



Open Communications Platform (OCP) REST API Guide and Reference

Version 1.1

950 Tower Lane
Foster City, CA 94404
T: 650-525-9200
F: 650.287.2628
www.IntelePeer.com

Copyright

© 2008 by IntelPeer™

All rights reserved. Published in 2008

Printed in the United States of America

Table of Contents

Using the OCP Platform	6
What is the Open Communications Platform (OCP)?	6
Managing Users	7
Making Calls with the OCP	9
OCP Methods	11
session GET	17
user GET	17
user POST	19
user DELETE	21
user PUT	22
phonenumber GET	23
phonenumber POST	25
phonenumber PUT	26
phonenumber DELETE	27
huntorder PUT	29
huntorder POST	34
huntorder GET	36
huntorder DELETE	37
calltimeexception POST	38
calltimeexception DELETE	41
calltimeexception GET	42
dnd POST	45

dnd GET	46
dnd DELETE	47
blacklist POST	49
blacklist GET	52
blacklist DELETE	52
email POST	53
email GET	54
email DELETE	55
email PUT	56
carriers GET	57
currency GET	59
currency POST	60
currency PUT	61
currency DELETE	62
mouallowed GET	63
mouallowed POST	64
moused GET	66
moused DELETE	67
group POST	68
group GET	69
group PUT	71
group DELETE	72
call POST	73
call GET	75

autoconference POST	76
autoconference GET	78
autoconference PUT	80
autoconference DELETE	82
conference POST	83
conference PUT.....	85
blastvoice POST	87
blastvoice PUT.....	89
blastvoice GET.....	91
blastvoice DELETE.....	94
ttspreview POST	95

Using the OCP Platform

The IntelPeer Open Communications Platform (OCP) is an application programming interface (API) that allows programmers to embed voice telephony in Web and business applications. This document introduces you to the programming environment, its concepts, and terminology. The documentation also provides a comprehensive reference for each resource and method available in the API.

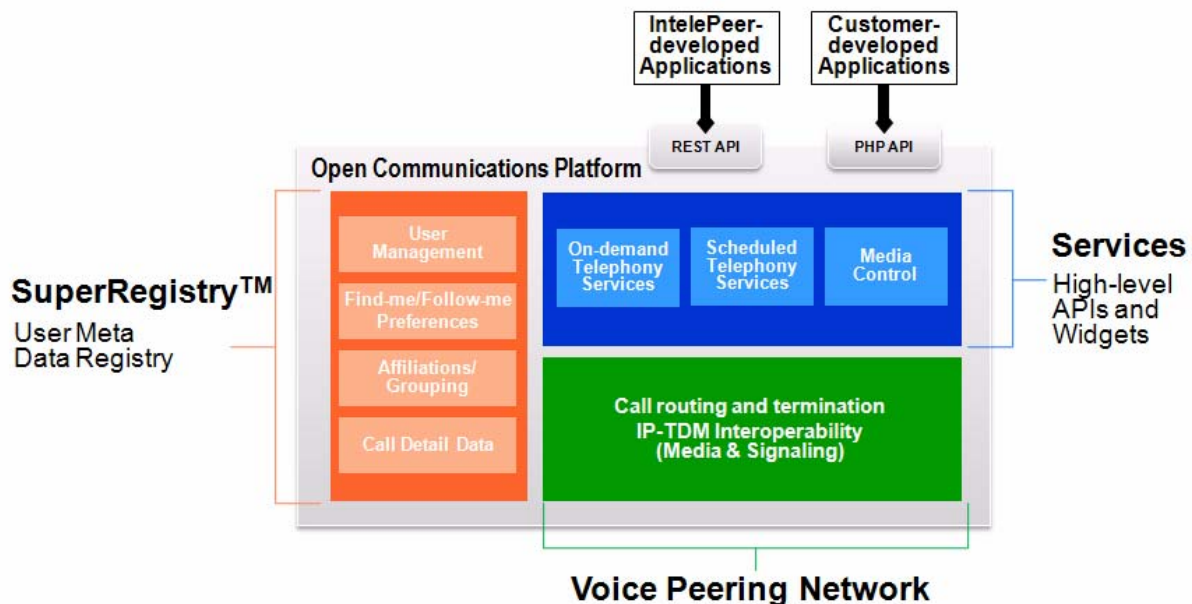
About this Documentation

You should read this document if you are interested in programming with REST. If you would like to use the PHP interface, contact your IntelPeer representative for a copy of that documentation.

Throughout this document, the names of application resources, their methods, variables, and responses appear in *courier* font.

What is the Open Communications Platform (OCP)?

The IntelPeer OCP API is a suite of telephony services encompassing auto conferencing, click-to-call, blast (group voice reminders), and conference-on-demand. These services reside on top of and make use of IntelPeer's carrier-grade infrastructure. The following diagram depicts the platform:



The API also includes user management and accounting methods. With these methods your application can easily do any of the following:

- create, edit, and delete users

- set and adjust user find-me /follow-me preferences
- ensure private/anonymous calling
- manage phone numbers
- set and edit minutes-of-use (MOU) data
- set and edit incurred charges on a user

Most OCP services require that a user exist in the IntelPeer database, but not all of them. However, use of the OCP itself requires your organization to have a valid Enterprise Identifier (EID). IntelPeer assigns this EID after a partnership agreement has been reached.

Note: To obtain the URL you use to make REST API calls, speak with your IntelPeer representative.

How you secure your application depends on your environment. If you are working on a private network, you can use private network connectivity to secure your application. If your application resides on a public network you can use https.

Managing Users

This section is an overview of the methods you will use to work with users. The OCP user management features allow you to create a user, manage the user's find-me/follow-me preferences, and set up accounting for minutes-of-use and currency.

Creating a User

The `user` resource represents all the standard attributes associated with a human being in your application. Creating a user requires you to specify the following information:

- first name
- last name
- email address
- default phone number

The email address and the phone number you specify must be unique within the IntelPeer system.

When you call `user POST` and pass in an instance of this class, the system creates a user, gives it an ID (`userId`), and adds it to the IntelPeer database. Your user is now an IntelPeer system user. Most methods in the API require you to supply a user's `userId`.

A `phonenumber` can represent any number belonging to any carrier any where in the world. Currently, however, the OCP telephony services can only make calls to the United States and Canada. A user can have multiple phone numbers where they might wish to be contacted such as home, office, mobile phone, and so forth. Each time you add a `phonenumber` to the system you specify the phone number and the type. If the phone number belongs to a cell phone, you must supply a carrier ID as well.

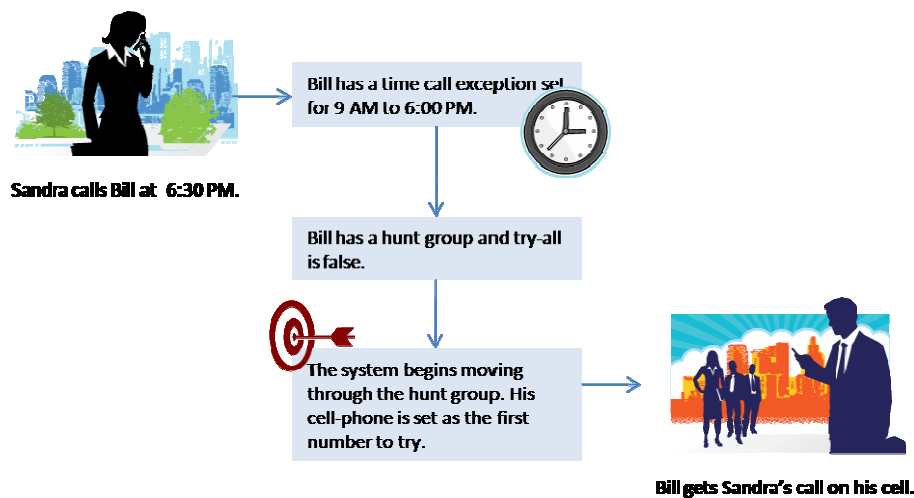
Calling Preferences

Users will have preferences regarding when they will accept calls on a particular phone. You use call-time exceptions (`calltimeexception`) to describe specific time periods. For example, your application can allow users to specify a block of time such as Monday through Friday 9 AM to 5 PM when they prefer to receive calls on their office phone. You use `calltimeexception POST` to associate these time periods with a particular number belonging to a user.

You can also define a hunt group for a user. A hunt group orders the user's phone numbers from first to last. When a call is made to the user, the system begins at the first number and calls each number in turn until either a human being answers or it reaches the last number.

Hunt groups also include a try-all attribute that instructs the system to try-all the user's numbers in the group when connecting a call. If this option is set, the system rings all the user's phones *simultaneously*. The first number to be answered by a human voice, receives the call.

A system considers a user's calling preferences when attempting to reach a user with a call. The following diagram depicts a typical scenario and how the system progresses as it interprets user preferences:



When calling a user, the system always first uses a user's call-time exception preferences. Then, the system attempts to contact a user through a hunt group. If the user has no hunt group set, the system uses the default number on the user. If a hunt group exists, the system considers the try-all attribute, calling all numbers simultaneously if this option is true. If try-all is false, the system attempts to contact the user by sequentially trying each number in the hunt group.

A user can also set a DND preference. This allows a user to set blocks of time in which all calls are blocked.

