

ShoreTel Sky Hosted API Developer Guide

Version 1.1 – December 2015

This document contains proprietary and confidential information from ShoreTel. Please do not distribute to any third party without permission.

About this document

This document is drafted for the application integrator. It specifies the details of how to issue commands to the ShoreTel Sky Hosted API service.

When building your application, please ensure that you can easily accommodate changes, should they occur.

The service may present connection or operational problems or down-time. The system may be brought down for maintenance at any time without advance notice. Please design your applications so that they can gracefully deal with service availability issues.

Please direct any questions to the engineer or product specialist you received this document from. If you have not been in touch with anyone specifically, reach out to skysupport@shoretel.com

Contents

About this document	1
Overview	3
Servers	3
API Usage and Limitations	3
API Variables	4
<DID>.....	4
<PIN>.....	4
<EX_TN>.....	4
<Endpoint>.....	4
<Call ID>	4
General format of API request.....	5
Example of GET request.....	5
Example of XML request	5
Example of reply	6
Call object.....	7
Available API Commands	8
Dial	8
Answer	9
Ignore	10
Release	11
Hold	12
Resume	13
List Calls.....	14
Subscribe	15
Errors.....	16
Common error messages	16
Known Issues.....	18

Overview

There are two methods in which to connect to and utilize the ShoreTel Sky Hosted API. The first is by issuing a standard GET request via HTTPS. Parameters in the request dictate the action to be taken. The second is by opening a long lived HTTP/1.1 connection via HTTPS and communicating back and forth via [Chunked Transfer Encoding](#). Both of these methods will be explored in-depth.

In response to all requests, the server will return a small XML fragment describing the result of the action.

Server

The server that ShoreTel Sky users will connect to <https://hostedconnect.m5net.com>

The resource being requested is /bobl/bobl. When issuing GET requests, the URL will be <https://hostedconnect.m5net.com/bobl/bobl> with query parameters to control behavior. When utilizing the long lived connect, a POST request is issued to /bobl/bobl and the actions are sent as XML fragments in the connection stream. Please refer to the examples documented in this document and this will become clear.

API Usage and Limitations

This API is provided for all Managed Profiles and may be used free of charge. Provisioning is not required before use.

With that said, there are some limitations to be aware of:

- This API interacts directly with a physical phone. Without a phone connected, API calls will fail. Therefore, this API cannot be used for certain situations such as creating an [auto-dialer](#).
- This API does not provide DTMF support. You can place calls with the API but you cannot send DTMF tones once on a call (such as interacting with auto attendants or dialing an extension).
- Certain API calls may be subject to rate limiting. These are outlined in this document, but are subject to change without notice. If your calls are being rate limited, [you will be informed in the error message](#).
- Users can be temporarily or permanently blocked from using the API service to prevent abuse or fraud. If your user is blocked, [you will be informed in the error message](#). If this happens, reach out to skysupport@shoretel.com to learn why you are blocked and how to remedy it.
- You may not receive full control of your phone when using the API. An example is that you may not be able to dial out to international calls using the API in an effort to prevent call fraud. Restrictions such as this are subject to change without notice.
- While a best effort attempt is made to provide 24/7 availability of this API, it is subject to be taken down at any time for maintenance. You should not make this API vital to your business operations.

API Variables

This document will refer to several variables. They are explained here.

<DID>

The DID is the telephone number of the ShoreTel user utilizing the API. The full ten digit number should be used (i.e. 6465551234). For the purposes of this API, the DID will be used as the username.

<PIN>

The PIN refers to the password that is used when logging into the DID or accessing the voicemail. For the purposes of this API, this PIN is used as the password.

<EX_TN>

When performing an action such as dialing, EX_TN refers to the external number being dialed.

