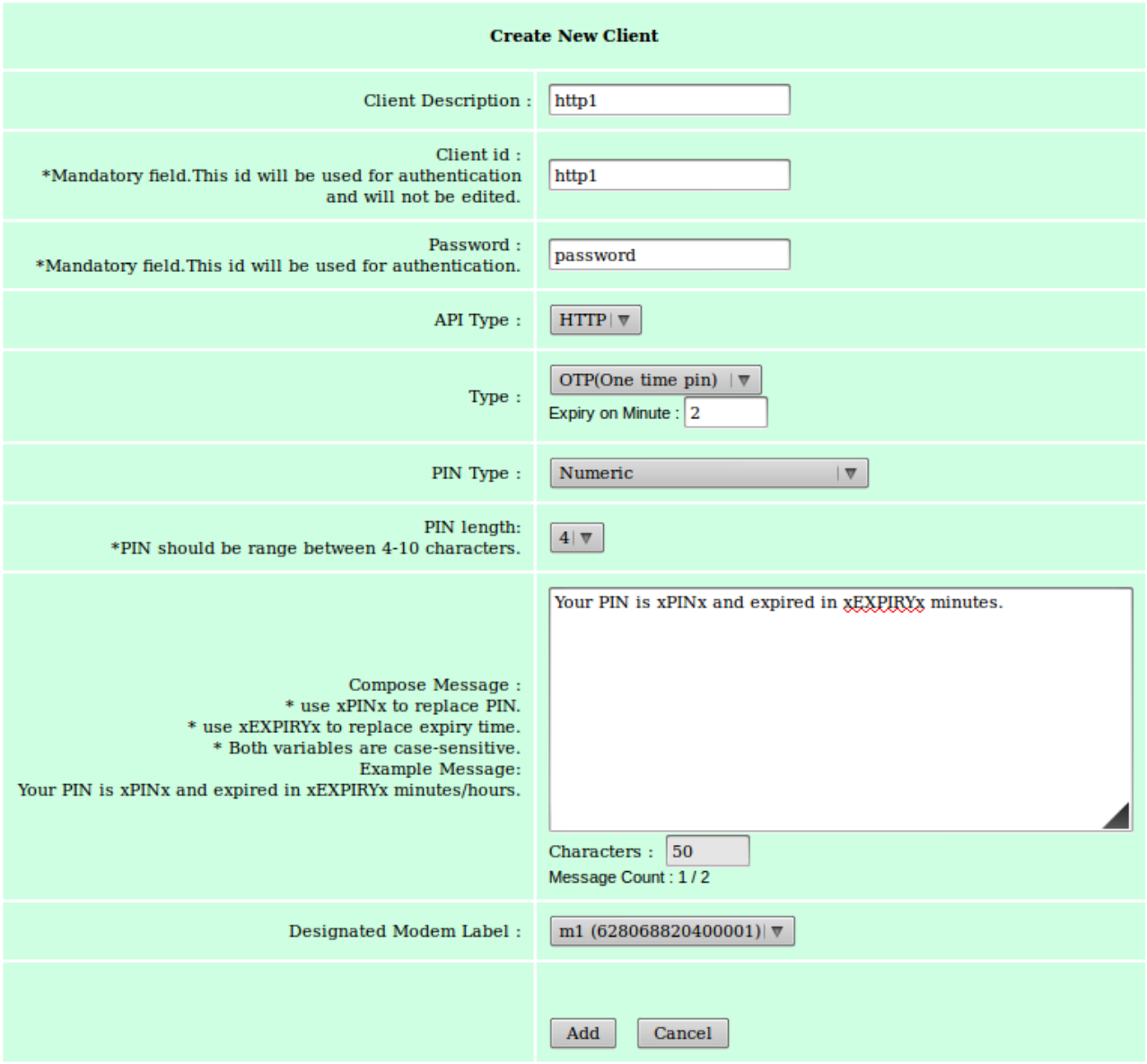


Figure 2: Configure a New Wen OTP Client

Client Description	Short description for Web OTP client
Client ID	Unique ID for Web OTP client.
Password	Password for this client.
API Type	*[HTTP | XML | SOAP]* Web OTP request method. Refer to [Section 13. Web OTP API]
Type	*[OTP | STP]* OTP – One Time password. Use for one time only within minutes. STP – Short Term password. Can be used for max_attempts times within hours.
PIN Type	*[Numeric | Alpha Numeric | Alpha Numeric – Case Sensitive]* OTP characters format.
PIN length	OTP length.
Compose Message	OTP message template.

