

TringMe Application Programming Interface (API) Specifications

Version 3.2

Last Update: July 2, 2010

Notice: Use of TringMe API is subjected to API Terms of Service available from TringMe's website. TringMe reserves the right to update and change the Terms of Service from time to time without notice.

Table 1 Revision History

Date	Version	Changes
Jan 25 th , 2010	3.0 Draft	<p>This is a major release from the previous version and quite a few features have been added to the platform. While we suggest you to go through the complete API documentation to better understand the additions, here are some of the highlights of the changes that have been made to the existing ones:</p> <ul style="list-style-type: none"> • API URL (BaseURL) will be changed from http://login.tringme.com/api.php to http://api.tringme.com/api.php • Some API operations have been renamed. For e.g. <ul style="list-style-type: none"> ○ 'click2call' and 'sendsms' operations have been renamed to 'call' and 'sms' respectively. ○ 'resetauthphone' and 'confirmauthphone' is replaced by 'setcallerid' and 'verifycallerid' respectively. • Simplified time management – i.e. all APIs that returned absolute time (e.g. 'getcdr', 'getvoicemail' etc) will now return relative time (in seconds since invocation of the API). • 'gettempcookie' and 'connectto' capability is now being replaced by a much powerful concept of channel. • To get real-time notifications, URL set by 'setnotifyurl' API will be invoked. • 'getvoicemail' and 'getcdr' now returns the data in JSON format for ease of consumption. • Deprecated APIs: <ul style="list-style-type: none"> ○ adddid ○ reversetransaction ○ gettempcookie ○ deleteallvoicemails
Feb 1 st , 2010	3.0	<ul style="list-style-type: none"> • Updated Result codes • Description of 'flag' in 'setchannel' API • Added Description of 'getfile' API • A few minor fixes.
Feb 6 th , 2010	3.1	<ul style="list-style-type: none"> • Added section on how to use conferencing APIs • Added bulk SMS support
June 30 th , 2010	3.2	<ul style="list-style-type: none"> • Added support for the multiple callerids.

Table of Contents

1	Introduction.....	6
1.1	TringMe API Type	7
1.1.1	REST APIs	7
1.1.2	Adobe Flash APIs	7
1.1.3	VoicePHP APIs	7
2	Basic Concepts.....	9
3	REST API.....	10
3.1	SSL Support.....	10
3.2	API Return Value	10
3.3	User Management APIs.....	10
3.3.1	Creating a User	11
3.3.2	Activate User	11
3.3.3	Reactivate User	12
3.3.4	Login User	12
3.3.5	Logout User	13
3.3.6	Get User Information	13
3.3.7	Reset Password	13
3.3.8	Change Password	13
3.4	Telephony, Messaging and Conferencing APIs	14
3.4.1	Placing a call	14
3.4.2	Send SMS.....	16
3.4.3	Cancel a scheduled Call /SMS / Conference.....	16
3.4.4	Get Voicemails.....	17
3.4.5	Delete Voicemails.....	17
3.4.6	Create a Conference.....	17
3.4.7	Adding Third Party SIP Server.....	18
3.4.8	Deleting a SIP Server	18
3.4.9	Set CallerID	19
3.4.10	Verify CallerID.....	19
3.4.11	DeleteCallerID.....	20

3.5	Billing, Charging and Call Records APIs	20
3.5.1	Retrieving Call Records.....	20
3.5.2	Retrieving Files (Recordings, Voicemails etc.).....	21
3.5.3	Notification URL	21
3.5.4	Setting Phone Credits In User Account	21
3.5.5	Buy SIP Credits.....	22
3.5.6	Getting Phone and SMS Rates.....	22
3.5.7	Getting either Phone or SMS Rates.....	22
3.5.8	API Usage Information	23
3.5.9	API Results	23
3.6	Using PHP To Invoke TringMe API.....	24
3.6.1	Generating Signature	24
3.6.2	Generating API URL	25
3.7	Debugging.....	25
3.8	Destination – Various Ways to Reach Out To the World of Devices!	26
3.9	Real-time Status Notifications.....	27
3.9.1	Call Notification Parameters	27
3.9.2	Notification Security.....	28
3.10	PHP Helper Functions.....	29
4	Adobe Flash API.....	30
4.1	Making Call.....	30
4.2	NetConnection targetURI Specification	30
4.3	Ending Call.....	31
4.4	Call Status Notifications in Flash	32
4.5	Using TringMe’s Premade Flash Widgets.....	33
4.5.1	Embedding TringPhone widget	33
4.5.2	Embedding Push-n-talk widget	33
5	Channels.....	35
5.1	Channel Creation.....	36
5.2	Channels APIs.....	36
5.2.1	Create or Modify a Channel	36
5.2.2	Get Channel Information.....	37

5.2.3	Delete Channel	38
5.3	Accessing Channels using SIP	38
5.3.1	Mapping Channel to DID (Direct Inward Dialing)	38
6	Billing and Charging	39
6.1	Simultaneous Calls	39
6.2	Conference Charges	39
7	How To	40
7.1	Create a Conference	40
7.2	Best Practices	40
7.2.1	Calculating available credits	40
8	Support	41

1 Introduction

TringMe unified communication platform empowers developers to create powerful voice & messaging applications without spending time & fortune in doing so. Developers can create voice & messaging applications, whether it's peer-to-peer, voice broadcast, conferencing or IVR in just a matter of few hours. Best part is that it's a 'unified' communication platform which allows application to communicate not only over phone but also via Browser(web), Adobe Flash, Instant Messenger OR Mobile devices (Blackberry, Nokia & iPhone).