Contact Groups and Black Lists

A particular user may have a set of numbers that he or she calls frequently. For example, Bill calls Sandra his coworker, Alice his wife, his son Todd, and his running buddy Zhao. Your application can allow Bill to

create one or more set of contact groups. For example, Bill might set up a group for his family members. Then, when Bill is going to be late, he can send a blast call to his family group informing them.

Occasionally, a user might want to block one or more numbers. This is special type of group associated with a user called a black list (`blacklist`). Numbers on a black list cannot use the IntelPeer system to dial the user. Your application can allow a user to add and remove numbers from a black list.

User Accounting: Minutes and Charges

Your application establishes each user with a minutes-of-use (MOU) value using the `mouallowed SET` method. The OCP keeps track of user calls and increments the user's MOU value when a call is made. You can use the `mouused GET` method to retrieve a user's current MOU. If you wish to set MOU limits, your application is responsible for ensuring a user does not exceed their MOU limit. For example, before placing a call your application can use `mouused GET` and compare it to `mouallowed GET` to check if a user has the MOU available to make a call.

Similar to MOU, you can attach a currency value (charge) to a user using the `currency POST` method. The OCP has the `currency GET` method, `currency PUT` method, and the `currency DELETE` method. Your application is responsible for using these methods to edit a user's currency in accordance with the schema your company defines. The denomination of currency is up to your application.

Making Calls with the OCP

The OCP allows you to write applications that make several types of phone calls. Before your application can make use of the OCP, it must use your company's enterprise ID and key to start a session with the `session GET` method. Once a session is established the system can manage users and make calls. The system supports the following call types:

<code>call</code>	The system connects one point to another immediately.
<code>blastvoice</code>	The system sends an announcement from one point to one or more recipients.
<code>autoconference</code>	A user elects to participate in a conference and the system joins that user to the conference when it occurs.
<code>conference</code>	A user, acting as a moderator, schedules a conference and invites one or more participants. The system joins the moderator and participants into a "conference room" on the IntelPeer system.

When your application creates a call, it supplies an origination and destination for both "legs" of the call. When connecting two calls, the IntelPeer software always connects the destination first and then connects back to the call originator.

In all cases except for the **blastvoice** call, the system considers it a success when a call reaches a human voice. If the system reaches a human voice, the system requests that the recipient press 1 to accept the incoming call. When it receives a positive response, the system acts appropriately according to the type of call:

call	Connects back to the originator..
autoconference	Connects the recipient to the conference. The recipient need not know the conference connection information.
conference	Connects the recipient to the conference. The recipient need not know the conference connection information.

A **blastvoice** call considers an answering machine response a success as well as a human voice. When a blast reaches a successful answer, it plays the blast message.

OCP Methods

This section contains the methods for managing users and making use of IntelPeer's services. Before you make these calls, you must use your IntelPeer provided enterprise ID, secret, and the current GMT Unix time to request a session ID.

You pass the session ID to each method. Each method has a set of variables. The system requires you to supply all the variables specified in a method's syntax.

All of the OCP methods return an XML response. When a response fails, the system returns an error code and text. This text may or may not be suitable for presentation to your end users.

Note: To obtain the URL you use to make REST API calls, speak with your IntelPeer representative.

Resources

The REST API acts on the following resources:

Resource	Description
<code>session</code>	A session on the IntelPeer service.
<code>user</code>	A human being who is using the service, also known as an IntelPeer system user
<code>phonenumbers</code>	A telephone number.
<code>huntorder</code>	A list of a user's phone numbers in the order the user wishes to be contacted.
<code>calltimeexception</code>	The phone number to contact a user on during a specific time period.
<code>dnd</code>	A period of time when the user wishes not to be disturbed by calls.
<code>blacklist</code>	A list of numbers the user wishes to block.
<code>email</code>	A user's email address.
<code>carriers</code>	A telephone carrier associated with a user's cell phones.
<code>currency</code>	The amount of money associated with a user.
<code>minutesallowed</code>	The number of minutes a user is permitted to use in the system.

<code>mouused</code>	The amount of time a user has used on the system.
<code>group</code>	A grouping of contacts associated with a user.
<code>call</code>	An immediate call between two points.
<code>autoconference</code>	A conference that a user elects to take part in.
<code>conference</code>	A conference that a user schedules.
<code>blastvoice</code>	A call that goes out to multiple destinations, usually an announcement.
<code>ttspreview</code>	A text-to-speech conversion preview.
<code>Smsphone</code>	Represents a phone number that accepts SMS calls.

Session IDs and Authorization Headers

Before you can use an OCP method, you must first obtain a session ID. You use the IntelPeer-provided enterprise ID, secret, and the current GMT Unix time to request a session ID. The time must be within 5 minutes of the server's GMT time. Before making the request, create an MD5 hash of the time and the secret. Then, request a session ID by sending an Authorization header in the following format:

```
Basic enterpriseid:gmttime:md5hash
```

After you obtain a session ID, each subsequent call your application has an Authorization header that includes the supplied session ID. This header should have the format:

```
Basic enterpriseid:sessionId
```

In practice, a typical Authorization header will appear as follows:

```
Basic 1:1cac4a15372b15f4ff37292f3ed909e6edeb0d7a0ebc6a7e2a4c7d943904ddab
```

When obtaining a session, the Authorization header has a different format. See `session Get` for more information on forming this Authorization header.

Specifying Origination and Destination

Multiple methods in the API require you to construct an **origination** and/or **destination** value. These parameters take a type-value pair in the following format:

```
origination=type:value
destination= type:value
```

The *type* is a required, numeric specifying what *value* you are passing. You separate these with a colon. For some methods, you can specify multiple *type:value* pairs. In this case, you separate each pair with a comma and no intervening spaces as follows:

```
destination= type:value,type:value,type:value
```

Valid types are 1, 2, or 3 and they take the following value_strings:

- 1 an IntelePeer `userId`.
- 2 an email address
- 3 a phone number

For example, if you specified a `type` of 2, you must specify a valid email address. If you specify an email (2) or a `userId` (1), the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call. You can mix and match the different types. For example, you can send:

```
destination=3:3035551212,1:157,3:7205551212,2:user@yahoo.com
```

Specifying a Phone Number Value

When specifying a phone number, you must provide the proper format for domestic and international numbers in the `mDigits` field. If you are specifying a domestic number, simply enter the digits. For international calls, you must provide a delimiting space between the phone number and the country and city code. The format for international numbers is as follows:

```
countrycode citycode phonenumber
```

The following is recognized by the API as a domestic number as it has 10 digits without spaces:

```
3035551212
```

The following number with a country code of 1 the API also recognizes as a domestic number:

```
1 3035551212
```

The following international number is for London. The country code is 44 = UK, city code 20 = London, the remainder the system recognizes as the phone's digits:

```
44 20 77994044
```

The following is a number in Paris (country code 33 = France, city code 1 = Paris):

```
33 1 49 54 46 46
```

The following are invalid international numbers. In the first, 4420 is an invalid country code and in the next it is more than 10 digits and there is no country code specified:

```
4420 77994044  
33149544646
```

Data Types

OCP's REST methods accept the following data types:

double	A number with up to 2 decimal places. The system allows numbers and one decimal point allowed; it strips all other characters.
DTMF	Telephone keypad values (numbers, # and *) only. The system strips all other characters including letters and decimals, for example: <ul style="list-style-type: none"> • <code>123456#</code> becomes <code>123456#</code> • <code>###ab44</code> becomes <code>**##44</code> • <code>default pin</code> becomes an empty string
email	A valid email address. The system validates both the format and the DNS record.
numeric	Numbers only 0-9. The system strips all other characters including letters and decimals. For example: <ul style="list-style-type: none"> • <code>1</code> becomes <code>1</code> • <code>One</code> becomes an empty string • <code>Test5</code> becomes <code>5</code> • <code>54.01</code> becomes <code>5401</code>
phone	If you are specifying a domestic number, simply enter the digits. For international calls, you must provide a delimiting space between the phone number and the country and city code. The format for international numbers is as follows: <p><i>countrycode citycode phonenumber</i></p> <p>Do not provide delimiters in domestic numbers. For international numbers supply only spaces for delimiters.</p>
Unix time stamp	The number of seconds elapsed since midnight Coordinated Universal Time/Greenwich Mountain Time (UTC/GMT) of January 1, 1970. The system accepts numbers only and strips all other characters including letters and decimals.
string	The strips only HTML and PHP tags, all other characters it considers valid. For example: <ul style="list-style-type: none"> • <code>Hello</code> becomes <code>Hello</code> • <code>Hi there</code> becomes <code>Hi there</code> • <code><? I'm a hacker ?> named joe</code> becomes <code>named joe</code>

Optional Variables

You can send two additional variables with any REST query:

- output** You can set this to `json` or `xml`. JSON is a Javascript-oriented response that is faster to process and more convenient to work with for many developers. See <http://en.wikipedia.org/wiki/Json> for more information.
- The default output format is xml.
- verbose** Set this to 1 (one) to see verbose output. Set this to 0 (zero) to turn this off. Occasionally, the HTTP response headers make the XML output redundant. For example, an HTTP response code of 200 ("OK") with the XML of `<status>ok</status>`.

Handling Common Errors Returned from REST

Each OCP method can return error codes from the underlying system. You will find these error codes documented in the **Errors** section of each method page. Additionally, the REST version of OCP has a number of error codes specific to it. These error codes are within the range 900-999.

