



# Number Verification

## Asynchronous SOAP

### Interface information

tyntec's Global Number Verification (GNV) minimizes message loss and unsuccessful call attempts with real-time phone number verification, providing key information on your users' phone numbers, such as subscriber status and roaming data.

Global Number Verification is available with the Asynchronous SOAP API.

SOAP is a protocol used for exchanging messages over a computer network. The messages that are sent do not rely on the underlying transport protocol, which is HTTP post in most cases (and in the specific case of the tyntec infrastructure).

The messages themselves are structured as XML documents and are defined in two parts:

- **SOAP standard**  
The SOAP standard describes elements necessary for communication, such as the "SOAP Envelope" which encloses the actual content to be sent. This content is application-specific and is in this case defined by the WSDL file provided by tyntec. A simple XML parser can be used for extracting the data from the soap response.
- **WSDL file**  
The WSDL file can be accessed on one of the URLs below, depending if HTTP or HTTPS is used:  
<http://78.110.226.74:8080/soap/services/SuperQueryService?wsdl>  
<https://78.110.226.74:8080/soap/services/SuperQueryService?wsdl>

To implement and use Global Number Verification, you should have certain resources:

- The call of a Web Service as defined by the SOAP specification (<http://www.w3.org/TR/soap12-part1/>) with the help of the provided WSDL file.
- A receiving Web Service on the server side in order to accept requests to deliver the results (CheckDeliverRequest, defined in the WSDL file). This is only necessary for the asynchronous interface.

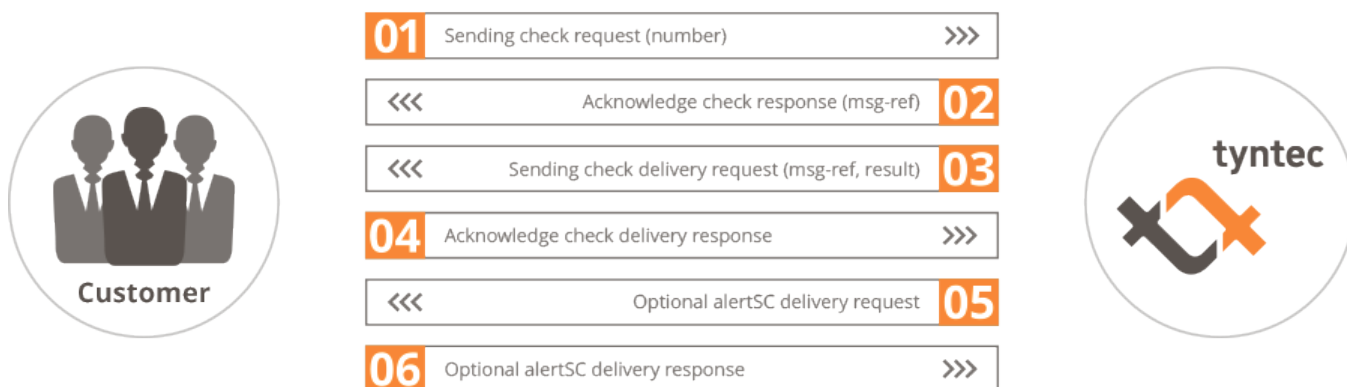
Furthermore, you need a computer system with a fixed IP address from where the SOAP call is initiated, where the Web Service runs and which accepts the results if using the asynchronous API (Check Delivery Requests). This computer system must be reachable from the tyntec IP. For the fastest implementation and expert support from tyntec, we recommend implementing the GNV using:

- Java Axis2 1.3 Web Services framework  
<http://ws.apache.org/axis2/>
- The call of a Web Service as defined by the SOAP specification:  
<http://www.w3.org/TR/soap12-part1/> (with the help of the provided WSDL file)  
<http://java.sun.com/webservices/docs/1.6/tutorial/doc/>
- Tomcat 5.5 or higher for hosting the axis framework:  
<http://tomcat.apache.org/>

The following link to documents that are useful in implementing the Global Number Verification service with the above components:

- [http://ws.apache.org/axis2/1\\_3/userguide.html](http://ws.apache.org/axis2/1_3/userguide.html)
- [http://ws.apache.org/axis2/1\\_3/installationguide.html](http://ws.apache.org/axis2/1_3/installationguide.html)
- <http://www.onjava.com/pub/a/onjava/2001/10/24/xml databind.html>