Designated Modem Label	Select designated modem label to send SMS for this Web OTP client. Need to configure at System Admin page. (Menu → SMS System Setup → Modem Routing)

3. WEB OTP LOG

This feature allow searching for the Web OTP logs based on mobile number, client ID and selected date range as shown in Figure 3 below.

[Export OTP Sent Log]

No	Sent Dtm	Mobile Number	Message Content	Status	Session Validate Dtm	Resp Code
1	30-05-2012 12:20	91234567	Your PIN is 73945 and expired in 5 minutes/hours.	Y	30-05-2012 12:28	201
2	30-05-2012 12:17	91234567	Your PIN is 82497 and expired in 5 minutes/hours.	Y	30-05-2012 12:30	201

Figure 3: Search Logs Information

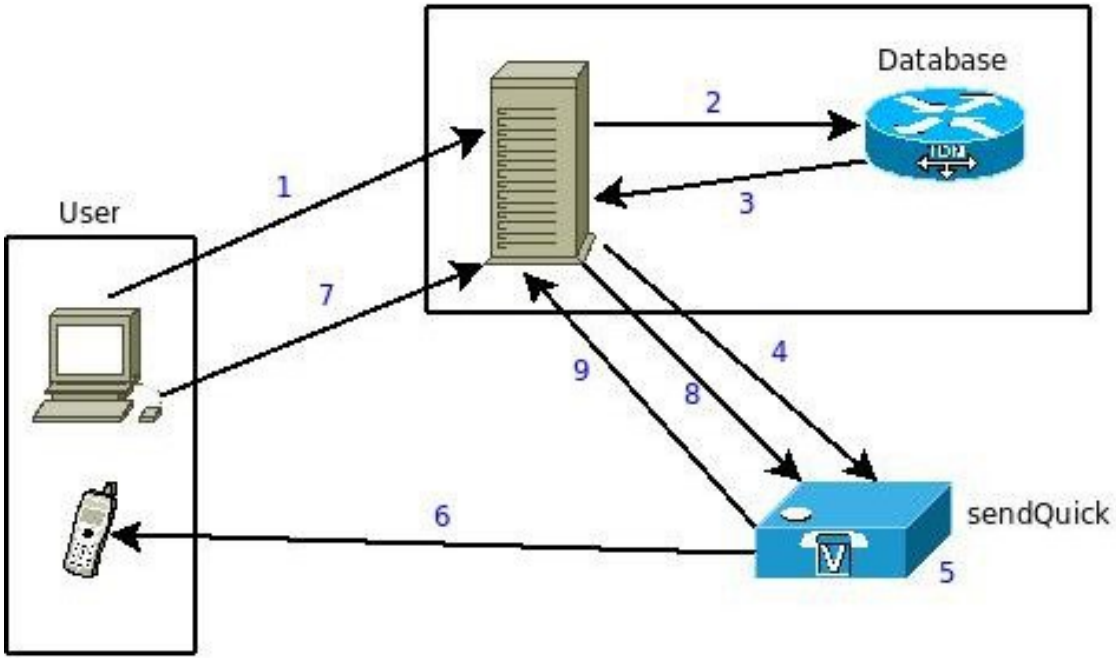
4. WEB OTP API EXPLAINED

The Web OTP is a module designed to perform the OTP generation and OTP verifications for any web based applications. Uisng Web OTP, the developer does not need to be involved in the OTP generation and this function is easily 'offloaded' to the Web OTP module.

As this is a module, the developer can decide when they wish to use the OTP and use the preferred API call to perform the OTP function. There is no restriction on when the API call can be used. Hence, the OTP generation can be used at any stage of any applications like login, transaction confirmation and any other actions that requires 2-FA.

4.1 Web OTP Process/Transaction Flow

The diagram below explains the Web OTP transaction flow. The individual steps are explained and despicted in the flow diagram below.



Process/Transaction Flow is explained below:

1. User Login via Web Portal by using userid + password or PIN
2. Authenticate user by using login information i.e username, password
3. Aunthentication is successful (for username and password)
4. Send user information (username, mobile) to SendQuick via HTTP or XML or SOAP
5. sendQuick generate OTP/STP and store the user information in SendQuick's Database for session validation
6. sendquick send OTP/STP to user's mobile
7. User keyin the “PIN” that he received via SMS
8. Send user information (username, OTP) for sesssion validation
9. SendQuick will response whether the session is valid or not.

4.2 Request for New OTP API

This function is used when the application need to generate an OTP. The API, URL and explanations is as below.

4.2.1 Request using HTTP Method

The HTTP API file name is: **http://[Server's IP]/webotp/otp_http.php**

The following parameters are fixed and predefined by SendQuick.