Typically, the 900-999 error codes deal with client request data. The system validates the request data received from the client cleaning it to prevent hacking and data injection. The system also verifies the format of data such as phone numbers, email, and so forth.

Any of the REST APIs can return the 900 error codes. Your application should be prepared to handle them. The following table lists the error codes and their accompanying messages. The system replaces the *name* and *value* elements of each message with the appropriate data from the client request.

900	<i>name</i> is required but was not received
901	<i>name</i> must be one of the following: ____, ____, ____
902	<i>name (value)</i> does not contain a valid value
903	<i>name (value)</i> does not meet the length requirements. It must be between __ and __ in length
912	<i>name (value)</i> does not contain a valid amount
913	<i>name (value)</i> does not contain a valid number
915	<i>name (value)</i> does not contain a valid email
916	<i>name (value)</i> does not contain a valid date
917	<i>name (value)</i> does not contain a valid time
918	<i>name (value)</i> does not contain a valid datetime
920	<i>name (value)</i> does not contain a valid phone
921	<i>name (value)</i> does not contain a valid dtmf"
922	<i>name (value)</i> does not contain a valid user id
923	<i>name (value)</i> does not contain a valid group"

session GET

Starts a new session for an application and returns a session ID. To obtain a session you use the IntelePeer-provided enterprise ID, secret, and the GMT Unix time to request a session ID. A Unix time stamp is the number of seconds elapsed since midnight Coordinated Universal Time/Greenwich Mountain Time (UTC/GMT) of January 1, 1970. The time must be within 5 minutes of the server's GMT time or the system rejects the session.

Before making the request, create an MD5 hash of the time and the secret. Then, request a session ID by sending an Authorization header in the following format:

Basic enterpriseid:gmttime:md5hash

Once the system responds with a sessionId, you must use it in your subsequent calls to format an Authorization header. This authorization header should have the format:

Basic enterpriseid:sessionId

Sessions expire after 24 hours. If your session expires, you must request a new session before proceeding.

URL

`/session`

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <sessionId>18fa6fd3d1e3ac4fe5684e41723069eecf70232cc2e0 619dbfb8a0c23a36fb2a</sessionId> </ResponseData></pre>

Errors

- 100 Incorrect timestamp in request
- 101 Unknown enterprise id
- 102 Invalid credentials
- 114 Internal error: Authentication system failed

Example

Assume that IntelPeer provides you with an enterprise ID of 1 and a secret value of 234. If the current GMT time is 234, you would MD5 transform the time and secret as one string **2345679**. For example:

```
md5(23456789)
```

Then, you send the resulting hash value with your session request, for example:

```
Basic 1:234:4428c6c474502e61151877825bb41961
```

user GET

Retrieves the user information associated with the ID. The information contains only the non-phone attributes of the user. To get information regarding specific phone attributes such as a user's default phone number or calling preferences, you will need to use the specific **GET** methods for those objects.

URL

/user/userId

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <user> <firstName>John</firstName> <lastName>Doe</lastName> <address1>1234 Main Street</address1> <address2></address2> <city>Springfield</city> <state>IL</state> <zip></zip> <gender></gender> </user> </ResponseData></pre>

Errors

106	User account does not exist
105	Internal error: SQL execution failed

user POST

Creates a system user and, upon success, returns a `userId` for the user. You use this `userId` in subsequent call management methods. You must specify a `firstName` and `lastName` value as well as a unique the `email` value, a `phoneNumber`, and `phoneType`.

By default, the system does not associate the user you create with your enterprise. This means that the user is accessible to all the enterprises on the IntelPeer OCP platform. To associate the user exclusively with your enterprise, set `exclusive` to 1. This ensures that only applications with your `enterpriseId` can manipulate the user's call preferences or attributes.



The system always associates a user's MOU and accounting values with a `enterpriseId` and `userId` combination regardless of how you set the `exclusive` value.

If the user already exists, the system returns the user's current information `userId`.

URL

`/user`

Request

<code>firstName</code>	A required string specifying the user's first name.
<code>lastName</code>	A required string specifying the user's last name.
<code>email</code>	A unique email address. Addresses that do not resolve return an error. This is required.
<code>phoneNumber</code>	A required value specifying the phone number to add.
<code>phoneType</code>	A numeric value that identifies the phone as one of the following: <ul style="list-style-type: none"> 1 office 2 mobile 3 home 4 other This is required.
<code>carrierId</code>	A numeric value specifying the ID of the service provider. Specify this value only for cell phones.
<code>address1</code>	An optional string specifying the street address of the user.
<code>address2</code>	An optional string specifying the second half of the user's street address.
<code>city</code>	An optional string specifying the user's city.

state	An optional string specifying the user's state.
zip	An optional string specifying the user's zip code.
country	An optional string specifying the user's country.
gender	An optional string specifying the user's gender. You can specify an M or an F .
exclusive	An optional Boolean value specifying that the system should associate the user with the enterpriseId .

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <responseData> <userId>229</userId> </responseData></pre>

Errors

103	User account already exists
105	Internal error: SQL execution failed

user DELETE

Deletes a user and its associated data from the IntelPeer system. Before deleting a user, your application is responsible for checking if the user (`userId`) is participating in any scheduled calls. Pending calls against the user's `userId` do not complete if you remove the user before the call. If a pending call is against the user's phone number, that call will complete.

URL

`/user/{userId}`

Response

HTTP Response Code:	200 ("OK")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105	Internal error: SQL execution failed
107	Incorrect email address for user
108	Cannot delete primary email address

user PUT

Updates a user's non-phone attributes such as name, address, and gender.

URL

/user/userId

Request

address1	An optional string specifying the street address of the user.
address2	An optional string specifying the second half of the user's street address.
city	An optional string specifying the user's city.
state	An optional string specifying the user's state.
zip	An optional string specifying the user's zip code.
country	An optional string specifying the user's country.
gender	An optional string specifying the user's gender. You can specify an M or F.

Response

HTTP Response Code:	200 ("OK")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

106	User account does not exist
105	Internal error: SQL execution failed

phonenumber GET

Gets the list of phone numbers associated with the system user at a specific time. This method takes into account a user's call-time-exceptions. Meaning, it returns only those numbers in effect at the specific point in time.

If you do not specify a time, the system uses the current time. If you specify the `all` option, the system ignores the time if you specified it and instead returns all of a user's phone numbers.

URL

`/phonenumber`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .
<code>all</code>	An optional value specifying 0 (no) or 1 (yes). The default is 0 (no) meaning the system does not return all the numbers and instead uses the specified <code>time</code> to resolve the numbers. If you specify 1 (yes), the system returns all of a user's numbers and ignores the <code>time</code> value if you supply one.
<code>time</code>	An optional Unix time stamp specifying the time. The system verifies which phone numbers are available at that time. If you do not specify a time, the system uses the current time and places the call immediately.

Response

For returning all numbers	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <numbers> <number> <id>861</id> <countryCode>1</countryCode> <cityCode /> <phoneType>2</phoneType> <digits>3035551212</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <isDefault>0</isDefault> </number> <number> <id>867</id> <countryCode>1</countryCode></pre>

	<pre> <cityCode /> <phoneType>1</phoneType> <digits>3039844345</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <isDefault>0</isDefault> </number> </numbers> </ResponseData> </pre>
Response for returning a specific number	
HTTP Response Code:	200 ("OK")
Body:	<pre> <?xml version="1.0" ?> <ResponseData> <tryall>0</tryall> <dnd>0</dnd> <number> <id>859</id> <countryCode>1</countryCode> <cityCode /> <phoneType>2</phoneType> <digits>7202108298</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> </number> </ResponseData> </pre>

Errors

105	Internal error: SQL execution failed
106	User account does not exist
110	Phone number not found

phoneNumber POST

Adds a new phone number to a system user. Each phone number requires you to specify two values, the `phoneNumber` and the `phoneType`. If the `phoneNumber` resolves to a cell phone, you must also specify the `carrierId`.

URL

`/phoneNumber`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> . This is required.
<code>phoneNumber</code>	A required value specifying the phone number to add.
<code>phoneType</code>	A numeric value that identifies the phone as one of the following: <ul style="list-style-type: none"> 1 office 2 mobile 3 home 4 other This is required.
<code>carrierId</code>	A numeric value specifying the ID of the service provider. Specify this value only for a cell phone.

Response

HTTP Response Code:	201 ("Created")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105	Internal error: SQL execution failed
106	User account does not exist

phonenumber PUT

Set a phone number as a default or, alternatively, changes the number or type associated with an existing number. This method does not change any of the calltime exceptions or rules associated with the phone.

URL

/phonenumber
/phonenumber/{existing number}

Request

- userId** The user's numeric ID. The system assigns this value when you call `user POST`. This is required.

- phoneNumber** A required value specifying the phone number to add. This is optional when changing an existing number.

- phoneType** A numeric value that identifies the phone as one of the following:
 - 1 office
 - 2 mobile
 - 3 home
 - 4 other

This is required. This is optional when changing an existing number.

- carrierId** A numeric value specifying the ID of the service provider. Specify this value only for a cell phone.

Response

HTTP Response Code:	200 ("OK")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

- 105 Internal error: SQL execution failed

- 106 User account does not exist

- 110 Phone number not found

Example

That means that to change from 5555551212 to 4445551212 you would specify a URL of:

```
PUT /phonenumber/5555551212
```

And supply the data of:

```
phoneNumber=4445551212
```

phoneNumber DELETE

Removes one or more phone numbers from the system user with the specified `userId`.