This number should follow some formatting rules:

- **No outbound dial digit is needed.** The API determines if the number is external and will place the call appropriately on the phone.
 - 6465551234 – valid
 - 16465551234 – valid
 - 916465551234 - invalid
- If this number is an extension on the same system, it can simply be the 4 digit extension (i.e. 1234). If the full DID is provided in this instance, it will still be placed as an internal extension dial.
- International numbers should be prefaced with the international access code (011) and then the appropriate country code. You can refer to howtocallabroad.com to see how to format an international number. Please refer to the [API limitations regarding international calls](#). They may be blocked at any time to combat fraud. If this is the case, the user will be required to dial the number on their physical phone. The call may be manipulated with other API calls once placed.
 - 01161212345678 – valid
 - 101161212345678 – invalid
 - 61212345678 – invalid

<Endpoint>

This is the MAC address of the phone. Unless there is some special provisioning in place that requires a certain device to be specified, this should be left blank.

<Call ID>

This refers to the unique ID of a call. In most cases, this can be left blank. When left blank, the first call found that matches the appropriate state will be used. For example, an ANSWER command with no call ID specified will answer the first (and often only) call that is in a ringing state. If your company's work flow often has several calls active on a phone at once, this Call ID field can be used to ensure the action is performed on the appropriate line.

The Call ID can be determined by either the [Subscribe](#) or [List Calls](#) command.

General format of API request

Each API request has several fields:

- name – This is the command being utilized. It begins with org.m5.apps.v1.cti.ClickToDial.
- user – See the [<DID>](#) variable above. This is the user running the command.
- password – See the [<PIN>](#) variable above. This is for authentication of the user running the command.
- args – This is an argument for the command being used. Commands often require multiple arguments in which case several can be provided. The order is important and must be provided as specified. Arguments that may be blank must still have a corresponding args element.

Example of GET request

Below is an example of one of [Dial](#) request being made with a simple GET request. This can be tested by simply loading the URL with a browser. An XML response will be returned. Simple click to dial applications often make this request and disregard the output.

<https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.dial&user=6465551000&password=12345&args=6465551234&args=>

The above command uses the org.m5.apps.v1.cti.ClickToDial.dial command, which places an outbound call from a phone. The user is 6465551000 and is authenticated with their PIN of 12345. The first argument specifies that this user is dialing out to number 6465551234. The second argument is the endpoint specifier and is left blank.

Example of XML request

Requests can also be sent in as an XML request over a long lived HTTP/1.1 connection using [Chunked Transfer Encoding](#). Please refer to the Wikipedia article or examine the source code of the provided SDKs to see how to establish this connection.

The below XML request is equivalent to the dial request sent in the GET example.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.dial</Name>
  <User>6465551000</User>
  <Password>12345</Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">6465551234</Arguments>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></Arguments>
  <Id>1</Id>
</Command>
```

One additional aspect to using XML requests is the Id field. This field is used to pass in an arbitrary numerical identifier that is returned in the response. Because multiple requests can utilize the same connection, this Id is used to correlate responses back to their respective requests. This field can be omitted in which case a random Id will be returned.

Note on Chunked Transfer Encoding

One thing to note on Chunked Transfer Encoding which is not often explained is that a 0 length chunk marks the end of all requests. Your application should not send a 0 length chunk so that the API server

keeps the connection alive and can accept further requests on the same connection. If your application follows a request with a 0 length chunk and then sends another request, the server will ignore the second request and may invalidate the connection as a whole.

If you're not implementing the connection at the socket level, you may need to experiment to determine how best to utilize Chunked Transfer Encoding.

For example, if you're utilizing a C# `HttpWebRequest`, you will need to hold the `RequestStream` open for the lifetime of the connection because a 0 length chunk will be sent when it is closed.