1. id – this parameter refer to the “client id” which can be created by the client at SendQuick web interface
2. passwd – this parameter ties to “id” parameter – also can be created by the client at SendQuick web interface
3. username – this parameter refers to the username that used by the end user
4. mobile – the mobile number of the end user
5. session_id - the session id is generated by SendQuick and can be re-used for resending the OTP with the same sessionid. For new OTP request, the value is '0'
6. resend – use '1' for the Re-send request, '0' for new OTP request

Request
http://[Server's IP]/webotp/otp_http.php?
id=http&passwd=1234&mobile=+6581234569&username=ym&session_id=0&resend=0

Response
For successful request: <response code>,<sessionId> Example: 205,AbCd123
For unsucessful request: <response code> Example: 101

Please take note that the client has to keep the session id and use it for OTP verification.

4.2.2 Request using XML Method

The XML API filename is: **http://[Server's IP]/webotp/otp_xml.php**

The following parameters are the fixed and predefined by SendQuick.

1. id – this parameter refer to the “client id” which can be created by the client at SendQuick web interface
2. passwd – this parameter ties to “id” parameter – also can be created by the client at SendQuick web interface
3. user – this parameter refers to the username that used by the end user
4. mobile – the mobile number of the end user
5. session_id - the session id is generated by SendQuick and can be re-used for resending the OTP with the same sessionid. For new OTP request, the value is '0'
6. resend – use '1' for the Re-send request, '0' for new OTP request

Request
```
<?xml version="1.0"?>
<post_data>
      <info>
            <id>abc123</id>
            <passwd>password</passwd>
            <mobile>+6581234569</mobile>
            <user>YM</user>
            <session_id>0</session_id>
            <resend>0</resend>
      </info>
</post_data>
```

Response
For successful request: <response code>,<sessionId> Example: 205,AbCd123
For unsucessful request: <response code> Example: 101

Pls take note that the client has to keep the session id and use it for OTP verification.

4.2.3 Request using SOAP Method

The SOAP api filename is: **http://[Server's IP]/webotp/otp_soap.php**

The following parameters are the fixed and predefined by SendQuick.

1. id – this parameter refer to the “client id” which can be created by the client at SendQuick web interface
2. password – this parameter ties to “id” parameter – also can be created by the client at SendQuick web interface
3. username - this parameter refers to the username that used by the end user
4. mobile - the mobile number of the end user
5. session_id - the session id is generated by SendQuick and can be re-used for resending the OTP with the same sessionid. For new OTP request, the value is '0'
6. resend – use '1' for the Re-send request, '0' for new OTP request

The WSDL is as follow:

<definitions targetNamespace="urn:otpwsdl"><types><xsd:schema
targetNamespace="urn:otpwsdl"><xsd:import
namespace="http://schemas.xmlsoap.org/soap/encoding/"/><xsd:import
namespace="http://schemas.xmlsoap.org/wsdl/"/></xsd:schema></types><message
name="generateOTPRequest"><part name="id" type="xsd:string"/><part name="password"
type="xsd:string"/><part name="mobile" type="xsd:string"/><part name="username"
type="xsd:string"/><part name="session_id" type="xsd:string"/><part name="resend"
type="xsd:string"/></message><message name="generateOTPResponse"><part name="return"
type="xsd:string"/></message><portType name="otpwsdlPortType"><operation
name="generateOTP"><documentation>OTP</documentation><input
message="tns:generateOTPRequest"/><output
message="tns:generateOTPResponse"/></operation></portType><binding name="otpwsdlBinding"
type="tns:otpwsdlPortType"><soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/><operation name="generateOTP"><soap:operation
soapAction="urn:otpwsdl#generateOTP" style="rpc"/><input><soap:body use="encoded"
namespace="urn:otpwsdl"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input><output><soap:body use="encoded"
namespace="urn:otpwsdl"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output></operation></binding><service
name="otpwsdl"><port name="otpwsdlPort" binding="tns:otpwsdlBinding"><soap:address
location="http://entera.sendquick.messengr/webotp/otp_soap.php"/></port></service></definitions>

The details of generateOTP is as follow:

Name: generateOTP
Binding: otpwsdlBinding
Endpoint: http://entera.sendquick.messengr/webotp/otp_soap.php
SoapAction: urn:otpwsdl#generateOTP
Style: rpc
Input:
  use: encoded
  namespace: urn:otpwsdl
  encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
  message: generateOTPRequest
  parts:
    id: xsd:string
    password: xsd:string
    mobile: xsd:string
    username: xsd:string
    session_id: xsd:string
    resend: xsd:string
Output:

use: encoded
namespace: urn:otpwsdl
encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
message: generateOTPResponse
parts:
return: xsd:string
Namespace: urn:otpwsdl
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: OTP
Request
POST /webotp/otp_soap.php HTTP/1.0
Host: 127.0.0.1
User-Agent: NuSOAP/0.7.3 (1.114)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: ""
Content-Length: 723