URL

`/phoneNumber`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`.

`phoneNumber` The number to remove.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Error

105 Internal error: SQL execution failed

109 Cannot delete primary phone number

smsphone GET

Gets the list of SMSphone numbers associated with the system user at a specific time. This method takes into account a user's call-time-exceptions. Meaning, it returns only those numbers in effect at the specific point in time.

If you do not specify a time, the system uses the current time. If you specify the `all` option, the system ignores the time if you specified it and instead returns all of a user's phone numbers.

URL

`/smsphone`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .
<code>all</code>	An optional value specifying 0 (no) or 1 (yes). The default is 0 (no) meaning the system does not return all the numbers and instead uses the specified <code>time</code> to resolve the numbers. If you specify 1 (yes), the system returns all of a user's numbers and ignores the <code>time</code> value if you supply one.
<code>time</code>	An optional Unix time stamp specifying the time. The system verifies which phone numbers are available at that time. If you do not specify a time, the system uses the current time and places the call immediately.

Response

For returning all numbers	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <responseData> <numbers> <number> <id>861</id> <countryCode>1</countryCode> <cityCode /> <phoneType>2</phoneType> <digits>3035551212</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <isDefault>0</isDefault> </number> <number> <id>867</id> <countryCode>1</countryCode></pre>

	<pre> <cityCode /> <phoneType>1</phoneType> <digits>3039844345</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <isDefault>0</isDefault> </number> </numbers> </ResponseData> </pre>
Response for returning a specific number	
HTTP Response Code:	200 ("OK")
Body:	<pre> <?xml version="1.0" ?> <ResponseData> <tryall>0</tryall> <dnd>0</dnd> <number> <id>859</id> <countryCode>1</countryCode> <cityCode /> <phoneType>2</phoneType> <digits>7202108298</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> </number> </ResponseData> </pre>

Errors

105	Internal error: SQL execution failed
106	User account does not exist
110	Phone number not found

smsphone POST

Adds a new SMS phone number to a system user. Each phone number requires you to specify two values, the `phonenumber` and the `phoneType`. If the `phonenumber` resolves to a cell phone, you must also specify the `carrierId`.

URL

`/smsphone`

Request

- `userId` The user's numeric ID. The system assigns this value when you call `user POST`. This is required.

- `phonenumber` A required value specifying the phone number to add.

- `phoneType` A numeric value that identifies the phone as one of the following:
 - 1 office
 - 2 mobile
 - 3 home
 - 4 other
 This is required.

- `carrierId` A numeric value specifying the ID of the service provider. Specify this value only for a cell phone. This is optional.

Response

HTTP Response Code:	201 ("Created")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

- 105 Internal error: SQL execution failed

- 106 User account does not exist

smsphone DELETE

Removes one or more SMS phone numbers from the system user with the specified `userId`.

URL

`/smsphone`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`.

`phone` The number to remove.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Error

105 Internal error: SQL execution failed

109 Cannot delete primary phone number

huntorder PUT

Adds or more phones to the end of a hunt order. This method also toggles the try-all attribute on a system user.

URL

/huntorder

Request

userId	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .
tryall	An optional numeric value setting or unsetting the try-all attribute. When 0 (zero), the system uses the hunt order. When this is 1 (one), the system calls all the user's numbers simultaneously. The default is 0 (zero).
phoneNumber	An optional value specifying the phone number to add at the end of the hunt order.
phoneType	An optional numeric value that identifies the phone as one of the following: <ul style="list-style-type: none"> 1 office 2 mobile 3 home 4 other <p>You should only specify this value if adding a new phone.</p>
carrierId	A numeric value specifying the ID of the service provider. Specify this value only for cell phones.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105	Internal error: SQL execution failed
-----	--------------------------------------

huntorder POST

Sets a hunt order for a system user. Calling this method overwrites an existing hunt sequence.

Passing a value in the `tryall` variable sets or unsets this attribute on the user. If `tryall` is 1 (one), the system ignores the hunt order and rings all the phones simultaneously. If it is 0 (zero), the system rings the numbers in the specified order until a person answers the call.

You can call this method with phone numbers that do not yet exist on the user. The system adds the number to the user.

URL

`/huntorder`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .
<code>tryall</code>	An optional numeric value setting or unsetting the try-all attribute. When 0 (zero), the system uses the hunt order. When this is 1 (one), the system calls all of a user's numbers simultaneously. The default is 0 (zero).
<code>phoneNumber</code>	A required value specifying the phone number to add as the default.
<code>phoneType</code>	A numeric value that identifies the phone as one of the following: <ul style="list-style-type: none"> 1 office 2 mobile 3 home 4 other You should only specify this value if adding a new phone.
<code>carrierId</code>	A numeric value specifying the ID of the service provider. Specify this value only for cell phones.

Response

HTTP Response Code:	201 ("Created")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

huntorder GET

Gets a list of the user's current numbers in the hunt order. The system also returns the current value of the user's try-all attribute.

URL

/huntorder

Request

userId The user's numeric ID. The system assigns this value when you call `user POST`.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <tryall>1</tryall> <numbers> <number> <digits>7205551212</digits> <phoneType>2</phoneType> <id>859</id> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <default>1</default> </number> <number> <digits>3035551212</digits> <phoneType>1</phoneType> <id>867</id> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <default>0</default> </number> </numbers> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

huntorder DELETE

Deletes all the entries in a user's hunt order.

URL

/huntorder

Request

userId The user's numeric ID. The system assigns this value when you call **user POST**.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

calltimeexception POST

Sets call time exceptions on the system user. A call time exception defines a number to call for a specified time period. A system user may have multiple call-time exceptions. This exception supersedes default numbers and hunt sequences.

If the time period specified in this exception overlaps another already on the user, the new one will overwrite the previous exception where they overlap. For example, a system user may have a number preference from 9 AM to Noon and the new exception is from 11:30 to 5:00 PM. After you call this method, the user has two call time exceptions — one from 9 AM to 11:30 and another from 11:30 to 5:00 PM.

URL

`/calltimeexception`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .														
<code>startHour</code>	A required number specifying the hour the exception should start. This value ranges from 0 to 23.														
<code>endHour</code>	A required number specifying the hour the exception should end. This value ranges from 0 to 23.														
<code>day</code>	A required numeric value that represents the days when the system should enforce the exception. This value is the sum of the following bitmask values: <table><tr><td>1</td><td>Sunday</td></tr><tr><td>2</td><td>Monday</td></tr><tr><td>3</td><td>Tuesday</td></tr><tr><td>4</td><td>Wednesday</td></tr><tr><td>5</td><td>Thursday</td></tr><tr><td>6</td><td>Sunday</td></tr><tr><td>7</td><td>Sunday</td></tr></table>	1	Sunday	2	Monday	3	Tuesday	4	Wednesday	5	Thursday	6	Sunday	7	Sunday
1	Sunday														
2	Monday														
3	Tuesday														
4	Wednesday														
5	Thursday														
6	Sunday														
7	Sunday														
<code>phoneType</code>	An optional numeric value that specifies the type of <code>phoneNumber</code> you are supplying. This is one of the following values: <table><tr><td>1</td><td>office</td></tr><tr><td>2</td><td>mobile</td></tr><tr><td>3</td><td>home</td></tr><tr><td>4</td><td>other</td></tr></table>	1	office	2	mobile	3	home	4	other						
1	office														
2	mobile														
3	home														
4	other														

<code>phoneNumber</code>	The required, numeric phone number associated with this exception.
<code>carrierId</code>	An optional numeric value specifying the ID of the service provider. Specify this value only for cell phones.
<code>timezone</code>	An optional numeric value that specifies an offset from GMT. You can specify a range from -11 to +12. The default is 0 (zero)

Response

HTTP Response Code:	201 ("Created")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

See Also

The `calltimeexception DELETE` method on page 41.

calltimeexception DELETE

Removes one or more call-time exceptions from a user by overwriting the existing call time exceptions. You must explicitly declare the call times to remove. Meaning that if an exception is currently set for MTWTF for from 9-10 AM then you must set all 5 days in the `day` value.

URL

`/calltimeexception`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .														
<code>phoneNumber</code>	The required, numeric phone number associated with this exception.														
<code>startHour</code>	A required number specifying the hour the exception should start. This value ranges from 0 to 23. When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.														
<code>endHour</code>	A required number specifying the hour the exception should end. This value ranges from 0 to 23. When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.														
<code>day</code>	<p>A required numeric value that represents the days when the system should enforce the exception. This value is the sum of the following bitmask values:</p> <table> <tr><td>1</td><td>Sunday</td></tr> <tr><td>2</td><td>Monday</td></tr> <tr><td>3</td><td>Tuesday</td></tr> <tr><td>4</td><td>Wednesday</td></tr> <tr><td>5</td><td>Thursday</td></tr> <tr><td>6</td><td>Sunday</td></tr> <tr><td>7</td><td>Sunday</td></tr> </table> <p>When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.</p>	1	Sunday	2	Monday	3	Tuesday	4	Wednesday	5	Thursday	6	Sunday	7	Sunday
1	Sunday														
2	Monday														
3	Tuesday														
4	Wednesday														
5	Thursday														
6	Sunday														
7	Sunday														
<code>phoneType</code>	<p>An optional numeric value that specifies the type of <code>phoneNumber</code> you are supplying. This is one of the following values:</p> <table> <tr><td>1</td><td>office</td></tr> <tr><td>2</td><td>mobile</td></tr> <tr><td>3</td><td>home</td></tr> <tr><td>4</td><td>other</td></tr> </table> <p>When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.</p>	1	office	2	mobile	3	home	4	other						
1	office														
2	mobile														
3	home														
4	other														

all An optional value specifying 0 (no) the system should not delete all exceptions or 1 (yes). The default is 0 (no).