TringMe was the first company in the world to demonstrate the commercial (Adobe) Flash Telephony. It enabled one to make calls directly from a browser without requiring downloading or installing anything – true web-based telephony! . For VoiceXML developers or anyone else who is looking to develop interactive voice applications, TringMe VoicePHP platform, a VoiceXML replacement platform hailed by industry stalwarts is the ideal solution.

The flexibility of TringMe platform makes it suitable for range of application only limited by your imagination. The platform has evolved to meet all the needs of telephony, messaging and conferencing and the APIs have designed to make it easy for developers to deploy the applications quickly and reliably. Take a look at some of the areas where TringMe's capabilities can be used. For example,

- Social networking, dating sites can easily integrate voice as a means of communication in innovative ways
- VoIP providers can quickly deploy and offer their services using TringMe's easy to use Flash based web-phone, widgets or mobile devices enabled via cheap to use Mobile-VoIP application.
- Game developers, virtual world creators can enable making phone calls from virtual world to real world and vice-versa.
- CRM, Sales Management tool can integrate complete call-control, messaging and conferencing solution.
- Ad-hoc conferencing, Enterprise conferencing solutions.
- IVR – VoicePHP.

Advantage

TringMe API is very powerful, yet simple and easy to understand. Take a look at some of the key advantages of TringMe:

1. No Telco or VoIP expertise required to create voice application. No need to code in those complex and specialized languages meant for voice programming only - like VoiceXML. Create voice and messaging applications using any language of your choice - Adobe Flash, Flex, AIR, PHP, AJAX, Silverlight or any technology or platform.
2. The platform is written from scratch to cater to the emerging needs of the internet, mobile devices and their diverse usage – in real world or virtual. Applications can be written
 - a. Web
 - b. Mobile Devices
 - c. Desktop Devices

- d. Any connected device, handheld, netbook etc.

The best part is that these all devices can interact with each other over calls and SMS.

3. TringMe has abstracted the inner details of call signaling and call processing by exposing easy to use, standards based APIs for developers to essentially innovate at the application level
4. TringMe provides a highly scalable carrier grade platform with built-in load balancing, optimized call routing, redundancy and automatic switch-failover capabilities. There is nothing special one has to do to avail these features./ switch
5. TringMe's platform handles over a million calls per day and hence you can be assured that it is built to handle your application's telephony and messaging needs.
6. TringMe's platform provides complete support of TTS (Text-to-speech), Speech recognition, DTMF etc.
7. Support for auxillary functions like user management, billing and call records is also built-in to the platform.

1.1 TringMe API Type

At a high level, TringMe's platform offers three type of APIs:

1.1.1 REST APIs

This is the most basic API. TringMe REST API calls are made over the internet by sending HTTP GET or POST requests to our server. For example, to get the TringMe user id, following URL should be constructed.

<http://api.tringme.com/api.php?apikey=12sdas8dadadsadajs3&sig=4b61af9cd869706edc6a4569fc4e7b33&op=getuid&email=api@anexample.com>

TringMe's REST API is covered in Section 0 below.

1.1.2 Adobe Flash APIs

This enables developers to quickly create their own Adobe Flash, Flex or AIR application based to utilize TringMe's standards based APIs for telephony and messaging. Given the powerful functionality of Adobe Flash and the ease of use of TringMe's APIs, developers will be able to produce compelling voice enabled apps using Adobe's – Flash, Flex or AIR – technologies. For example, developers can create a fully-functional desktop or web-based phone on top of TringMe APIs using Adobe AIR technology. TringMe's Flash API is covered in Section 0 below.

1.1.3 VoicePHP APIs

In simple terms, VoicePHP is the same old PHP which now enables you to create voice applications. It's not an extension to PHP; infact it's the same PHP which now outputs voice instead of text and also takes input as voice instead of text. In technical terms, its PHP whose standard text based input & output (stdio, stdout) are replaced by voice equivalent. Please visit <http://voicephp.com> to know more

about VoicePHP API. Also take a look at <http://code.voicephp.com> to sample some of the types of IVR applications that can be built using VoicePHP.

Welcome to the world of voice programming, the way it has never been done before!

2 Basic Concepts

Telephony and Messaging is not new to us; we use it every day in our life. Infact it has become an integral part of our day-to-day communication Whether it is just a phone call, a conference we attend, interact with an automated bank assistant via IVR, voice chat with our buddy half-way around the world via instant messenger or just send an SMS, it is clear that communication has only increased and the types of devices we use to interact have grown as well.

Hence, with evolving trends, we are not restricted to phone to converse anymore. One could be using phone to call another user on phone or SIP or say Gtalk and vice versa. Apart from the peer to peer calls, we often dial to automated voice response system (say phone banking) or multi-party destination (say conference). This gives a plenty of combinations how one calls and how one receives the call.

TringMe's APIs allows you to communicate with all of the above mentioned source an destination types. Before we get deeper into this, let's define some terminology which will be used throughout this document.