```
HttpWebRequest request =
HttpWebRequest.Create("https://hostedconnect.m5net.com/bob1/bob1") as HttpWebRequest;
    request.Method = "POST";
    request.ContentType = "text/xml";
    request.SendChunked = true;
    var requestStream = request.GetRequestStream(); // Hold this request stream
open for the lifetime of the connection
    string subscribe = "<?xml version='1.0' encoding='UTF-8'
standalone='yes'?>... command continues here";
    var buffer = Encoding.UTF8.GetBytes(subscribe);
    requestStream.Write(buffer, 0, buffer.Length);

    var reply = request.GetResponse();
    StreamReader text = new StreamReader(reply.GetResponseStream());

    while (!text.EndOfStream)
        Console.WriteLine(text.ReadLine());
```

Example of reply

The reply to a request is the same XML fragment regardless of how the request was sent in.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:boolean" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</Result>
</org.m5.api.v1.Response>
```

The reply will come back as an `org.m5.api.v1.Response` XML object. Each response will have the following three fields:

- **ErrorCount** – This refers to the number of problems encountered with the API. If one or more errors did occur, this will be followed by an `Errors` node that has attached error objects. Please refer to the `Errors` section of this document for more information. It should be noted that these refers only refer to the API itself. A call that was not able to connect would not be considered an error as it is an issue with the phone and not the API.
- **Id** – This is the same `Id` passed in with the request. If no `Id` was provided, this will be a random value.
- **Result** – This is the result of the action and will differ in type depending on the command. In the example above, it is a Boolean with a value of `true` indicating that the call was placed properly.

Call object

When developing an application that provides functionality beyond simple click to dial capabilities, it will likely need to deal with call states.

The API provides this information with an XML node of type `org.m5.data.v1.cti.Call`. This node has the following format:

```
<item xsi:type="org.m5.data.v1.cti.Call">
  <Ani>16465551234</Ani>
  <CLID>6465551000</CLID>
  <DID>6465551000</DID>
  <Duration>46</Duration>
  <Endpoint>0010492e1234</Endpoint>
  <Id>65885</Id>
  <Originated>true</Originated>
  <State>CONNECTED</State>
</item>
```

The fields provide the following information:

- **Ani** – The telephone number of the other party. The formatting of this number can vary so you may need to sanitize it if you will be performing lookups using it.
- **CLID** – This is the number that was initially dialed. Useful for when a call comes in on a main line and winds up at a user after navigating an auto attendant. **This field may not always be present.** Your application should gracefully deal with its absence.
- **DID** – The user this call state is for.
- **Duration** – The duration of this call in seconds. This duration includes the full lifetime of the call including ringing, not just the time it's been active.
- **Endpoint** – The endpoint that this state is for. Except for special circumstances, this field can be ignored.
- **Id** – The unique ID of this call. This is the Id used when manipulating the call with other commands. The ID argument in those commands can be ignored if the user only has one call but specifying it will help in instances where there are multiple calls active.
- **Originated** – This specifies which party initiated the call. If true, the call is outbound. If false, the call was an incoming call.
- **State** – This is the current state of the call. This state can be one of the following:
 - **RINGING** – The call is ringing *into* phone
 - **PENDING** – The call is ringing *out* from the phone
 - **CONNECTED** – The call is connected and active
 - **HOLDING** – The call is connected but has been placed on hold
 - **DELETED** – The call has ended and is no longer relevant to this phone (been hang up, ignored, transferred away, etc)

Available API Commands

Below are the available API commands your application may make.

Dial

The dial request will initiate an outbound call from the phone to an external phone number or extension.

Arguments:

1. The number to dial
2. The endpoint to dial from (may be left blank)

Returns:

- Boolean representing success. True means the call was placed, not necessarily that the call went through.