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

calltimeexception GET

Gets the call-time exceptions on a system user.

URL

/calltimeexception

Request

userId The user's numeric ID. The system assigns this value when you call `user POST`.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre> <?xml version="1.0" ?> <ResponseData> <exceptions> <exception> <starthour>6</starthour> <stophour>8</stophour> <days>28</days> <expiration /> <number> <id>859</id> <countryCode>1</countryCode> <cityCode /> <phoneType>2</phoneType> <digits>7205551212</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <isDefault>0</isDefault> </number> </exception> <exception> <starthour>6</starthour> <stophour>6</stophour> <days>96</days> <expiration /> <number> <id>908</id> <countryCode>1</countryCode> <cityCode /> <phoneType>1</phoneType> <digits>3035551212</digits> <carrierId>0</carrierId> <validationStatus>2</validationStatus> <isDefault>1</isDefault> </number> </exception> </exceptions> </ResponseData> </pre>

	<pre></exceptions> </responseData></pre>
--	--

Errors

105 Internal error: SQL execution failed

See Also

`calltimeexception DELETE` on page 41.

dnd POST

Sets a do-not-disturb (DND) call time exception on the system user. A DND call time exception defines a specified time period when the user will not accept phone calls. A system user may have multiple DND call time exceptions. This exception supersedes default numbers and hunt sequences.

If the time period specified in this exception overlaps another already on the user, the new one will overwrite the previous exception where they overlap. For example, a system user may have a DND preference from 9 AM to Noon and the new exception is from 11:30 to 5:00 PM. After you call this method, the user has two call time exceptions — one from 9 AM to 11:30 and another from 11:30 to 5:00 PM.

URL

/dnd

Request

userId	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .
startHour	A required number specifying the hour the exception should start. This value ranges from 0 to 23.
endHour	A required number specifying the hour the exception should end. This value ranges from 0 to 23.
day	A required numeric value that represents the days when the system should enforce the exception. This value is the sum of the following bitmask values: <ul style="list-style-type: none"> 1 Sunday 2 Monday 3 Tuesday 4 Wednesday 5 Thursday 6 Sunday 7 Sunday
timezone	An optional numeric value that specifies an offset from GMT. You can specify a range from -11 to +12. The default is 0 (zero)

Response

HTTP Response Code:	201 ("Created")
---------------------	-----------------

Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>
----------------------	--

Errors

105 Internal error: SQL execution failed

dnd GET

Gets the do-not-disturb (DND) call-time exceptions currently assigned to a user. This array represents the user's do-not-disturb schedule.

URL

/dnd

Request

userId A required, numeric value representing the user's ID. The system assigns this value when you call `user POST`. This is required.

Response

None found	
HTTP Response Code:	204 No Content
Body:	empty
One or more found	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <exceptions> <exception> <starthour>6</starthour> <stophour>8</stophour> <days>28</days> <expiration /> </exception> <exception> <starthour>6</starthour> <stophour>6</stophour> <days>96</days> <expiration /> </exception> </exceptions> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

dnd DELETE

Deletes one or more call-time exceptions from a user by overwriting the existing call time exceptions. You must explicitly declare the call times to remove. Meaning that if an exception is currently set for MTWTF for from 9-10 AM then you must set all 5 days in the call-time exception.

If you set the `all` option, the system removes all of a user's `dnd` exceptions. If you set `all`, you need not specify `startHour`, `endHour`, `day`, or `timezone`. If you do specify these values, the system ignores them.

URL

`/dnd`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> . This is required.
<code>startHour</code>	A numeric specifying the hour the exception should start. This value ranges from 0 to 23. When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.
<code>endHour</code>	A numeric value specifying the hour the exception should end. This value ranges from 0 to 23. When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.
<code>day</code>	A numeric value that represents the days when the system should enforce the exception. This value is the sum of the following bitmask values: <ul style="list-style-type: none"> 1 Sunday 2 Monday 3 Tuesday 4 Wednesday 5 Thursday 6 Sunday 7 Sunday When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.
<code>timezone</code>	An optional numeric value that specifies an offset from GMT. You can specify a range from -11 to +12. The default is 0 (zero). When <code>all</code> is 1 (yes), you need not specify this value. Otherwise, it is required.
<code>all</code>	An optional value specifying 0 (no) the system should not delete all DND exceptions

or 1 (yes). The default is 0 (no).

Response

HTTP Response Code:	200 ("OK")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

blacklist POST

Blocks a user or telephone number from calling a user. You can block by user ID, email, or phone. If you specify an email or a user ID, the system resolves these to a user account. The system blocks all the numbers associated with the user account. This block applies only to calls originating within your application.

Use the `blacklist DELETE` method to remove the phone number from the list.

URL

`/blacklist`

Request

<code>userId</code>	The user's numeric ID. The system assigns this value when you call <code>user POST</code> . This is required.
<code>terminationType</code>	A required numeric specifying the type of termination value. Valid values are 1, 2, or 3. The following lists the valid types and what <code>terminationValue</code> you must supply for them. <ul style="list-style-type: none"> 1 an IntelPeer <code>userId</code>. 2 an email address 3 a phone number
<code>terminationValue</code>	A required value defining the blocked value. This value depends on the <code>terminationType</code> . For example, if you specified a <code>terminationType</code> of 2, you must specify a valid email address. If you specify an email or a <code>userId</code> , the system resolves these to a user account.

Response

HTTP Response Code:	201 ("Created")
Body: (verbose only)	<pre><ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105	Internal error: SQL execution failed
113	Duplicate phone number

See Also

The methods `blacklist GET` on page 52 and `Errors`

105 `Internal error: SQL execution failed`

`blacklist DELETE` on page 52.

blacklist GET

Retrieves a user's blacklist.

URL

/blacklist

Request

userId The user's ID. The system assigns this value when you call `user POST`.

Response

None found:	
HTTP Response Code:	204 No Content
Body:	Empty
One or more items on the blacklist	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <contact> <type>1</type> <id>227</id> </contact> <contact> <type>3</type> <id>3035551212</id> </contact> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

blacklist DELETE

Removes a number or user previously added to a user’s blacklist. Once removed, the number or user can call the user.

URL

`/blacklist`

Request

- userId** The user’s numeric ID. The system assigns this value when you call `user POST`.
- terminationType** A required, numeric specifying the type of termination value. Valid values are 1, 2, or 3. The following lists the valid types and what `terminationValue` you must supply for them.
 - 1 an IntelePeer `userId`.
 - 2 an email address
 - 3 a phone number
- terminationValue** A required value defining the blocked value. This value depends on the `terminationType`. For example, if you specified a `terminationType` of 2, you must specify a valid email address. If you specify an email or a `userId`, the system resolves these to a user account.

Response

HTTP Response Code:	200 ("OK")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

- 105 Internal error: SQL execution failed
- 113 Duplicate phone number

email POST

Adds an email addresses to a user's preferences. Each address must be unique. The system ensures the addresses resolve.

URL Errors

105	Internal error: SQL execution failed
106	User account does not exist

/email

Request

<code>userId</code>	A required, numeric user ID. The system assigns this value when you call <code>user POST</code> .
<code>email</code>	An email address to add. This is required.

Response

HTTP Response Code:	201 ("Created")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status> email address user@domain.com added</status> </ResponseData></pre>

Errors

105	Internal error: SQL execution failed
106	User account does not exist

email GET

Returns the user's single default email and an array of all the other email addresses in the user's preferences.

URL

/email

Request

userId The user's numeric ID. The system assigns this value when you call `user POST`.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <emails> <default>tester@intelepeer.com</default> <address>tester@intelepeer.com</address> <address>jward@intelepeer.com</address> </emails> </ResponseData></pre>

See Also

The methods **Errors**

105 Internal error: SQL execution failed

113 Duplicate phone number

email on page 53 and **Errors**

105 Internal error: SQL execution failed

106 User account does not exist

email on page 55.

Errors

105 Internal error: SQL execution failed

106 User account does not exist

email DELETE

Removes one or more user email addresses from user.

URL

/email

Request

userId The user's numeric ID. The system assigns this value when you call **user POST**.

email The email address to remove. This value is required.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>name@domain.com deleted from user 227</status> </ResponseData></pre>

Errors

- 105 Internal error: SQL execution failed
- 107 Incorrect email address for user
- 108 Cannot delete primary email address

email PUT

Sets the default email address in a user's preferences. The user's default email address must be unique.

URL

/email

Request

userId The user's numeric ID. The system assigns this value when you call `user POST`.

Email The email address to set as the default. This is required.

Response

HTTP Response Code:	200 ("OK")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>name@domain.com has been set as default email for user 227</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

106 User account does not exist

carriers GET

Returns the carriers associated with a user (`userId`) through its associated phone numbers. For each carrier, the system returns the carrier ID, name, and gateway.