Source: Who is initiating call and how. Source could be placing the call from phone, Flash widget or Flash phone, SIP, Instant Messenger, or a mobile device using VoIP.

Destination: This identifies to whom the call is initiated to. It can be an individual, a multi-party conferencing or an IVR (VoicePHP) application. The individuals can receive the call on wide range of devices that TringMe's unified communication platform offers and supports - Phone, Instant Messenger (Gtalk, Skype), SIP, etc. Take a look at the "Supported Destination" section below to understand the specifics of how various types of destination can be specified.

As you go through the document, it will become clear as to how various combinations of source and destination can be used by your application. However to give you a quick glimpse of how simple to use source & destination, invoking following URL will place call between two phone numbers 18585551234 & 15015551234

<http://api.tringme.com/api.php?apikey=12sdas8dadadsadajs3&sig=4b61af9cd869706edc6a4569fc4e7b33&op=call&src=18585551234&dest=15015551234>

Simple isn't? In later sections, we will show more powerful examples like how to play greetings, defining destinations for recurring use (channels), scheduling calls and conferences etc.

Without further ado, let's dive deeper into the APIs now.

3 REST API

TringMe REST API calls are made over the internet by sending HTTP GET or POST requests to our server. For example, to get the TringMe user id, following URL should be constructed.

<http://api.tringme.com/api.php?apikey=12sdas8dadadsadajs3&sig=4b61af9cd869706edc6a4569fc4e7b33&op=getuid&email=api@anexample.com>

You will require API keys in order to use TringMe REST API. To obtain API keys, send your request to a special email address support@tringme.com along with your TringMe user ID. API keys are available in a pair, public and private keys.

To access the TringMe API, you should pass your public API key, operation code and signature of the request to the TringMe's API server. Depending on requested operation, you will also need to pass one or more required parameters. All API request must include checksum signature using private key to avoid any tempering of the request. Note that, you **MUST NOT** include your private key for any transaction. Private key is only used to generate signature as described in section **Error! Reference source not found.**

3.1 SSL Support

TringMe supports strong encryption. TringMe API is also accessible over https connection using the same URL (<https://api.tringme.com/api.php>).

3.2 API Return Value

Unless otherwise specified, all the APIs return following:

- If successful, it will return 'OK' or 'OK <result>'.
- On failure, it will return 'FAIL <reason>'. For explanation of reason code, please refer to the **API Results** section below.
- TringMe API only allows clients to make a limited number of calls in a given hour. For most **usages, our** limits are sufficient. It is best practice for applications to use local database and invoke TringMe API only when needed. TringMe dynamically throttles API requests if necessary, especially when invoked without real usage or calls.

3.3 User Management APIs

User Management enables you to create users and assign them various capabilities. For e.g. provide them phone credits, configure SIP settings for them, restrict their calls to only certain destinations etc. Although, it's not necessary to create users in order to use the TringMe APIs, creating users can give you an additional flexibility, for example restricting usage per user, per user call records etc. You can channel your entire user's need (e.g. call or sms) though TringMe's user management API and keep yourself free from implementing user management at your local end.

TringMe's users can be categorized into two categories:

Public Users: Public users are those which can be accessed by any API holder. Public users can even login directly at <http://tringme.com>:

1. Can directly login at TringMe's website.
2. Is not restricted to the developer who has created this user using the API.
3. TringMe can reach out to these users since they are not private to the developer who created them.

Private Users: These users are exclusively created in a 'private' space for the developer whose API keys were used to create them. These users are also created using 'createuser' API but with parameter 'private' set. Private users are more flexible if you need:

1. To create TringMe users who are restricted only to your application. No API holder can even access or change any settings of the private users.
2. Better user management capabilities like changing the password, reversing the funds etc.
3. Recommended

Let's go through user management APIs to better understand the capabilities.

3.3.1 Creating a User

This API is used for creating a user. It is the fundamental API to the user management capabilities in TringMe.

3.3.1.1 Parameters

op	'createuser'
email	Email address of the user being created.
password[optional]	Password to be used for this account. Password is optional for the public users. In that case, system will generate the password and send in welcome email.
name [optional]	Name of the user.
private [optional]	Create a private user. If this parameter is specified, user will be created in API user's private space. No emails will be sent by TringMe for any private users

3.3.1.2 Example

`http://api.tringme.com/api.php?op=createuser&email=joe.dirt%40somedomain.com&name=Joe+Dirt&password=joedirt&apikey=b12345678901234567890123456789012345&sig=f1234567890123456789012345678901`

3.3.2 Activate User

This API is used for activating a public user after user creation. This is an essential step before the user can login. This API is NOT needed for private users since private users are pre-activated.

3.3.2.1 Parameters

op	'activateuser'
email	Email address of the user being activated.
activationcode	Activation code which was sent to the user on user's email address which was passed in 'createuser' API. This is required only for public users. Private users are automatically activated on creation.

3.3.3 Reactivate User

This API is used for re-activating an existing public user. It resends the activation email to the user. This is only applicable if the user was created using 'createuser' API and password was specified during user creation. This API is NOT needed for private users.

3.3.3.1 Parameters

op	'reactivateuser'
email	Email address of the user.