GET Request

https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.dial&user=<DID>&password=<PIN>&args=<EX_TN>&args=<Endpoint>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.dial</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><EX_TN></Arguments>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Endpoint></Arguments>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:boolean" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</Result>
</org.m5.api.v1.Response>
```


Answer

This command will answer an inbound ringing call on the user's phone. If the handset is on-hook, the call will be answered on speaker phone.

Arguments:

1. The Call ID to answer (may be left blank to answer any ringing call)
2. The endpoint to answer on (may be left blank)

Returns:

- Boolean representing success.

GET Request

<https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.answer&user=<DID>&password=<PIN>&args=<Call ID>&args=<Endpoint>>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.answer</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Call ID></Arguments>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Endpoint></Arguments>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:boolean" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</Result>
</org.m5.api.v1.Response>
```

Ignore

Ignoring a call will stop it from ringing on your phone. The call will continue to ring on the callers end before going to voicemail. There is no difference to the caller between ignoring the call and simply not picking it up. In other words, the caller won't know that the user ignored the call.

Arguments:

1. The Call ID to ignore (may be left blank to ignore any ringing call)

Returns:

- Boolean representing success.

GET Request

<https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.ignore&user=<DID>&password=<PIN>&args=<Call ID>>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.ignore</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Call ID></Arguments>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:boolean" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</Result>
</org.m5.api.v1.Response>
```

Release

Releasing a call will hang up an active call.

Arguments:

1. The Call ID to hang up (may be left blank to release any active call)

Returns:

- Boolean representing success.

GET Request

<https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.release&user=<DID>&password=<PIN>&args=<Call ID>>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.release</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Call ID></Arguments>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:boolean" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</Result>
</org.m5.api.v1.Response>
```

Hold

Places an active call on hold.

Arguments:

1. The Call ID to place on hold (may be left blank to hold any active call)

Returns:

- Boolean representing success.

GET Request

<https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.hold&user=<DID>&password=<PIN>&args=<Call ID>>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.hold</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Call ID></Arguments>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:boolean" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</Result>
</org.m5.api.v1.Response>
```

Resume

Resumes a call that was previously placed on hold.

Arguments:

1. The Call ID to resume (may be left blank to resume any held call)

Returns:

- Boolean representing success.

GET Request

<https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.resume&user=<DID>&password=<PIN>&args=<Call ID>>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.resume</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Call ID></Arguments>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:boolean" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">true</Result>
</org.m5.api.v1.Response>
```

List Calls

The list calls method will check the current state of the phone and return a list of calls currently active on it.

This function is intended for infrequent use. If your application requires constant updates (for screen pops or a CTI), please refer to the Subscribe command. The use of List Calls is subject to rate limiting and will fail if your application makes this request too often.

Arguments:

1. The endpoint to check (may be left blank)

Returns:

- xmlList with a list of call objects (see the Call object reference above). If the phone is idle, the xmlList will have no child nodes.

GET Request

<https://hostedconnect.m5net.com/bobl/bobl?name=org.m5.apps.v1.cti.ClickToDial.listCalls&user=<DID>&password=<PIN>&args=<Endpoint>>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.listCalls</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Arguments xsi:type="ns3:string" xmlns:ns3="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Endpoint></Arguments>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="xmlList" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <item xsi:type="org.m5.data.v1.cti.Call">
      <Ani>16465551234</Ani>
      <DID>6465551000</DID>
      <Duration>46</Duration>
      <Endpoint>0010492e1234</Endpoint>
      <Id>65885</Id>
      <Originated>true</Originated>
      <State>CONNECTED</State>
    </item>
  </Result>
</org.m5.api.v1.Response>
```

Subscribe

The subscribe command instructs the API to push out call events asynchronously as they happen.

The subscribe command only sends out new call events, it does not relay the current state of the phone. For this reason, it is helpful to issue a List Calls command before or after the Subscribe command so that the current call states are picked up as well. This can be done on the same connection.

Arguments: This command takes no arguments.

Returns:

- If the subscription is successful, the API will return a result of '0'. Otherwise it will return an error message.

GET Request

This command cannot be used with a GET request.

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Command>
  <Name>org.m5.apps.v1.cti.ClickToDial.subscribe</Name>
  <User><DID></User>
  <Password><PIN></Password>
  <Id>1</Id>
</Command>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="ns5:int" xmlns:ns5="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">0</Result>
</org.m5.api.v1.Response>
```