This is the overview of the workflow for querying phone number information:



In addition, the following information is relevant if you're using the Asynchronous SOAP protocol:

As previously mentioned, tyntec's SOAP implementation uses HTTP as its transport layer. The initial request consists of the number to be queried, in international format. The result will contain:

#### Number range holder network information (MCC/MNC)

- Number range holder country
- Number range holder country
- Number range holder time zone

#### Subscription network information

- IMSI network info (MCC/MNC)
- IMSI country
- IMSI time zone
- HLR network information (MCC/MNC)

- HLR country
- HLR time zone

### Servicing network information

- MSC network information (MCC/MNC)
- MSC country
- MSC time zone

### Other

- Presence information of the handset (true/false)
- Roaming information (true/false)
- Porting information (true/false)
- SS7 network error code

## Connection steps

### Step 1 - Sending check request (number)

The communication protocol is HTTP, whereby a request is submitted to a URL that tyntec specifies. Here's a sample request:

```
<tyn:checkrequest xmlns:tyn="http://www.tyntec.biz/">
<tyn:allnetworkquery>
<tyn:destination>
<tyn:number tyn:ton="Unknown" tyn:npi="Unknown">+491239876543</tyn:number>
</tyn:destination>
</tyn:allnetworkquery>
</tyn:checkrequest>
```

Please note that the XML request must be surrounded by a proper SOAP body in a proper SOAP envelope.

### Step 2 - Acknowledge check response

tyntec will acknowledge the receipt of a Check Request. Please note that this response will be delivered via the HTTP response to the previous HTTP request. If the Check Request has been received successfully, tyntec will acknowledge the Check Request with a MessageRef, which you should store for future reference. Here is the corresponding response:

```
<tyn:CheckResponse xmlns:tyn="http://www.tyntec.biz/">
<tyn:MessageRef>15-21088981464922+491239876543</tyn:MessageRef>
</tyn:CheckResponse>
```

If the Check Request has been received but contains an error, tyntec will acknowledge the Check Request with an error code. Here is one of the possible return codes:

```
<tyn:CheckResponse xmlns:tyn="http://www.tyntec.biz/">
<tyn:ErrorCode>InvalidData</tyn:ErrorCode>
</tyn:CheckResponse>
```

Error Code	Description
HTTP 401 (unauthorized)	This error is returned if the number of queries exceeds the previously agreed to amount of queries, or the username and password are wrong or do not match. This error is not transmitted in a SOAP envelope but as a direct response to the HTTP request.
Invalid Data	This error is returned if the number transmitted is wrong (e.g. there are spaces or letters in the number).
RespondBackURL invalid	This error is returned if <ol style="list-style-type: none"> <li>(1) The Respond-Back-URL given is not a valid one</li> <li>(2) The IP that is used is a local one (if it starts with "192.168", "10" or "127").</li> <li>(3) The host part is "localhost" or empty</li> <li>(4) Any other error occurs when handling the URL.</li> </ol>
Internal Error	This error should be reported to tyntec Support together with the message ID.

### Step 3 - Sending check delivery request

tyntec employs three response codes to determine the outcome of a Global Number Verification query. The table below shows the Response Codes and their associated Check State values:

Check State	Response Code	Description
Success	0 - Success	The requested MSISDN information is returned.
Failure	1 - No Response	Temporarily not connected to the network or the network is not covered by tyntec.
Failure	2 - Error	Network has confirmed that MSISDN does not exist.

A response code is returned only when the Check State value = 'Failure'.

#### Response Code 0 – Success

A successful Check Delivery Request will return all data regarding the MSISDN. The data is delivered in the AllNetworkInfo and will contain information regarding number range holder network, the subscription network (composed of IMSIInformation and HLRInformation) and, if available, the servicing network.

```
<tyn:allnetworkinfo>
<tyn:numberrangeholderinfo></tyn:numberrangeholderinfo>
<tyn:subscriptionnetworkinfo></tyn:subscriptionnetworkinfo>
<tyn:servicingnetworkinfo></tyn:servicingnetworkinfo> // if available
<tyn:presence></tyn:presence>
<tyn:ported></tyn:ported> //optional
<tyn:roaming></tyn:roaming> // optional
<tyn:ss7errorcode></tyn:ss7errorcode>
</tyn:allnetworkinfo>
```