3.3.4 Login User

This API is used to login a user. One user logs in, system will return the cookie which can be used in all subsequent API calls to associate the user with that API operation (e.g. call)

3.3.4.1 Parameters

op	'login'
email	Email address of the user.
password	Password of the user.
validity [optional]	Validity of login in minutes. Default, 90 minutes.
private	Private User Login. This parameter should only be set if the user was created - 'createuser' - with the private parameter set.

3.3.4.2 Return Value

If successful, it will return 'OK <cookie>'. The cookie will be used for all the API operations later to identify the user.



Some of the APIs take an optional 'cookie' parameter (e.g. 'call'). If 'cookie' is omitted when invoking such APIs, the API holder's account is used (i.e. if 'call' API is invoked without specifying cookie, then the call will be placed from the API holder's account thereby charging the API holder). If 'cookie' is provided, the API uses the user's account whose credentials were passed to obtain 'cookie' - using 'login' (i.e. in the case of 'call', the call will be charged to the user's account and not API holder).

3.3.4.3 Example

`http://api.tringme.com/api.php?op=loginuser&email=joe.dirt%40somedomain.com&password=joedit&apikey=b12345678901234567890123456789012345&sig=f1234567890123456789012345678901`

3.3.5 Logout User

This API is used to logout a user from the system.

3.3.5.1 Parameters

op	'logout'
cookie	Cookie which was obtained using 'login'.

3.3.6 Get User Information

This API is used to get user information.

3.3.6.1 Parameters

op	'getuser'
cookie	Cookie which was obtained using 'login'.

3.3.6.2 Return Value

If successful, it will return 'OKJ' with user information in JSON format.

3.3.7 Reset Password

This API is used to change a user password. An email will be sent to user with code that should be used to change the password in 'changepassword' API.

This API is only needed for public users. Use 'changepassword' API for private users.

3.3.7.1 Parameters

op	'resetpassword'
email	Email of the user where the reset password email should be sent.
subject	Subject of the email.
message	Message that should be sent to the user. Note that you can include %TRINGMEAUTHCODE% as a placeholder in your message where TringMe will insert the verification code. This verification code should be used in 'changepassword'.

3.3.8 Change Password

This API is used to change a user password. For public users, change password requires either the old password or reset code sent by 'resetpassword' API. For private users, password can directly be changed by giving new password.

3.3.8.1 Parameters

op	'changepassword'
email	Email of the user.
password	New password.
oldpassword [optional]	Old password of the user. Optional for private users. If this is provided then, 'code' parameter is not required.
code [optional]	A verification code that was sent by 'resetpassword'. This parameters is mandatory if 'oldpassword' is not being supplied.

3.4 Telephony, Messaging and Conferencing APIs

3.4.1 Placing a call

TringMe provides multiple ways to place calls – using REST API, using Flash, SIP, Instant messenger (e.g. Google-Talk), Mobile VoIP etc. This particular API will give you all the information needed to place a call using REST.

Fundamentally speaking, this will create a callback between two parties. The system will call 'src' first and then 'dest' and then bridge both the calls. By default, the 'src' will receive the call with the caller-id of the invoker (as set by the 'setcallerid' API). The 'dest' will receive the 'src' number as the caller-id. The caller-id behavior can be modified by setting the private caller-id flag. In private caller-id case, both 'src' and 'dest' will receive the caller id of the invoker thereby enabling the communication between two parties without revealing each other's number. The API is extremely powerful and supports a lot of combinations (for e.g. broadcasting a message to multiple phone numbers, playing greeting messages, taking user input and so on).

3.4.1.1 Parameters

op	'call'
cookie [optional]	Cookie which was obtained using 'login'.
src	First party phone number. Call will be first placed to this number. If the call is not answered, the API will terminate.

You can also specify comma separated multiple phone numbers to make bulk calls. This feature is specifically useful when broadcasting a voice message or inviting multiple people to a conference (e.g. 18585551212,19125551212). If you specify multiple phone numbers, then 'dest' should not be specified (Read more about 'dest' below).

dest [optional]	<p>Second party phone number. To know about all the supported types of destination, please refer to section 3.8.</p> <p>Destination is optional if 'greeting' parameter is specified. This will enable call (the 'src') and play a message functionality.</p>
callerid [optional]	<p>Caller-ID to be used (set using 'setcallerid' API).</p>
tag [optional]	<p>User supplied parameter to associate a call in the Call Data Records.</p> <p>This should be alphanumeric only (a-z, A-Z, 0-9). All other characters will be replaced by 'X'. It can be a maximum of 31 characters.</p> <p>If you wish to be notified via a notify URL ('setnotifyurl'), please ensure that tag is specified. Otherwise the notify URL will not be invoked. This is to ensure that you can identify the call with the tag.</p>
duration [optional]	<p>Duration after which the call should automatically disconnect.</p> <p>The duration is the timeframe from the moment 'src' is answered till the time when the call disconnects. (Note, also see Section 3.5.8.3)</p>
greeting [optional]	<p>Greeting that should be played immediately after the 'src' answers. It can be either:</p> <ul style="list-style-type: none"> • URL of the wave file. Check supported formats below. • A text string that should be spoken. It needs to be prefixed with "male:" or "female:" to specify the type of voice to be used. For e.g. ("male>Welcome to XYZ Bank")
postcallgreeting [optional]	<p>Greeting that should be played before disconnecting the call. It can be either:</p> <ul style="list-style-type: none"> • URL of the wave file. Check supported formats below. • A text string that should be spoken. It needs to be prefixed with "male:" or "female:" to specify the type of voice to be used. For e.g. ("male:Thank you for using XYZ Bank")
dtmfmaxdigits [optional]	<p>Maximum number of DTMF digits expected to enter by user after playing a greeting. The default value for this option is 0.</p>
dtmfretries [optional]	<p>Number of times DTMF entry will be retried. The default value is 1. This is valid only if 'dtmfmaxdigits' > 0 and 'greeting' is specified.</p>
dtmftimeout [optional]	<p>DTMF timeout in seconds. The default value is 10 seconds. This is valid only if 'dtmfmaxdigits' > 0 and 'greeting' is specified.</p>
dtmftimeoutgreeting [optional]	<p>Greeting that should be played if DTMF timeout is reached. It can be either:</p> <ul style="list-style-type: none"> • URL of the wave file. Check supported formats below. • A text string that should be spoken. It needs to be prefixed with "male:" or "female:" to specify the type of voice to be used. For e.g. ("male:You have entered the PIN. Please try again")