This response is then followed up with [call state](#) updates as they occur. They will look like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>0</ErrorCount>
  <Id>1</Id>
  <Result xsi:type="xmlList" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <item xsi:type="org.m5.data.v1.cti.Call">
      <Ani>6465551234</Ani>
      <CLID>6465551000</CLID>
      <DID>6465551000</DID>
      <Duration>0</Duration>
      <Endpoint>0010492e1234</Endpoint>
      <Id>66517</Id>
      <Originated>false</Originated>
      <State>RINGING</State>
    </item>
  </Result>
</org.m5.api.v1.Response>
```

Errors

While working with the API, a number of error conditions may arise. Your application should gracefully handle errors and log them or alert the user when applicable.

Every response from the API will include an `ErrorCount` node that should be checked. Below is an example of a response that includes an error.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<org.m5.api.v1.Response xmlns:m5="http://www.m5net.com/org/m5/data/v2/cti"
xmlns:csta="http://www.ecma-international.org/standards/ecma-323/csta/ed5">
  <ErrorCount>1</ErrorCount>
  <Errors>
    <item xsi:type="org.m5.api.v1.Error" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <MajorErrorCode>5</MajorErrorCode>
      <Message>Authorization or authentication failed</Message>
      <MinorErrorCode>0</MinorErrorCode>
    </item>
  </Errors>
  <Id>13</Id>
</org.m5.api.v1.Response>
```

Common error messages

In most cases, the error message will make it obvious what the problem is. Here are some common error codes to check for. If you are receiving an error not listed in this section, reach out to skysupport@shoretel.com to receive assistance.

Error code 2, 3, or 4

- Command is missing a class name.
- Command is missing a method name.
- No such method

These error codes occur when an incomplete, non-existent, or mistyped command was sent in. Check that the name portion of the command matches one available in the Available API Commands section.

Error code 5

- Authorization or authentication failed
- Your user has been temporarily blocked for API abuse.

This error code occurs when the command was blocked from running because an invalid DID and PIN was provided or this DID has been blocked for abuse. The API abuse message refers to a block put into place by ShoreTel and is not to be confused with rate limiting. If this happens, contact skysupport@shoretel.com to determine why.

Error code 6

- This application has been temporarily blocked for API misuse.

This error occurs when rate limiting has prevented the command from running. If your application receives this error, it should slow down the frequency of the command it was using. Currently this only affects the List Calls command.

Error code 8

- Endpoint not found. Log into your phone.

This occurs when the API cannot find the phone logged into the DID specified. Verify that your phone is connected and operational. If it is, try unassigning/logging out of the phone and then logging back into it. If the error persists, contact skysupport@shoretel.com for assistance.

Error code 13

- The command expected arguments of different type or amount

This occurs when a command was sent in with an invalid amount or type of arguments. Refer to the [Available API Commands](#) section to see the proper way to format the request. Note that any optional arguments must still be present even if they are left blank.

Error code 17

- A temporary problem forced the unsubscription from updates. Try subscribing again. Notify system@m5net.com if this condition persists

This error may occasionally occur when subscribed to events. The application should simply send another subscribe event and not alert the user. If this error occurs immediately following a subscription request, contact skysupport@shoretel.com.

Error code 19

- Failed to unmarshall XML

The API server was unable to understand the XML request. Refer to [Available API Commands](#) section and verify your request matches. This Message field of this error will attempt to explain why the XML could not be processed.

Known Issues

- If there are [Call Screening or Find Me](#) rules set up on profile, call events from the Subscribe command will be unreliable. For instance, an incoming call may receive the PENDING, CONNECTED, and DELETED states all at once as the call manager receives the call and then transfers it out.
- The API may periodically unsubscribe a user from receiving call events. An error message will be sent stating “A temporary problem forced the unsubscription from updates. Try again.” In this case, the application should simply send another subscription request and not alert the user.