For each network, a NetworkInfo tag will contain the mobile country code (MCC) defining the country, the mobile network code (MNC) defining the network, the tyntecID (a unique ID for each operator) and the OperatorName (the name of the carrier). The NetworkInfo tag will look like this:

```
<tyn:NetworkInfo>
<tyn:MCC />
<tyn:MNC />
<tyn:TynTecID />
<tyn:OperatorName />
<tyn:Country />
</tyn:NetworkInfo>
```

Each Country will contain information about the country name, the isoalpha3code and, if available, the time zone:

```
<tyn:country>
<tyn:countryname></tyn:countryname>
<tyn:timezone> // if available
<tyn:daylightsavingtime> // if country is using DayLightSavingTime
<tyn:hasdst></tyn:hasdst>
<tyn:dstcurrentlyactive></tyn:dstcurrentlyactive> // if country is using DayLightSavingTime
<tyn:dstoffset></tyn:dstoffset> //if country using DayLightSavingTime
</tyn:daylightsavingtime>
<tyn:name></tyn:name>
<tyn:gmtoffset></tyn:gmtoffset>
</tyn:timezone>
</tyn:country>
```

Here's an example, containing all information:

```
<tyn:checkdeliverrequest xmlns=""http://www.tyntec.biz/">
<tyn:messageref>15-21088981464922+491239876543</tyn:messageref>
<tyn:checkstate value=""Success""></tyn:checkstate>
<tyn:allnetworkinfo>
<tyn:numberrangeholderinfo>
<tyn:networkinfo>
<tyn:mcc>262</tyn:mcc>
<tyn:mnc>03</tyn:mnc>
<tyn:tyntecid>17</tyn:tyntecid>
<tyn:operatorname>E-Plus Mobilfunk</tyn:operatorname>
<tyn:country>
<tyn:countryname isoalpha3code=""DEU"">Germany</tyn:countryname>
<tyn:timezone>
<tyn:daylightsavingtime>
<tyn:hasdst>true</tyn:hasdst>
<tyn:dstcurrentlyactive>>false</tyn:dstcurrentlyactive>
<tyn:dstoffset>3600000 ms</tyn:dstoffset>
</tyn:daylightsavingtime>
<tyn:name>Europe/Berlin</tyn:name>
<tyn:gmtoffset>3600000 ms</tyn:gmtoffset>
</tyn:timezone>
</tyn:country>
</tyn:networkinfo>
</tyn:numberrangeholderinfo>
<tyn:subscriptionnetwork>
<tyn:imsiinformation>
<tyn:networkinfo>
<tyn:mcc>262</tyn:mcc>
<tyn:mnc>03</tyn:mnc>
<tyn:tyntecid>17</tyn:tyntecid>
<tyn:operatorname>E-Plus Mobilfunk</tyn:operatorname>
<tyn:country>
<tyn:countryname isoalpha3code=""DEU"">Germany</tyn:countryname>
<tyn:timezone>
<tyn:daylightsavingtime>
<tyn:hasdst>true</tyn:hasdst>
<tyn:dstcurrentlyactive>>false</tyn:dstcurrentlyactive>
<tyn:dstoffset>3600000 ms</tyn:dstoffset>
</tyn:daylightsavingtime>
<tyn:name>Europe/Berlin</tyn:name>
<tyn:gmtoffset>3600000 ms</tyn:gmtoffset>
</tyn:timezone>
</tyn:country>
</tyn:networkinfo>
</tyn:imsiinformation>
<tyn:hlrinformation>
<tyn:networkinfo>
<tyn:mcc>262</tyn:mcc>
```

```

<tyn:mnc>03</tyn:mnc>
<tyn:tyntecid>17</tyn:tyntecid>
<tyn:operatorname>E-Plus Mobilfunk</tyn:operatorname>
<tyn:country>
<tyn:countryname isoalpha3code=""DEU"">Germany</tyn:countryname>
<tyn:timezone>
<tyn:daylightsavingtime>
<tyn:hasdst>true</tyn:hasdst>
<tyn:dstcurrentlyactive>>false</tyn:dstcurrentlyactive>
<tyn:dstoffset>3600000 ms</tyn:dstoffset>
</tyn:daylightsavingtime>
<tyn:name>Europe/Berlin</tyn:name>
<tyn:gmtoffset>3600000 ms</tyn:gmtoffset>
</tyn:timezone>
</tyn:country>
</tyn:networkinfo>
</tyn:hlrinformation>
</tyn:subscriptionnetwork>
<tyn:servicingnetwork>
<tyn:networkinfo>
<tyn:mcc>262</tyn:mcc>
<tyn:mnc>03</tyn:mnc>
<tyn:tyntecid>17</tyn:tyntecid>
<tyn:operatorname>E-Plus Mobilfunk</tyn:operatorname>
<tyn:country>
<tyn:countryname isoalpha3code=""DEU"">Germany</tyn:countryname>
<tyn:timezone>
<tyn:daylightsavingtime>
<tyn:hasdst>true</tyn:hasdst>
<tyn:dstcurrentlyactive>>false</tyn:dstcurrentlyactive>
<tyn:dstoffset>3600000 ms</tyn:dstoffset>
</tyn:daylightsavingtime>
<tyn:name>Europe/Berlin</tyn:name>
<tyn:gmtoffset>3600000 ms</tyn:gmtoffset>
</tyn:timezone>
</tyn:country>
</tyn:networkinfo>
</tyn:servicingnetwork>
<tyn:presence>true</tyn:presence>
<tyn:ported>>false</tyn:ported>
<tyn:roaming>>false</tyn:roaming>
<tyn:ss7errorcode>0</tyn:ss7errorcode>
</tyn:allnetworkinfo>
</tyn:checkdeliverrequest>
Response Code 1 - No Response
Response code '1' indicates that the network did not return a response, e.g.:
<tyn:checkdeliverrequest xmlns:tyn=""http://www.tyntec.biz/"">
<tyn:messageref>1521088981464922+491239876543</tyn:messageref>
<tyn:checkstate tyn:value=""Failure""/></tyn:checkstate>
<tyn:networkerrorcode tyn:networktype=""GSM"" tyn:value=""1"">
<tyn:networkerrorcodedescription hexerrorcode=""0xe040""></tyn:networkerrorcodedescription></tyn:networkerrorcode></
tyn:checkdeliverrequest>
DIALOGUE TIMED OUT