URL

`/carriers`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`. This is required.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <carriers> <carrier> <id>0</id> <name>(unknown)</name> <gateway></gateway> </carrier> <carrier> <id>1</id> <name>AT&T</name> <gateway>txt.att.net</gateway> </carrier> <carrier> <id>2</id> <name>Verizon</name> <gateway>vtext.com</gateway> </carrier> <carrier> <id>3</id> <name>Sprint</name> <gateway>messaging.sprintpcs.com</gateway> </carrier> <carrier> <id>4</id> <name>Nextel</name> <gateway>messaging.nextel.com</gateway> </carrier> <carrier> <id>5</id> <name>T-Mobile</name> <gateway>tmomail.net</gateway> </carriers> </ResponseData> </xml></pre>

	<pre></carrier> </carriers> </ResponseData></pre>
--	---

Errors

- 105 Internal error: SQL execution failed
- 111 Internal error: System data missing or corrupt

currency GET

Retrieves the current amount of the user's currency.

URL

`/currency`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <amount>50.25</amount> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

currency POST

Set's the user's currency to the specified `amount`. The system allows you to specify a negative integer for the `amount`.

URL

`/currency`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`.

`amount` A required, double value that specifies the amount to set for the user currency.

Response

HTTP Response Code:	201 ("Created")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

currency PUT

Increments the user's currency by the specified `amount`. The system allows you to specify a negative integer for the `amount`.

URL

`/currency`

Request

`userId` The user's numeric ID. The system assigns this value when you call user POST.

`amount` A required, double value that specifies the amount by which to increment the user currency.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

currency DELETE

Decrements the specified **amount** from the user's currency. A user can have a negative currency value.

URL

`/currency`

Request

userId The user's numeric ID. The system assigns this value when you call **user POST**.

amount A required, double value that specifies the amount to set for the user's currency.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData</pre>

Errors

105 Internal error: SQL execution failed

mouallowed GET

Gets the user's allowed minutes-of-use (MOU).

URL

/mouallowed

Request

userId The user's numeric ID. The system assigns this value when you call **user POST**. This is required.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <mou>50</mou> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

mouallowed POST

Assign the minutes-of-use (MOU) for a user.

URL

/mouallowed

Request

userId The user's numeric ID. The system assigns this value when you call `user POST`.

mou A required numeric value specifying the minutes to allocate.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

See Also

The methods `Errors`

105 Internal error: SQL execution failed

mouallowed GET on page 63.

mouused GET

Retrieves the MOU for the specified `userId`.

URL

`/mouused`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`. This is required.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <mou>25</mou> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

See Also

The `Errors`

105 Internal error: SQL execution failed

`mouallowed GET` method on page 63.

mouused DELETE

Resets the MOU for the specified `userId` to 0 (zero).

URL

`/mouused`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

See Also

The **Errors**

105 Internal error: SQL execution failed

`mouallowed GET` method on page 63.

group POST

Creates a call group for the specified `userId`. or adds a new member to an existing group. For example, a group called “Family” would hold all of a user’s family numbers. To do this you, you would:

1. Create a group.
2. Obtain the `groupId` from the response.
3. Add one or more members to the group by specifying a destination.

To add a group member, you create a URL like the following:

```
/group/141/member/3:3035551212
/group/141/member/2:joe@yahoo.com
```

URL

```
/group
/group/groupId
/group/groupId/member/destination
```

Request

<code>userId</code>	The user’s numeric ID. The system assigns this value when you call <code>user POST</code> . This is required.
<code>groupName</code>	A string representing the group’s display name. This value is required when adding a new group.
<code>destination</code>	A <code>type:value</code> pair specifying a destination. You can specify any of the following combinations: <ul style="list-style-type: none"> 1: <code>userId</code>. 2: <code>email_address</code> 3: <code>phone_number</code> If you specify an email or a <code>userId</code> , the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.

Response for Creating a New Group

HTTP Response Code:	201 (“Created”)
Body:	<pre><?xml version="1.0" ?> <ResponseData> <groupid>145</groupid> </ResponseData></pre>

Response for Adding a Group Member with groupId

HTTP Response Code:	201 ("Created")
Body: (verbose only)	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

- 105 Internal error: SQL execution failed
- 115 Group not found

group GET

Gets the groups on the user preferences or gets the group members.

URL

/group
/group/groupId

Request

userId The user's numeric ID. The system assigns this value when you call **user POST**. This is required.

Response for All Groups

None found	
HTTP Response Code:	204 No Content
Body:	Empty
1 or more found	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <group> <id>126</id> <name>Coworkers</name> </group> <group> <id>145</id> <name>Family Members</name> </group> </ResponseData></pre>

Response for a groupID

None found	
HTTP Response Code:	204 No Content
Body:	Empty
1 or more found	

HTTP Response Code:	200 ("OK")
Body:	<pre> <?xml version="1.0" ?> <ResponseData> <member> <type>1</type> <id>227</id> </member> <member> <type>3</type> <id>3039844345</id> </member> </ResponseData> </pre>

Errors

105 Internal error: SQL execution failed

group PUT

Renames the group with the specified GID on the user preferences.

URL

`/group/groupId`

Request

userId The user's numeric ID. The system assigns this value when you call `user POST`. This is required.

groupName A required string specifying the group's new display name.

Response for Adding a New Group

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed

group DELETE

Deletes the group with the specified `groupId` or removes a member from the group. This removes the group only. Any users referenced by the group remain in the system; you are only deleting organizational structure.

Delete one or more members to the group by specifying a destination as follows:

```
/group/156/member/2:bigbob@yahoo.com
```

To delete multiple members, send multiple destinations as follows:

```
/group/156/member/2:bigbob@yahoo.com,11:alice@netcom.com,25:6504116525
```

URL

```
/group
/group/groupId
/group/groupId/member/destination
```

Request

userId	The user's numeric ID. The system assigns this value when you call <code>user POST</code> .
destination	<p>A <code>type:value</code> pair specifying a destination. You can specify any of the following combinations:</p> <ul style="list-style-type: none"> 1: <code>userId</code>. 2: <code>email_address</code> 3: <code>phone_number</code> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>

Response for Deleting a Group or Group Member

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

105	Internal error: SQL execution failed
112	Group not owned by user

call POST

Creates a two-way call bridge between the **origination** and **destination**. As with all OCP calls, by default the system first connects to the destination side. This call connects the destination side when it detects a human response. Answering machine responses are failures by default.

You can set the `callOrigination` value to 1 (true) to force the system to connect the origination side first. This is the opposite of the system's default behavior. This means the system will ignore constraints on the destination side and connect the call regardless of whether an answering machine picks up or not. The benefit of this feature is that the originator can leave a message if the destination reaches an answering machine or other voice mail system.

Upon answer, the system plays an announcement to the call destination instructing the recipient to press 1 to accept the call. When the recipient accepts the call, the system calls the originator and bridges the two call legs.

By default, the system attempts to detect an answering machine on the termination side. This means, when the system connects it waits to hear a human voice or an answering machine respond. If you are testing the system and you receive a call, you must respond to activate the call sequence. If you wish the system to ignore detection and go straight to the announcement, set `disableAMD`.

URL

`/call/callId`

Request

<code>userId</code>	The numeric call requester's user ID. The system assigns the ID when you create the user. This is required.
<code>origination</code>	A type:value pair specifying an origination. You can specify any of the following combinations: 1: userId. 2:email_address 3:phone_number If you specify an email or a <code>userId</code> , the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.
<code>destination</code>	A type:value pair specifying a destination. You can specify any of the following combinations: 1: userId. 2:email_address 3:phone_number If you specify an email or a <code>userId</code> , the system resolves these to a user account.

Then, it uses the call preferences on the user to identify the proper number to call.

callOrigination An integer value that when 1 (true), causes the system to connect to the origination side before the termination side. By default this value is 0 (false).

text A required string value specifying an announcement to play to the caller. The system converts this value to speech.

disableAMD Sets answering machine detection on or off.

Response

HTTP Response Code:	201 Created
Body:	<pre><?xml version="1.0" ?> <ResponseData> <callId>744</callId> </ResponseData></pre>

Errors

- 301 Invalid Termination Type supplied
- 302 Invalid Origination Type supplied
- 303 Blacklisted origination
- 304 Internal error: SQL execution failed

call GET

Returns the final status of the completed call specified by the `callIdentifier`. If a call has multiple legs, this method returns the final status for each leg. The `callStatus` returns a text string detailing the final status.

URL

`/call/callId`

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <userId>235</userId> <callId>1177</callId> <mou>0.2</mou> <success>1</success> <timeScheduled>1212186144</timeScheduled> <timeOriginated>1212186144</timeOriginated> <timeCompleted>1212186168</timeCompleted> <callingNumber>7202108298</callingNumber> <attempts> <attempt> <number>7209814345</number> <result>Success</result> </attempt> </attempts> <currentState>complete</currentState> <callComplete>1</callComplete> </ResponseData></pre>

Errors

106	User account does not exist
105	Internal error: SQL execution failed

autoconference POST

Dials and connects a user (`userId`) into a conference at the specified `conferenceTime`. The system connects to the user first using the user's call preferences. The `autoconference POST` method considers the connection to the user a success when it reaches a human voice, it ignores answering machines or voice mail.

Upon answering the call, the user does not need the conference number or pin. Instead, the system presents an announcement to the user. The user presses 1 to accept the call and the system joins the user to the conference. At this point, the system initiates the second leg of the call to the conference number. The system supplies the `conferencePin` to the conference engine and then bridges the conference (origination) and the user (termination) legs of the call.