flag [optional]	<p>One or more combinations of the flags below:</p> <p>0 – Disables all notifications. 1 – Send call completed and DTMF input notifications. This is the default value. 2 – Send call-in-progress notifications (e.g. SRCANSWERED) 16 – Private Callerid.</p> <p>For more details on real-time notifications, refer to section 3.9.</p>
when [optional]	<p>Time in seconds, after which call should be invoked. This is how calls can be scheduled in the future. Value should be more than 60 seconds.</p>

Supported Formats in Greeting Audio File:

- WAV file - 8KHz sampled, 16 bit linear, mono, uncompressed.

3.4.1.2 Return Value

- If successful, it will return 'OK'
- If call was scheduled in future using 'when' parameter and successful, it will return 'OK <id>'. This id is the call identifier which can be used in 'cancel' if the call needs to be cancelled.

3.4.2 Send SMS

This API is used to send an SMS to worldwide destinations.

3.4.2.1 Parameters

op	'sms'
cookie [optional]	Cookie which was obtained using 'login'.
callerid [optional]	Caller-ID to be used (set using 'setcallerid' API).
dest	Destination phone numbers where the SMS should be sent. You can also specify comma separated multiple phone numbers to send bulk SMS.
msg	Message to be sent (HTTP URL Encoded)
when [optional]	Time in seconds, after which call should be invoked. This is how SMS can be scheduled to be sent at a later time.

3.4.2.2 Return Value

If successful, it will return 'OK'

3.4.3 Cancel a scheduled Call /SMS / Conference

This API is used to cancel a scheduled call, SMS or conference. This is a premium API and will be available on request only. Note that it's not always possible to cancel the scheduled call, SMS or conferences if there is not a sufficient time to cancel or calls/sms are already placed in queue.

3.4.3.1 Parameters

op	'cancel'
id	Call/SMS/Conference identifier

3.4.4 Get Voicemails

This API is used to retrieve voicemails.

3.4.4.1 Parameters

op	'getvoicemail'
cookie	cookie which was obtained using 'login'.

3.4.4.2 Return Value

If successful, it will return 'OKJ' along with voicemail information in JSON format. The timestamp of the voicemail will be in seconds relative to the time of invocation of the API.

3.4.4.3 Example

http://api.tringme.com/api.php?op=getvoicemail&email=joe.dirt%40somedomain.com&cookie=1234567890123456789012345678901&json=1&apikey=b12345678901234567890123456789012345&sig=f1234567890123456789012345678901

Response:

```
OKJ [{"voicemail": "v1t412345678901234567891", "srcphone": "", "recvtime": "45"}, {"voicemail": "v1t412345678802234567892", "srcphone": "", "recvtime": "138"}]
```

3.4.5 Delete Voicemails

This API is used to delete voicemails.

3.4.5.1 Parameters

op	'deletevoicemail'
cookie	cookie which was obtained using 'login'.
id	Voicemail identifier available in the result obtained using 'getvoicemail'. If 'id' is set to 0, then all voicemails will be deleted.

3.4.6 Create a Conference

This API is used to schedule a conference call.

3.4.6.1 Parameters

op	'createconference'
pin	PIN number that every attendee will be required to specify for authentication purposes before joining the conference call.
apin	PIN number of the administrator. Administrator has special privileges to control the conferencing parameters.
maxusers [optional]	Max number of users that can attend this conference call. The default value is 10.
flag [optional]	Flags
duration [optional]	Duration of this conference call in seconds. The default value is 3600 seconds.
when [optional]	Time in seconds, after which conference should be activated.

3.4.6.2 Return Value

If successful, it will return 'OK <id>'. The <id> specifies the conference identifier. All members that are attending this conference will be required to know this id.

3.4.7 Adding Third Party SIP Server

This API is used to specify a SIP server that you may want to use for call routing. You can then route all the calls through this SIP server.

3.4.7.1 Parameters

op	'setsipserver'
cookie [optional]	Cookie which was obtained using 'login'.
id [optional]	Identifier of the SIP server
host	URL of the SIP proxy server that you wish to use.
username	Username to be used at the specified 'host'.
password	Password to be used at the specified 'host'.
auth	SIP auth name to be used at the specified 'host'.