```

There are a number of reasons why a network may not return a response. Often, poor number quality results in the network being unable to respond. Check the quality of the number to ensure that it consists only of the '+' sign and decimal digits - no other characters or spaces are permitted. If number quality is good, there may be a problem with the network and we recommend that you query again at a later time. Another possibility is that the number belongs to an operator that tyntec cannot reach.

## Response Code 2 - Error

Response Code '2' indicates that the network has confirmed that the MSISDN does not exist, e.g.:

```
<tyn:checkdeliverrequest xmlns:tyn="http://www.tyntec.biz/">
<tyn:messageref>15-21088981464922+491239876543</tyn:messageref>
<tyn:checkstate tyn:value="Failure"/></tyn:checkstate>
<tyn:networkerrorcode tyn:networktype="GSM" tyn:value="2">
<tyn:networkerrorcodedescription hexerrorcode="2">
UNKNOWN_ADDRESS
</tyn:networkerrorcodedescription>
</tyn:networkerrorcode>
</tyn:checkdeliverrequest>
```

#### Step 4 - Acknowledge check delivery response

The receipt of the Check Delivery Request must be acknowledged to tyntec. Please note that this response must be delivered via the HTTP response to the previous HTTP request. To acknowledge successful delivery, please send back a response in the following format:

```
<tyn:CheckDeliverResponse xmlns:tyn="http://www.tyntec.biz/">
</tyn:CheckDeliverResponse>
```

If there is an issue in the processing of the response from tyntec, a message indicating an error will be returned. For example, returning the following message will trigger a retry from tyntec:

```
<tyn:checkdeliverresponse xmlns:tyn="http://www.tyntec.biz/">
  <tyn:errorcode>Error while Processing</tyn:errorcode>
</tyn:checkdeliverresponse>
```