By default, the system attempts to detect an answering machine on the termination side. This means, when the system connects it waits to hear a human voice or an answering machine respond. If you are testing the system and you receive a call, you must respond to activate the call sequence. If you wish the system to ignore detection and go straight to the announcement, set `disableAMD`.

The system applies the MOU for this conference to the user.

URL

`/autoconference`

Request

<code>userId</code>	The numeric conference requester's user ID. The system assigns the ID when you create the user. This is required.
<code>phoneNumber</code>	The conference telephone number. This is a phone type.
<code>pin</code>	The conference's PIN number. This is a DMTF type. The system outpulses these digits as DTMF (touch tones) to the conference engine. Valid digits are 0-9, # (pound sign), and * (asterisk).
<code>origination</code>	<p>A <code>type:value</code> pair specifying an origination. You can specify any of the following combinations:</p> <p>1: <code>userId</code>. 2: <code>email_address</code> 3: <code>phone_number</code></p> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
<code>conferenceTime</code>	The time to contact the conference participant. Specify this value as UNIX

time in GMT. Set to 0 (zero) for no expiration.

text The announcement text to play to the call termination point. This is a string.

disableAMD Sets answering machine detection on or off.

Response

HTTP Response Code:	201 ("Created") [when scheduled for immediate creation] 202 ("Accepted") [when scheduled for the future]
Body:	<?xml version="1.0" ?> <ResponseData> <callId>752</callId> </ResponseData>

Errors

301 Invalid Termination Type supplied

302 Invalid Origination Type supplied

304 Internal error: SQL execution failed

autoconference GET

Retrieves the set of conference parameters associated with an auto conference and a particular `userId` or a particular `callId`. The `userId` identifies the user who joined the conference. You must specify the IntelePeer assigned `callId`.

URL

`/autoconference`
`/autoconference/callId`

Request

The system requires all of the following values:

`userId` The numeric conference requester’s user ID. The system assigns the ID when you create the user. This is required.

Response for All

None found:	
HTTP Response Code:	204 No Content
Body:	empty
1 or More Found	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <callIds> <callId>1031</callId> </callIds> </ResponseData></pre>

Response for a Specific Call Identifier

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <conferenceNumber>3035551212</conferenceNumber> <conferencePin>123456#</conferencePin> <conferenceTime>1241133720</conferenceTime> <announcementText>International conference about new marketing plan</announcementText></pre>

	<pre><origination> <type>3</type> <value>7205551212</value> </origination> </ResponseData></pre>
--	--

Errors

- 304 Internal error: SQL execution failed
- 305 Conference does not exist
- 307 Internal error: Corrupted Data

autoconference PUT

Updates a scheduled autoconference. You supply to this call the `callId` returned when you scheduled the call. You also supply the `userId` of the user who requested the conference.

URL

`/autoconference/callId`

Request

- `userId` The numeric conference requester's user ID. The system assigns the ID when you create the user. This is required.

- `callId` The `callId` returned from the `autoconference POST` call.

- `origination` A `type:value` pair specifying an origination. You can specify any of the following combinations:
 `1:userId.`
 `2:email_address`
 `3:phone_number`
 If you specify an email or a `userId`, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.

- `phoneNumber` A required, numeric value teleconference service number.

- `pin` A required, numeric value representing the conference PIN number.

- `text` A required string value specifying an announcement to play to the caller.

- `time` An optional Unix time stamp specifying when the system should place the call. If you do not specify a time, the system uses the current time and places the call immediately.

- `disableAMD` Sets answering machine detection on or off.

Response

HTTP Response Code:	202 ("Accepted")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

- 301 Invalid Termination Type supplied
- 302 Invalid Origination Type supplied
- 304 Internal error: SQL execution failed

autoconference DELETE

Deletes a conference you have scheduled with a POST call. You supply to this call the `callId` returned when you scheduled the call and the `userId` of the user who requested the conference.

URL

`/autoconference/callId`

Request

`sessionId` A string identifier returned from `session GET` request. This is required.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

304 Internal error: SQL execution failed

conference POST

Creates a conference call on behalf of the specified moderator identified by the `userId`. This call establishes the virtual “conference room” and places the moderator into it. The call returns two values, the conference ID and the ID of the first call to the moderator.

The conference is initiated the moment this call is successfully made. Your application must use `PUT` to add participants.

The destination you supply represents the conference moderator. The system attempts to contact the moderator according to the structure of the `destinationType` you specify. The system considers it a success when it reaches a human voice. At that point, the system plays the `text` and prompts the user to press 1 to accept the call and join the conference.

The `duration` is the amount of time the moderator wishes to reserve for the conference. The conference may go exceed its duration however you should ensure that you call to the `conference PUT` method to reserve the conference room’s time. The system applies the conference’s duration to the moderator’s MOU.

URL

`/conference`

Request

<code>userId</code>	The moderator’s user ID. The system assigns this value when you call <code>user POST</code> . This is required.
<code>duration</code>	A required value that specifies how many seconds to reserve the conference room.
<code>destinationType</code>	A required, numeric specifying the type of recipient. Valid values are 1, 2, or 3. The following lists the valid types and what <code>destinationValue</code> you must supply for them. <ul style="list-style-type: none"> 1 an IntelPeer <code>userId</code>. 2 an email address 3 a phone number
<code>destinationValue</code>	A required value defining the destination of the first conference call. This value depends on the <code>destinationType</code> . For example, if you specified a <code>destinationType</code> of 2, you must specify a valid email address. If you specify an email or a <code>userId</code> , the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.
<code>text</code>	A required string value specifying an announcement to play to the caller before

giving the caller the option to connect to the conference. The system converts this text to speech.

Response

HTTP Response Code:	201 ("Created")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <conferenceid>153</conferenceid> <callid>905</callid> </ResponseData></pre>

Errors

301	Invalid Termination Type supplied
303	Blacklisted origination
304	Internal error: SQL execution failed

conference PUT

Adds one or more users to a conference on demand and, alternatively, extends the length of a conference. You must specify which conference using the `conferenceId` value.

The system calls each user and plays the `text`. After pressing 1 to accept the conference, the user is added into the conference. The system returns a `callId` for each leg you add.

URL

`/conference/conferencId`

Request

<code>userId</code>	The moderator's user ID. The system assigns this value when you call <code>user POST</code> .
<code>destinationType</code>	A required, numeric specifying the type of recipient. Valid values are 1, 2, or 3. The following lists the valid types and what <code>destinationValue</code> you must supply for them. <ul style="list-style-type: none"> 1 an IntelPeer <code>userId</code>. 2 an email address 3 a phone number
<code>destinationValue</code>	A required value defining the destination of the first conference call. This value depends on the <code>destinationType</code> . For example, if you specified a <code>destinationType</code> of 2, you must specify a valid email address. If you specify an email or a <code>userId</code> , the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.
<code>text</code>	A required string value specifying an announcement to play to the caller before giving the caller the option to connect to the conference. The system converts this text to speech.
<code>duration</code>	A required value that specifies how many seconds to reserve the conference room.

Response

HTTP Response Code:	201 ("Created")
Body:	<?xml version="1.0" ?> <ResponseData>

	<pre><callid>746</callid> <callid>907</callid> <status>conference extended by 120</status> [verbose only] </ResponseData></pre>
--	---

Errors

301	Invalid Termination Type supplied
303	Blacklisted origination
304	Internal error: SQL execution failed
305	Conference does not exist
306	Conference cannot not be extended

blastvoice POST

Sends a message tone or more to recipients at the specified `blastTime`. Typically, this method is used for reminders. You can send a text blast or an audio file blast. You specify the recipients by passing in one or more pairs of `destination` values.

If you specify text, the system converts it to voice and presents it to the destination side of the call. If you specify an audio blast, you can supply either a `audioFile` or an audio clip on the Internet via a `url`. The file or the URL must be either a MPEQ Audio Stream (`.mp3`) or Windows Media Audio (`.wma`) format. The audio you want to blast cannot exceed 5 MB in size.

The `userId` specifies which user originates the blast while the `origination` pair specifies the origination information for that user. When `allowCallback` is true, recipients may have the option to call back to the originator by pressing 1 (one).

By default, the system presents the blast originator's number to blast recipients. Set the `hideANI` value to true to hide the originator's number.

URL

`/blastvoice`

Request

<code>userId</code>	The ID of the user initiating the blast. The system generates this value when you create the user.
<code>origination</code>	<p>A <code>type:value</code> pair specifying an origination. You can specify any of the following combinations:</p> <ul style="list-style-type: none"> 1: <code>userId</code>. 2: <code>email_address</code> 3: <code>phone_number</code> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
<code>destination</code>	<p>A <code>type:value</code> pair specifying a destination. You can specify any of the following combinations:</p> <ul style="list-style-type: none"> 1: <code>userId</code>. 2: <code>email_address</code> 3: <code>phone_number</code> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>

- text** A required string value specifying a blast announcement to play to the caller. The system converts this text to speech.

- audioFile** An optional string specifying the full path to a `.mp3` or `.wma` file to upload. The file cannot be larger than 5MB. The system stores each file to a unique filename in the Intelpeer network. This allows 10 different users to upload 10 different `hello.mp3` files and still have each file have a unique name.