3.4.7.2 Return Value

If successful, it will return 'OK <id>'. The <id> specifies the SIP identifier which can be used in other APIs.

3.4.8 Deleting a SIP Server

This API is used to delete a SIP server which was added using 'setsipserver'.

3.4.8.1 Parameters

op	'deletesipserver'
cookie [optional]	Cookie which was obtained using 'login'.
id	ID obtained using 'setsipserver' API.

3.4.9 Set CallerID

This API is used to set caller-id.

3.4.9.1 Parameters

op	'setcallerid'
cookie	cookie which was obtained using 'login'.
message	Message that should be sent to the user. Note that you can include %TRINGMEAUTHCODE% as a placeholder in your message where TringMe will insert the verification code. This verification code will be required for 'verifycallerid' API to complete the process of setting a verified caller-id. You can insert %TRINGMEAUTHCODE% as many times as you want in the message. For e.g. A typical use case of this could be to repeat the code if a call is placed to the user.
callerid	Phone number to be set as the callerid. To verify the authenticity of callerid, a call or an SMS will be placed with a verification code which can be passed to 'verifycallerid' API subsequently to complete the process. . The choice of using call or SMS for verification can be controlled by the 'flag' parameter below. Please note that SMS or Call charges as applicable will be charged to the user.
flag	One or more combinations of the flags below: Following bit values can be specified: <ul style="list-style-type: none">• Bit 0 – Use SMS (0) or Call (1) for verification. '0' is the default value• Bit 1 – Set this phone number as the default caller-id. This is useful when you have multiple verified caller-ids (default is ON).• Bit 2 – Forced verification. This is useful when you want to force a caller-id verification even if that 'callerid' was already verified earlier.

3.4.10 Verify CallerID

This API is used to verify the caller-id that was being requested to set via 'setcallerid' API.

3.4.10.1 Parameters

op	'verifycallerid'
cookie	cookie which was obtained using 'login'.

code	The verification code that was sent in the SMS or call. The SMS or call was placed in response to 'setcallerid' API invocation.
------	---

3.4.11 DeleteCallerID

This API is used to delete existing caller-id that was being set via 'setcallerid' API.

3.4.11.1 Parameters

op	'deletecallerid'
cookie	cookie which was obtained using 'login'.
callerid	The phone number that should be deleted as the caller-id.

3.5 Billing, Charging and Call Records APIs

3.5.1 Retrieving Call Records

Access to complete billing and call-records is provided by using this API. The call-record includes all the relevant call-details (phone number, call- duration, time, call-tag etc.). It returns the data in JSON format.

3.5.1.1 Parameters

op	'getcdr'
cookie	Cookie which was obtained using 'login'.
id [optional]	All call records greater than specified 'id' will be retrieved.
starttime [optional]	Starting time from which the call records should be retrieved. This should be specified in seconds relative to the time of invocation.
endtime [optional]	Ending time till which the call records should be retrieved. This should be specified in seconds relative to the time of invocation.
tag [optional]	If tag is specified, then call records matching that tag are retrieved. Wildcards (*, ?) are supported to retrieve matching call records as well.

3.5.1.2 Return Value

If successful, it will return 'OKJ' alongwith call records of completed calls in JSON format. The timestamp of the call records will be in seconds relative to the time of invocation of the API. Following are the 'type's of records that will be returned.:

type	Description
0	Call
1	SMS
2	Amount transferred using 'setphonecredits' API
4	SIP Credits Purchase

5	SIP
---	-----

3.5.1.3 Example

http://api.tringme.com/api.php?op=getcdr&cookie= 1234567890123456789012345678901&json=1&apikey=b12345678901234567890123456789012345&sig=f1234567890123456789012345678901

Response:

OKJ

```
[{"id":"12382","phone":"18585551212","duration":"4","charges":"0.0120","type":"0","calltag":"","calltime":"15365"},{"id":"12355","phone":"441234567890","duration":"3","charges":"0.0100","type":"0","calltag":"","calltime":"141335"},{"id":"12345","phone":"911234567890","duration":"0","charges":"0.0400","type":"1","calltag":"","calltime":"152396"}]
```

3.5.2 Retrieving Files (Recordings, Voicemails etc.)

Retrieve file from TringMe infrastructure. The file may be deleted once retrieved and hence should be stored by the invoker. Any file that is newly made available to the system (for e.g. a file recorded using VoicePHP ‘record’ or a newly received voicemail) will be available for download for a maximum period of 72 hours, after which it may be deleted.

3.5.2.1 Parameters

op	‘getfile’
id	File ‘id’ as returned by VoicePHP ‘record’ API or ‘getvoicemail’.

3.5.3 Notification URL

TringMe also provides a notification mechanism whereby developers can listen for notifications to retrieve call details including call-duration, total charges, call-tag and so on. API holder can listen to these notifications and do the appropriate mark-up processing before inserting the call-related information in their database.

3.5.3.1 Parameters

op	‘setnotifyurl’
url	URL where notifications should be sent.
flag [optional]	Reserved

3.5.4 Setting Phone Credits In User Account

This API provides a way for an API account holder to transfer credits to the user’s account.