#### Step 5 – AlertSC delivery request (optional)

This feature will inform you when the handset associated to the requested MSISDN is switched on again. This feature can be requested from tyntec Support. The notification contains the AlertSCReceiver, the AlertSCDate and the HLRInformation and will look like this:

```
<tyn:alertscdeliverrequest xmlns:tyn="http://www.tyntec.biz/">
<tyn:messageref>15-21088981464922+491239876543</tyn:messageref>
<tyn:alertsinfo>
<tyn:alertscreceiver>+491239876543</tyn:alertscreceiver>
<tyn:alertsdate>2009-02-25 14:28:27.0</tyn:alertsdate>
<tyn:hlrinformation>
<tyn:networkinfo>
<tyn:mcc>262</tyn:mcc>
<tyn:mnc>03</tyn:mnc>
<tyn:tyntecid>17</tyn:tyntecid>
<tyn:operatorname>E-Plus Mobilfunk</tyn:operatorname>
<tyn:country>
<tyn:countryname isoalpha3code="DEU">Germany</tyn:countryname>
<tyn:timezone>
<tyn:name>Europe/Berlin</tyn:name>
<tyn:gmtoffset>3600000 ms</tyn:gmtoffset>
<tyn:daylightsavingtime>
<tyn:hasdst>true</tyn:hasdst>
<tyn:dstcurrentlyactive>>false</tyn:dstcurrentlyactive>
<tyn:dstoffset>3600000 ms</tyn:dstoffset>
</tyn:daylightsavingtime>
```

```
</tyn:timezone>
</tyn:country>
</tyn:networkinfo>
</tyn:hlrinformation>
</tyn:alertscinfo>
</tyn:alertscdeliverrequest>
```

## Step 6 – AlertSC delivery response

The receipt of the AlertSC Delivery Request must be acknowledged to tyntec. Please note that this response must be delivered via the HTTP response to the previous HTTP request. To acknowledge successful delivery, please send back a response in the following format:

```
<tyn:AlertSCDeliverResponse xmlns:tyn="http://www.tyntec.biz/">
</tyn:AlertSCDeliverResponse>
```

If there is an issue in the processing of the response from tyntec, a message indicating an error will be returned. For example, returning the following message will trigger a retry from tyntec:

The receipt of successful results must be acknowledged by your HTTP server by terminating the stream. Failure to terminate the stream will result in a re-delivery from tyntec. Generally, a Web Services framework will automatically close the stream. We recommend that you do not use a hand-written parser for processing SOAP messages, but rather an automated system such as Castor or any other XML parsing technology.

```
<tyn:alertscdeliverresponse xmlns:tyn=""http://www.tyntec.biz/">
  <tyn:errorCode>Error while Processing</tyn:errorCode>
</tyn:alertscdeliverresponse>
```

## Example request

This example shows a Check Request enclosed in a SOAP Envelope with a full HTTP POST Header. Authorization information for „user:testpassword“ is encoded in the Authorization header (found in RFC 2617, Chapter 2).

```
HTTP-Post-Header and SOAP-Envelope:
===== begin =====
POST /soap/services/CheckService HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.2alpha
Host: http2.tyntec.biz:8968
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 583
Authorization: Basic dXNlcjpwZXN0cGFzc3dvcmQ=
<!--?xml version="1.0" encoding="UTF-8"?-->
<soapenv:envelope xmlns:soapenv=""http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd=""http://www.w3.org/2001/XMLSchema"
xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance" xmlns:tyn=""http://www.tyntec.biz/">
  <soapenv:body>
    <tyn:checkrequest>
      <tyn:allnetworkquery>
        <tyn:destination>
          <tyn:number tyn:ton=""Unknown" tyn:npi=""Unknown">+491239876543</tyn:number>
        </tyn:destination>
```



```

</tyn:allnetworkquery>
</tyn:checkrequest>
</soapenv:body>
</soapenv:envelope>
===== end =====

```

## Features

Feature	Activation	Functionality
Flexible respond-back-URL	Default	<p>You can use this to change the address that is used for delivering the results of each request. tyntec uses the given URL only for the accompanying request, and will not replace the general URL transmitted with the account setup.</p> <p>Here is an example: <b>Here is an example:</b></p> <pre> &lt;tyn:checkrequest xmlns:tyn="http://www.tyntec.biz/"&gt; &lt;tyn:allnetworkquery&gt; &lt;tyn:destination&gt; &lt;tyn:number&gt;+491239876543&lt;/tyn:number&gt; &lt;/tyn:destination&gt; &lt;/tyn:allnetworkquery&gt; &lt;tyn:respondbackurl&gt;http://www.yourserver.com/yourservice &lt;/tyn:respondbackurl&gt; &lt;/tyn:checkrequest&gt; </pre>
AlertSC notification	On request	<p>With this feature, you will be notified when a handset that is originally turned off gets turned back on. Here is how it works:</p> <ol style="list-style-type: none"> <li>1. You query a phone number and receive a "phone turned off" response</li> <li>2. tyntec instructs the mobile operator to provide a notification when the phone number becomes available</li> <li>3. When tyntec receives the information that the phone is available, we will notify you.</li> </ol> <p>This functionality depends on the capabilities of each mobile operator's infrastructure and is not guaranteed, so you might not receive AlertSC notifications from some mobile operators.</p>

