TELUS APIs

blocked URL

TELUS is providing **Alcatel onetouch idolX+** smart phones for you to use with your app. Contact a TELUS representative at the Open Hack event to check one out for the duration of the event.

Available TELUS Services

1. Get Terminal Location Service
2. Send SMS Service
3. Get SMS Delivery Status Service

These TELUS services allow you to create applications using features on the TELUS network. They work only on TELUS phone numbers, where the phone owner has authorized it.

To use these services (and many others that we have available) outside of the TM Forum Open Hack, you can sign-up on the TELUS Partner Portal. We then work closely with our registered partners to help them to responsibly consume our services.

Partners will receive a set of SSL certificates to use with their application and a set of credentials. The services can only be used on phone numbers where the owner has granted the application permission to use these services with their number. TELUS will help Partners to whitelist phone numbers where the owner has agreed to grant the application permission. This stringent authorization process is important to guarantee that the privacy of our Customers is protected.

For the purposes of the TM Forum Open Hack event, we have pre-provisioned some SSL certificates and credentials. Please contact a TELUS representative at the event to get certificates and credentials for your application to use**.** Alternatively, these services have also been exposed as REST services by IBM in Bluemix for you to use, where no certificates or credentials are required.

The Alcatel onetouch idolX+ smart phones that are available for your use have all been whitelisted for use with the below APIs.

Get Terminal Location Service

WSDL: http://webservices.telus.com/parlayx21/072007/parlayx_terminal_location_service_2_3.wsdl

The Terminal Location API allows you to look up the location of a cell phone by its phone number in your application. This location service strictly uses network triangulation to determine location. This means it is generally less accurate but will work in situations where GPS is not available (like in a basement) and is also able to report the last known location if the device cannot be found (which you cannot do with GPS).

REST Request

IBM has exposed our service for you so that for the Open Hack event you do not need to worry about SSL Certs within your application. If you go to the below link, you can see information about the API, examples on how to use it in various languages, and can even perform test operations from within the page.

https://sb-chlauusibmcom-tmf-hack.developer.eu.apiconnect.ibmcloud.com/

SOAP Request

You can use this service by sending a SOAP **request** with the following format:

```
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org
/schema/parlayx/terminal_location/v2_3/local">

        <soapenv:Header/>

        <soapenv:Body>

            <loc:getLocation>

                <loc:address>tel:16042917290</loc:address>

                <loc:requestedAccuracy>5000</loc:requestedAccuracy>

                <loc:acceptableAccuracy>5000</loc:acceptableAccuracy>

            </loc:getLocation>

        </soapenv:Body>

    </soapenv:Envelope>
```

loc:address - The telephone number to find the location of

loc:requestedAccuracy - The accuracy you would like on that location in meters

loc:acceptableAccuracy - Location accuracy must be within this number of meters

Depending on the accuracy that you specify, different technologies will be used. Lower accuracy will take less time to ascertain.

Type of Reading	Best Case	Worst Case
High Accuracy (<1000m)	20 seconds	50 seconds
Low Accuracy (>1000m)	8 seconds	20 seconds

We suggest using an accuracy level of 5000m or greater in order to get fast responses.

SOAP Response

The **response** you receive will look something like this:

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

        <soap:Body>

            <ns2:getLocationResponse xmlns:ns2="http://www.csapi.org/schema/parlayx/terminal_location/v2_3/local"
xmlns:ns3="http://www.csapi.org/schema/parlayx/common/v2_1">

                <ns2:result>

                    <latitude>49.267</latitude>

                    <longitude>-123.0068</longitude>

                    <altitude>22.0</altitude>

                    <accuracy>182</accuracy>

                    <timestamp>2008-07-10T12:54:40.000-04:00</timestamp>

                </ns2:result>

            </ns2:getLocationResponse>

        </soap:Body>

    </soap:Envelope>
```

latitude, longitude: Coordinates where the subscriber is located

altitude: The subscribers elevation

accuracy: how accurate the location result is in meters

timestamp: the time at which the subscribers location was found

Error Messages

Errors that you may see in querying the terminal location:

SVC0001 (generic service exception): There was an error with the service

SVC0002 (An invalid input value): Something in your request input was invalid

SVC0200 (accuracy of location is not within acceptable limit): The location that was found is less accurate than the acceptable accuracy level specified in your request

L101 (Unauthorized Application): Your application is not allowed to use this service

POL0001 (Policy Error Occurred): Policy Error

POL0002 (Privacy Error): The queried number is not whitelisted for your application

POL0006 (Groups Not Allowed): This API can only be used to look up a single subscriber at a time, and will not work on groups

POL0230 (Requested accuracy not supported): The requested accuracy specified in your request is not supported

Send SMS SOAP Service

WSDL: http://webservices.telus.com/parlayx_sms_send_service_2_3.wsdl

The SMS API allows you to send SMS messages through your application. While there are other services (like Twilio or, for TELUS customers, sending an email to <*phone#*>*@msg.telus.com*) that allow you to send SMS messages with fewer restrictions, those services have limitations in terms of the kind of content that can be sent and the frequency of content being sent. With this particular TELUS Send SMS service, there are no restrictions on content or frequency; it also allows for billing via SMS. As such, this service is most useful for premium SMS services.

Endpoint: https://webservices.telus.com/SendSmsService/services/SendSms

REST Request

IBM has exposed our service for you so that for the Open Hack event you do not need to worry about SSL Certs within your application. If you go to the below link, you can see information about the API, examples on how to use it in various languages, and can even perform test operations from within the page.

https://sb-chlauusibmcom-tmf-hack.developer.eu.apiconnect.ibmcloud.com/

SOAP Request

You can use this service by sending a SOAP **request** with the following format:

```
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org
/schema/parlayx/sms/send/v2_3/local">

        <soapenv:Header/>

        <soapenv:Body>

            <loc:sendSms>

                <!--1 or more repetitions:-->

                <loc:addresses>tel:16048183614</loc:addresses>

                <loc:senderName></loc:senderName>

                <loc:message>Message 1 on 12-01-2012</loc:message>

                <!-- Optional -->

                <loc:receiptRequest>
```

```
                <endpoint>http://yourendpoint.com/SmsNotificationService/services/SmsNotification</endpoint>
                <interfaceName>SmsNotification</interfaceName>
                <correlator>1010</correlator>
            </loc:receiptRequest>
        </loc:sendSms>
    </soapenv:Body>
</soapenv:Envelope>
```

loc:address - Telephone number of the subscriber. You can send a message to up to 10 subscribers in a single request.

loc:senderName - Short code or name that you want the subscriber to see as being the sender. Leave this blank for now; normally you would register a name with TELUS to use this.

loc:message - Message body that you want to send

loc:receiptRequest - Optional, request that the service send you a delivery receipt

endpoint - The URL to which to send the delivery receipt

interfaceName - Optional, name of your interface

correlator - ID used to track this SMS message

Response

The **response** you receive will look something like this:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns2:sendSmsResponse xmlns:ns2=http://www.csapi.org/schema/parlayx/sms/send/v2_3/local xmlns:ns3="http://www.csapi.org/schema/parlayx/common/v2_1">
            <ns2:result>267417690</ns2:result>
        </ns2:sendSmsResponse>
    </soap:Body>
</soap:Envelope>
```

result - String reference that can be used to retrieve additional message delivery information from getSmsDeliveryStatus method, detailed below.

Delivery Receipt

And if you asked for a delivery receipt, you will also get the following response when delivery is confirmed:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns2:notifySmsDeliveryReceipt xmlns:ns2="http://www.csapi.org/schema/parlayx/sms/notification/v2_2/local" xmlns:ns3="http://www.csapi.org/schema/parlayx/common/v2_1">
            <ns2:correlator>1010</ns2:correlator>
            <ns2:deliveryStatus>
                <address>tel:16048183614</address>
                <deliveryStatus>DeliveredToTerminal</deliveryStatus>
            </ns2:deliveryStatus>
        </ns2:notifySmsDeliveryReceipt>
    </soap:Body>
</soap:Envelope>
```

correlator - Matches the correlator value specified in your receiptRequest

address - Phone number that the message was sent to

deliveryStatus - Status of the message, indicating whether your target was confirmed to have received the message or not

Get SMS Delivery Status SOAP Service

This service is to be used in conjunction with the Send SMS service above. If you want to get further information out of your response from the Send SMS service, you can use the getSMSDeliveryStatus method.

REST Request

IBM has exposed our service for you so that for the Open Hack event you do not need to worry about SSL Certs within your application. If you go to the below link, you can see information about the API, examples on how to use it in various languages, and can even perform test operations from within the page.

https://sb-chlauusibmcom-tmf-vancouver.developer.us.apiconnect.ibmcloud.com/

SOAP Request

```
< soapenv:Envelope xmlns:soapenv = "http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc = "http://www.csapi.org/schema/parlayx/sms/send/v2_3/local" >
    < soapenv:Header />
    < soapenv:Body >
       < loc:getSmsDeliveryStatus >
          < loc:requestIdentifier >267417690</ loc:requestIdentifier >
       </ loc:getSmsDeliveryStatus >
    </ soapenv:Body >
</ soapenv:Envelope >
```

requestIdentifier - Matches the result value from a previous send SMS response

SOAP Response

Then, the response will look like this:

```
< soap:Envelope xmlns:soap = "http://schemas.xmlsoap.org/soap/envelope/" >
    < soap:Body >
       < ns2:getSmsDeliveryStatusResponse xmlns:ns2 = "http://www.csapi.org/schema/parlayx/sms/send/v2_3/local"
        xmlns:ns3 = "http://www.csapi.org/schema/parlayx/common/v2_1" >
          < ns2:result >
             < address >tel:16048183614</ address >
             < deliveryStatus >DeliveredToTerminal</ deliveryStatus >
          </ ns2:result >
       </ ns2:getSmsDeliveryStatusResponse >
    </ soap:Body >
</ soap:Envelope >
```

This looks a lot like the delivery receipt, except that there is no correlator.

the future is friendly®