- url** An optional string specifying the URL of an `.mp3` or `.wma` file to play. The file cannot be larger than 5 MB.

- time** An optional Unix time stamp specifying when the system should place the call. If you do not specify a time, the system uses the current time and places the call immediately.

- hideAni** A Boolean value specifying whether the caller's number from the recipient's caller ID. By default, this false and the system reveals the number.

- allowCallback** An optional numeric value specifying if the recipient can call back to the blast originator. By default, this is 0 (true) and the system allows the recipient to call back. Specify 1 (false) to prevent call backs.

Response

HTTP Response Code:	201 ("Created") [when scheduled for immediate creation] 202 ("Accepted") [when scheduled for the future]
Body:	<pre><?xml version="1.0" ?> <ResponseData> <blastid>123</blastid> </ResponseData></pre>

Errors

- 301 Invalid Termination Type supplied

- 302 Invalid Origination Type supplied

- 304 Internal error: SQL execution failed

blastvoice PUT

Updates a scheduled blast voice call. To change an existing value on the blast voice call, specify a replacement value with this request. Only specify the origination and destination values if you want to change them.

The new `destination` or `origination` value overwrites the old value. This means, for example, if your POST method sent the blast to 5 destinations, and you need to change a single destination in the set, you must resend all 5; the corrected value and the remaining 4.

URL

`/blastVoice/blastId`

Request

<code>userId</code>	The ID of the user initiating the blast. The system generates this value when you create the user.
<code>origination</code>	<p>A <code>type:value</code> pair specifying an origination. You can specify any of the following combinations:</p> <ul style="list-style-type: none">1: <code>userId</code>.2: <code>email_address</code>3: <code>phone_number</code> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
<code>destination</code>	<p>A <code>type:value</code> pair specifying a destination. You can specify any of the following combinations:</p> <ul style="list-style-type: none">1: <code>userId</code>.2: <code>email_address</code>3: <code>phone_number</code> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
<code>text</code>	A required string value specifying a blast announcement to play to the caller. The system converts this text to speech.
<code>audioFile</code>	An optional string specifying the full path to a <code>.mp3</code> or <code>.wma</code> file to upload. The file cannot be larger than 5MB.
<code>0</code>	An optional string specifying a 0 (zero) value this delete the previously

specified `audioFile` or `url` for an audio blast. You need not specify a replacement.

<code>url</code>	An optional string specifying the URL of an <code>.mp3</code> or <code>.wma</code> file to play. The file cannot be larger than 5 MB.
<code>time</code>	An optional Unix time stamp specifying when the system should place the call. If you do not specify a time, the system uses the current time and places the call immediately.
<code>hideAni</code>	A Boolean value specifying whether the caller's number from the recipient's caller ID. By default, this false and the system reveals the number.
<code>allowCallback</code>	An optional numeric value specifying if the recipient can call back to the blast originator. By default, this is 0 (true) and the system allows the recipient to call back. Specify 1 (false) to prevent call backs.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

304	Internal error: SQL execution failed
307	Internal error: Corrupted Data
308	Blast not found

blastvoice GET

Gets the details associated for a scheduled blast identified by the `blastId` parameter.

URL

```
/blastVoice
/blastVoice/blastId
```

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`. This is required.

Response for List Request

None found	
HTTP Response Code:	204 No Content
Body:	empty
1 or more found	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <blastIds> <blastId>1031</blastId> </blastIds> </ResponseData></pre>

Response for blastId Request

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <allowCallback>0</allowCallback> <hideAni>0</hideAni> <blastTime>1241133720</blastTime> <announcementText>testing tts preview for</announcementText> <audiofile> <url>http://www.intelepeer.com/soundpreview/6841351.mp3</url> <name>filename.mp3</name></pre>

	<pre> </audiofile> <origination> <type>3</type> <value>7202108298</value> </origination> <terminations> <termination> <type>1</type> <value>227</value> </termination> </terminations> </ResponseData> </pre>
--	---

Errors

- 304 Internal error: SQL execution failed
- 307 Internal error: Corrupted Data
- 308 Blast not found

See Also

The methods **Errors**

- 301 Invalid Termination Type supplied
- 303 Blacklisted origination
- 304 Internal error: SQL execution failed
- 305 Conference does not exist
- 306 Conference cannot not be extended

blastvoice POST on page 87 and blastvoice DELETE on page 94.

blastvoice DELETE

Deletes a scheduled blast voice with the specified `blastId`. You must supply the `userId` of the call originator.

URL

`/blastVoice/blastId`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

304 Internal error: SQL execution failed

blastsms POST

Sends a message tone or more to recipients at the specified `blastTime`. Typically, this method is used for reminders. You can send a text blast or an audio file blast. You specify the recipients by passing in one or more pairs of `destination` values.

The `userId` specifies which user originates the blast while the `origination` pair specifies the origination information for that user. When `allowCallback` is true, recipients may have the option to call back to the originator by pressing 1 (one).

By default, the system presents the blast originator's number to blast recipients. Set the `hideANI` value to true to hide the originator's number.

URL

`/blastsms`

Request

<code>userId</code>	The ID of the user initiating the blast. The system generates this value when you create the user.
<code>origination</code>	<p>A <code>type:value</code> pair specifying an origination. You can specify any of the following combinations:</p> <ul style="list-style-type: none"> 1: <code>userId</code>. 2: <code>email_address</code> 3: <code>phone_number</code> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
<code>destination</code>	<p>A <code>type:value</code> pair specifying a destination. You can specify one or more. You can specify any of the following combinations:</p> <ul style="list-style-type: none"> 1: <code>userId</code>. 2: <code>email_address</code> 3: <code>phone_number</code> <p>If you specify an email or a <code>userId</code>, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
<code>text</code>	A required string value specifying a blast announcement to play to the caller. The system converts this text to speech.
<code>time</code>	An optional Unix time stamp specifying when the system should place the call. If you do not specify a time, the system uses the current time and

places the call immediately.

Response

HTTP Response Code:	201 ("Created") [when scheduled for immediate creation] 202 ("Accepted") [when scheduled for the future]
Body:	<pre><?xml version="1.0" ?> <ResponseData> <blastid>123</blastid> </ResponseData></pre>

Errors

301	Invalid Termination Type supplied
302	Invalid Origination Type supplied
304	Internal error: SQL execution failed

blastsms PUT

Updates a scheduled blast voice call. To change an existing value on the blast voice call, specify a replacement value with this request. Only specify the origination and destination values if you want to change them.

The new **destination** or **origination** value overwrites the old value. This means, for example, if your POST method sent the blast to 5 destinations, and you need to change a single destination in the set, you must resend all 5; the corrected value and the remaining 4.

URL

`/blastsms/blastId`

Request

userId	The ID of the user initiating the blast. The system generates this value when you create the user.
origination	<p>A type:value pair specifying an origination. You can specify any of the following combinations:</p> <ul style="list-style-type: none">1: userId.2: email_address3: phone_number <p>If you specify an email or a userId, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
destination	<p>A type:value pair specifying a destination. You can specify one or more. You can specify any of the following combinations:</p> <ul style="list-style-type: none">1: userId.2: email_address3: phone_number <p>If you specify an email or a userId, the system resolves these to a user account. Then, it uses the call preferences on the user to identify the proper number to call.</p>
text	A required string value specifying a blast announcement to play to the caller. The system converts this text to speech.
time	An optional Unix time stamp specifying when the system should place the call. If you do not specify a time, the system uses the current time and places the call immediately.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

304	Internal error: SQL execution failed
307	Internal error: Corrupted Data
308	Blast not found

blastsms GET

Gets the details associated for a scheduled blast identified by the `blastId` parameter.

URL

```
/blastsms
/blastsms/blastId
```

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`. This is required.

Response for List Request

None found	
HTTP Response Code:	204 No Content
Body:	empty
1 or more found	
HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <blastIds> <blastId>1031</blastId> </blastIds> </ResponseData></pre>

Response for blastId Request

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <blastTime>1241133720</blastTime> <announcementText>testing tts preview for</announcementText> <origination> <type>3</type> <value>7202108298</value> </origination> <terminations> <termination></pre>

	<pre><type>1</type> <value>227</value> </termination> </terminations> </ResponseData></pre>
--	---

Errors

- 304 Internal error: SQL execution failed
- 307 Internal error: Corrupted Data
- 308 Blast not found

blastsms DELETE

Deletes a scheduled blast voice with the specified `blastId`. You must supply the `userId` of the call originator.

URL

`/blastsms/blastId`

Request

`userId` The user's numeric ID. The system assigns this value when you call `user POST`.

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <status>ok</status> </ResponseData></pre>

Errors

304 Internal error: SQL execution failed

ttspreview POST

Sends text to the server where it is converted to speech. This call returns a URL that you can use to preview how the text will sound when read over the phone.

URL

`/ttspreview`

Request

userId The user's numeric ID. The system assigns this value when you call **user POST**.

text A required String specifying the text to convert.

Response

HTTP Response Code:	201 ("Created")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <url>http://10.32.101.50/sounds/ttsPreview_1_221_87.wav</url> </ResponseData></pre>

uidbyemail GET

Gets the user ID associated with an email address.

URL

`/uidbyemail/email`

Response

HTTP Response Code:	200 ("OK")
Body:	<pre><?xml version="1.0" ?> <ResponseData> <id>66</id> </ResponseData></pre>

Errors

105 Internal error: SQL execution failed