## FAQs

### Q: Why doesn't the URL that tyntec provided for Global Number Verification work?

A: Check the firewall settings to ensure that requests to URLs with a port number other than 80 are permitted. If in doubt, please connect to <http://support.tyntec.biz:8968> and report to tyntec's customer support the date and time of the attempted connection.

---

### Q: I am able to send the request, but why didn't I receive a response?

A: The IP address from which the request originates must be the same as the IP address you have provided to tyntec, otherwise you won't receive a response from tyntec. If you're using the Flexible respond-back-URL feature, verify that the IP address of the respond-back-URL has been registered with tyntec. If the IP address has not been registered with tyntec, the request may have been intercepted by your firewall as an unrecognized IP address. The IP addresses of all respond-back-URLs must be registered with tyntec's Support Team. tyntec Support must also be notified of all changes to IP addresses.

---

### Q: Why do I receive multiple Check Delivery Requests with the same ID?

A - In the event that your server is unable to process a request, e.g. an error generating the XML document, the Check Delivery Request will be resent. Please check your processes for receiving Check Delivery Requests.

---

### Q: I can send a request and receive back a message ID, but why don't I get any results?

A: The URL to which results are to be directed must be registered at tyntec. Please check that the URL has the same IP address as the originating request (see the second question in this section for additional details). Also, please ensure that your SOAP service that accepts the results is running. The service must be kept running even if you have sent all requests, since there is a time delay between request and receipt of result. If you use a firewall, please ensure that tyntec's IP address (i.e., the IP address you sent the requests to) is accepted.

---

### Q: Why don't I receive results for operator xyz anymore?

A: The operator's name may have changed, while you are still using a transcription table based on the names of operators. Please refer to a transcription table containing MCC +MNC or the TynTecID parameter.

---

### Q: Why do I receive an authorization failure?

A - An authorization failure can be caused by an incorrect username and/or password. Please contact tyntec Support to verify your user details. If you have a test account and have previously been able to send requests successfully, you may have reached your test message limit.

---

**Q: How do I know when my test message limit has been reached?**

A: If you attempt to send messages beyond your test account limit, you will receive an HTTP 401 Error (Unauthorized).

---

**Q: I receive the same Check Delivery Request repeatedly, or results come in very slowly. What can I do?**

A: There are two possibilities for receiving multiple Check Delivery Requests:

- 1) tyntec received a Check Delivery Response with an error message included, or
- 2) There was a problem in the communication process

In the case of an error message in a Check Delivery Response, the Check Delivery Request will be resent, regardless of the content of the error message. If for some reason you no longer wish to receive a certain Check Delivery Request, please return a Check Delivery Response without an error message. An improperly structured Check Delivery Response is the most common cause of a communication problem. The Check Delivery Response may have incorrect syntax that could not be parsed by the XML compiler. Alternatively, a Check Delivery Response may never have been successfully sent. In either case, the Check Delivery Request will be resent, since it cannot be determined whether the Check Delivery Request has been successfully received.

Please make sure that your receiving service returns a proper SOAP Envelope with a proper Check Delivery Response included in the SOAP Body.

---

**Q: I send Check Requests but always get an error back. What's happening?**

A: This can be caused by a number of reasons:

- 1) You have not supplied a username and/or password: Please make sure that you supply both in the HTTP Header in the Basic Authorization Scheme (Refer to RFC 2616)
  - 2) The supplied username and/or password are invalid: Please check if you have mistyped the username and/or password provided to you
  - 3) The supplied number is not a valid phone number: the number must only consist of digits (except a leading +). Any other characters including spaces will not be accepted
  - 4) The Check Request cannot be parsed by our service: Please verify that the Check Request is equivalent to the example request given in this API guide (refer to Connection steps, Step 1). If you have created your program/service with an automatic tool and the WSDL file supplied by tyntec, this should not happen.
-