3.5.4.1 Parameters

op	'setphonecredits'
cookie [optional]	Cookie which was obtained using 'login'.
email [optional]	Email address of the user. This is only required in case of private account and if 'cookie' is omitted.
private [optional]	Private user. This specifies if the user to whose account the credits are being set is private. This will only be used if 'email' has been specified.
amount	Amount in USD that should be transferred. In case of private user, negative amount can be specified. It will debit the amount from the user account and credit it to API holder account.

3.5.4.2 Return Value

If successful, it will return 'OK <transaction number>'. The <transaction number> is only mean for reference purposes.

3.5.5 Buy SIP Credits

This API provides a way buy SIP credits.

3.5.5.1 Parameters

op	'buysipcredits'
cookie	Cookie which was obtained using 'login'.
amount	Amount in USD for which the SIP credits should be set. It deducts the specified amount of USD from phone credits and creates equivalent SIP credits.

3.5.6 Getting Phone and SMS Rates

This API provides a way to get the call and SMS rates for a specific phone number.

3.5.6.1 Parameters

op	'getrates'
dest	Phone number for which rates are required. It can be specified as a partial phone number too (for e.g. 44991 would retrieve rates for UK).

3.5.7 Getting either Phone or SMS Rates

This API provides a way to get the call or SMS rates for a specific phone number.

3.5.7.1 Parameters

op	'getrate'
----	-----------

dest	Phone number for which rates are required. It can be specified as a partial phone number too (for e.g. 44991 would retrieve rates for UK).
type	Type of rate requested. 0 specifies phone rate and 1 specifies SMS rate. The default value is 0.

3.5.8 API Usage Information

3.5.8.1 Mandatory API Parameters

Following sections describe the mandatory API parameters which should be a part of every API request that is sent by the client.

Parameter	Description
'sig'	Signature
'apikey'	Public API Key
'op'	Operation (see operation table)

3.5.8.2 Cookie Usage In APIs

In any API operation, if cookie is omitted, the operation will be performed from API user's account.

3.5.8.3 Duration Usage In APIs

Some APIs take duration as a parameter. Although we strive hard to ensure that duration as specified is honored, it may vary by a few seconds as call-setup and disconnection involves signaling.

3.5.9 API Results

API response is indicated by OK or Fail condition. Success will be indicated by following output:

OK [<value>]

'value' will be operation specific and optional. Please refer to the specific API for how to interpret the value.

Failure will be indicated by one of the values tabulated below:

Result	Description
FAIL	Operation Failed (however API key, signature etc were okay)
NOKEY	Missing Key
NOSIG	Missing Signature
NOOP	Missing Operation
BADSIG	Bad Signature
INVALIDKEY	Invalid API Key
INSUFFICIENTCREDITS	Insufficient Credits
BADDEST	Destination is invalid
MISSINGPARA	Missing parameters
USEREXIST	User already exists
AUTHFAIL	Authorization Fail or Session Expired/Invalid cookie (get new

	cookie)
NOTLOGGEDIN	Login Required
BADDURATION	Duration value is out of limit. Ensure that you are using the right unit (minute vs. second)
BADCHANNEL	Channel does not exist OR not owned by API holder
TOOMANYCHANNELS	Channel quota exceeded
TOOMANYCONFS	Number of conference you can schedule is exceeded

You should ignore any line in response starting with '#'

3.6 Using PHP To Invoke TringMe API

If you are using PHP, TringMe API source includes all the code necessary to call and execute desired API function. All you have to do is to create an array with parameters and call API function.

```
// complete php code to create your own click to call service
// From - http://blog.tringme.com/roll-out-your-own-click-to-call-service/

<?php

$publickey = "yourpublicapikey";
$privatekey = "yourprivateapikey"

include_once ("tringmeapi.php");

;

$parameters=array();
$parameters['op']='call';
$parameters['apikey']=$publickey;
$parameters['src']='18585551111';
$parameters['dest']='18585552222';
$response = TringMeAPI($parameters);
```

If you are not using PHP and like to use TringMe API, you need to generate signature and API URL and use appropriate library (e.g. libcurl) to invoke the API using URL. In above code, all mentioned functionalities are handled by function **TringMeAPI**.

Below is description of how you can generate signature and API URL manually.

3.6.1 Generating Signature

Signature is generated by concentrating all the parameters in HTTP REQUEST (\$_GET or \$_POST field in php) , private key and taking md5 of it. Example php code:

```
// sample php code to compute the signature
```

```

$parameters = array();
$parameters ['op'] = 'login';
$parameters ['apikey'] = $publickey;
$parameters ['email'] = 'api@example.com';
$parameters ['password'] = 'somepw';
$sig = GetSignature($parameters , $privatekey);

function GetSignature($params_array, $private) {
    $str = '';
    ksort($params_array);
    foreach ($params_array as $key=>$val) {
        $str .= "$key=$val";
    }
    $str .= $private;
    return md5($str);
}

```

3.6.2 Generating API URL

Following sample code shows how to create REST API URL. Although sample code is in PHP, you can use any language of choice (ASP, .Net, Perl, C, C#, VB, etc) :

```

// sample php code to generate REST API URL

$parameters = array();
$parameters ['op'] = 'login';
$parameters ['apikey'] = $publickey;
$parameters ['email'] = 'api@example.com';
$parameters ['password'] = 'somepw';
$sig = GetSignature($parameters , $privatekey);

$url = CreateAPIURL($parameters, $sig);

function CreateAPIURL($params_array, $signature) {
    $uri='http://api.tringme.com/api.php?';
    foreach ($params_array as $key=>$val) {
        $uri .= "$key=$val&";
    }
    $uri .= "sig=$signature";
    return $uri;
}

```

3.7 Debugging

An additional parameter 'debug' with value '1' can be passed in all the operations to get dump of the parameters received by TringMe API server. You can access the dump at the following location

<http://api.tringme.com/apidebug/<publickey>.txt>

To delete the debug logs, pass parameter 'debug' with value '2'.