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns5100:generateOTP xmlns:ns5100="http://tempuri.org"><id xsi:type="xsd:string">abc123</id><password xsi:type="xsd:string">password</password><mobile xsi:type="xsd:string">+6581234569</mobile><username xsi:type="xsd:string">ym</username><session_id xsi:type="xsd:int">0</session_id><resend xsi:type="xsd:int">0</resend></ns5100:generateOTP></SOAP-ENV:Body></SOAP-ENV:Envelope>

Response
HTTP/1.1 200 OK
Date: Thu, 04 Oct 2012 02:45:43 GMT
Server: Apache
X-Powered-By: PHP/5.2.6
Set-Cookie: PHPSESSID=48142254ecc95e037e3d6b273fea32fe; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-SOAP-Server: NuSOAP/0.7.3 (1.114)
Content-Length: 459
Connection: close
Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:generateOTPResponse xmlns:ns1="http://tempuri.org"><return xsi:type="xsd:string">205,jSdR</return></ns1:generateOTPResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
For successful request: <response code>,<sessionId> Example: 205,AbCd123
For unsuccessful request: <response code> Example: 101

Please take note that the client has to keep the session id and use it for OTP verification.

4.3 Request to Resend an OTP API

The application can perform a resend of the OTP (for the same session) by using this API. Please check the example value of session_id and resend parameters.

4.3.1 Request to Resend using HTTP Method

Request
http://[Server's IP]/webotp/otp_http.php?
id=http&passwd=1234&mobile=123&username=ym&session_id=Abc123&resend=1

Response
Example for Sucessful request: 205, Abc123
Example for Unsucessful request: 101

4.3.2 Request to Resend using XML Method

Request

```
<?xml version="1.0"?>
<post_data>
    <info>
        <id>abc123</id>
        <passwd>password</passwd>
        <mobile>+6581234569</mobile>
        <user>YM</user>
        <session_id>Abc123</session_id>
        <resend>1</resend>
    </info>
</post_data>
```

Response
Example for Sucessful request: 205, Abc123
Example for Unsucessful request: 101

4.3.3 Request to Resend using SOAP Method

Request

```
POST /webotp/otp_soap.php HTTP/1.0
Host: 127.0.0.1
User-Agent: NuSOAP/0.7.3 (1.114)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: ""
Content-Length: 723

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns5100:generateOTP
xmlns:ns5100="http://tempuri.org"><id xsi:type="xsd:string">abc123</id><password
xsi:type="xsd:string">password</password><mobile
xsi:type="xsd:string">+6581234569</mobile><username xsi:type="xsd:string">ym</username><session_id
xsi:type="xsd:int">Abc123</session_id><resend
xsi:type="xsd:int">1</resend></ns5100:generateOTP></SOAP-ENV:Body></SOAP-ENV:Envelope>
Response
```

HTTP/1.1 200 OK
Date: Thu, 04 Oct 2012 02:45:43 GMT
Server: Apache
X-Powered-By: PHP/5.2.6
Set-Cookie: PHPSESSID=48142254ecc95e037e3d6b273fea32fe; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-SOAP-Server: NuSOAP/0.7.3 (1.114)
Content-Length: 459
Connection: close
Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:generateOTPResponse xmlns:ns1="http://tempuri.org"><return xsi:type="xsd:string">**205,Abc123**</return></ns1:generateOTPResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>

4.4 The OTP Verification API

This function is to verify that the OTP submitted by the user is correct and generated by Web OTP and within the usage validity time period.

4.4.1 The OTP Verification API using HTTP Method

The api file name is: **http://[Server's IP]/webotp/session_http.php**

Following parameters are fixed and predefined by SendQuick:

1. username – the login username of the end user
2. token – the otp/stp that user key-in
3. mobile - the mobile number of the end user
4. session_id - the session id is generated by SendQuick and return together with the response code upon requesting of generating OTP

Request
http://[Server's IP]/webotp/session_http.php?
username=ym&token=035764&session_id=Abc123&mobile=+6581234569

Response
<response_code> Example: 201

4.4.2 The OTP Verification API using XML Method

The api file name is: **http://[Server's IP]/webotp/session_xml.php**

Following parameters are fixed and predefined by SendQuick:

1. username – the login username of the end user
2. token – the otp/stp that user key-in
3. mobile - the mobile number of the end user
4. session_id - the session id is generated by SendQuick and return together with the response code

upon requesting of generating OTP

Request
<?xml version="1.0"?>
<post_data>
<info>
<username>ym</username>
<token>5bc0G4</token>
<session_id>Abc123</session_id>
<mobile>+6581234569</mobile>
</info>
</post_data>

Response
<response_code> Example: 201

4.4.3 The OTP Verification API using SOAP Method

The api file name is: **http://[Server's IP]/webotp/session_soap.php**

Following parameters are fixed and predefined by SendQuick:

1. username – the login username of the end user
2. token – the otp/stp that user key-in
3. mobile - the mobile number of the end user
4. session_id - the session id is generated by SendQuick and return together with the response code upon requesting of generating OTP

The WSDL is as follow:

<definitions targetNamespace="urn:otptokenwsdl"><types><xsd:schema targetNamespace="urn:otptokenwsdl"><xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/><xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/></xsd:schema></types><message name="checkTokenRequest"><part name="username" type="xsd:string"/><part name="token" type="xsd:string"/><part name="session_id" type="xsd:string"/><part name="mobile" type="xsd:string"/></message><message name="checkTokenResponse"><part name="return" type="xsd:string"/></message><portType name="otptokenwsdlPortType"><operation name="checkToken"><documentation>OTP</documentation><input message="tns:checkTokenRequest"/><output message="tns:checkTokenResponse"/></operation></portType><binding name="otptokenwsdlBinding" type="tns:otptokenwsdlPortType"><soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/><operation name="checkToken"><soap:operation soapAction="urn:otptokenwsdl#checkToken" style="rpc"/><input><soap:body use="encoded" namespace="urn:otptokenwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></input><output><soap:body use="encoded" namespace="urn:otptokenwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/></output></operation></binding><service name="otptokenwsdl"><port name="otptokenwsdlPort" binding="tns:otptokenwsdlBinding"><soap:address location="http://entera.sendquick.messengr/webotp/session_soap.php"/></port></service></definitions>

The details of checkTokenRequest is as follow:

Name: checkToken
Binding: otptokenwsdlBinding
Endpoint: http://entera.sendquick.messengr/webotp/session_soap.php

SoapAction: urn:otptokenwsdl#checkToken
Style: rpc
Input:
  use: encoded
  namespace: urn:otptokenwsdl
  encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
  message: checkTokenRequest
  parts:
    username: xsd:string
    token: xsd:string
    session_id: xsd:string
    mobile: xsd:string
Output:
  use: encoded
  namespace: urn:otptokenwsdl
  encodingStyle: http://schemas.xmlsoap.org/soap/encoding/
  message: checkTokenResponse
  parts:
    return: xsd:string
Namespace: urn:otptokenwsdl
Transport: http://schemas.xmlsoap.org/soap/http
Documentation: OTP
Request
POST /webotp/session_soap.php HTTP/1.0
Host: 127.0.0.1
User-Agent: NuSOAP/0.7.3 (1.114)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: ""
Content-Length: 655

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1168:checkToken xmlns:ns1168="http://tempuri.org"><username xsi:type="xsd:string">ym</username><token xsi:type="xsd:string">778034</token><session_id xsi:type="xsd:string">jSdR</session_id><mobile xsi:type="xsd:string">+6581234569</mobile></ns1168:checkToken></SOAP-ENV:Body></SOAP-ENV:Envelope>

Response
HTTP/1.1 200 OK
Date: Thu, 04 Oct 2012 22:41:36 GMT
Server: Apache
X-Powered-By: PHP/5.2.6
Set-Cookie: PHPSESSID=a2b1ff58649be0962eeec421d9c5df2b; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-SOAP-Server: NuSOAP/0.7.3 (1.114)
Content-Length: 452
Connection: close
Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-ENV:Body><ns1:checkTokenResponse xmlns:ns1="http://tempuri.org"><return xsi:type="xsd:string">120</return></ns1:checkTokenResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>

5. Predefined Response Codes

Refer to the list of response codes defined by SendQuick System for the response.

Response Code	Reasons
101	IP is not allowed
102	Authentication Failed
103	Missing Parameter(s)
104	Mobile Number Missing
105	Message Content is Empty
106	Invalid Mobile Number
107	Invalid Message Type
108	Aunthetication Failed. (Password does not match)
110	Invalid Client ID
111	Maximum attempts exceeded
112	OPT Username Empty
113	Generate PIN failed
120	Invalid Username or Token
121	Session Expired
122	Invalid Session ID
130	DB Connection Error
205	OTP successfully generated
201	OTP verification successful