3.8 Destination – Various Ways to Reach Out To the World of Devices!

The complete magic of TringMe's platform is to unify various forms of communication over varying devices. These are all categorized as destinations and are accepted in APIs which take destination as a parameter. Let's look at all the types of destination that are supported and format in which it needs to be specified in API:

Destination Type	API Destination Format
Phone (worldwide mobile or landline)	Valid phone number (prefixed with country code. For e.g. to call US, the phone number should start with a '1' (as in 18585551212 to call a number in San-Diego, US)
SIP	<ul style="list-style-type: none"> • sip:<sip uri> , for example, sip:identity@mysipserver.com • sip:username:password@mysipserver.com • sip:<number>@<sip-id> - to call a <number> using a SIP server that was added using 'setsipserver' API
VoicePHP	<p>voicephp:<script URL></p> <p>Note: Depending on the extension of the script, it will be either downloaded once or every time. For example:</p> <ol style="list-style-type: none"> <i>voicephp:http://myserver.com/script.voicephp</i> – In this case, script will be always downloaded before the execution of VoicePHP application. <i>voicephp:http://myserver.com/script.txt</i> – In this case, the script will be downloaded only once.
Conference	<p>conf:<conf-id>:flag</p> <ol style="list-style-type: none"> <conf-id> is returned by 'createconference' API. Flag is combination of one or more of following: <ul style="list-style-type: none"> • 0 – Ask for PIN • 0x1 – Use User PIN • 0x2 – Use Admin PIN • 0x10 – Listen Only • 0x20 – Quiet Mode
Gtalk	gtalk: user@gmail.com

Voicemail	<i>voicemail</i>
Mobile VoIP	<i>mobilevoip</i>

3.9 Real-time Status Notifications

Live call status notification can be received by specifying the notify URL ('setnotifyurl'). The notification URL will be invoked with following parameters.

3.9.1 Call Notification Parameters

Parameter	Value	Description
'f'	'tm'	This is an identification field and will be populated with the value 'tm'.
'status'	Status	Status codes – could be one of the following: <ul style="list-style-type: none"> • SRCFAILED – Source call leg failed • SRCNOTANSWERED – Source call leg not answering • SRCANSWERED - Source call leg answered. In case of Flash based call, this will be invoked when source call leg gets connected to destination. • SRCCANCELED - Source disconnected the call (before destination was connected) • DESTNOTANSWERED - Destination not answering • DESTBUSY – Destination busy • DESTCONGESTED - Destination could not be connected due to congestion. • DESTNETWORKERROR DESTCONGESTED - Destination could not be connected due to destination network error. • USERDTMFINPUT – DTMF value • COMPLETED – Call completed successfully
'tag'	Call tag	Call tag passed during API invocation. Only for notifications that are generated by the system – i.e. deletechannel and voicemail notifications - 'TRINGME' will be sent as tag. For all other notifications, the user is required

		to pass tag when invoking the API.
'sig'	Signature	Hash Signature
'r'	Random Number	Random Number used for generating the signature
'srcduration'	Source Duration	Call Duration for source leg (in Seconds).
'destduration'	Destination Duration	Call Duration for destination leg (in Seconds).
'srccharge'	Source Charge	Source leg charge in USD cents (if applicable)
'destcharge'	Destination Charge	Destination leg charge in USD cents (if applicable)
'srcsipcharge'	SIP charge for source leg (if SIP is used)	Source leg charge in SIP credits (if applicable)
'destsipcharge'	SIP charge for destination leg (if SIP is used)	Destination leg charge in SIP credits (if applicable)
'p'	Parameters	Parameters specific to the type of Status. For USERDTMPINPUT – it contains the DTMF input digits. No other Status has any parameters.
'type'	Type of Notification	Following types of notifications are sent: <ul style="list-style-type: none"> • callstatus – This notification is sent to notify about changing call status. • cdr – This notification is sent when the call is completed. • voicemail - Voicemail notification is sent to API holder when the voicemail is available for the user. • deletchannel – This notification is sent when a channel expires due to validity. This notification will not be sent if 'deletchannel' is invoked by user.
'email'	Email of the user	Email of the user for whom the voicemail is available.
'voicemail'	Voicemail file name	Voicemail filename that is available to retrieve.
'src'	Phone number	Phone number of the leg for which this notification is being sent.

3.9.2 Notification Security

TringMe invoked the notification URL with MD5 signature and a random number. You can calculate the signature locally to verify the authenticity of invoker.

MD5 signature is calculated by concatenating random number, call tag and the private key.

3.9.2.1 Reverse DNS Lookup

Alternatively, you can do a reverse DNS lookup for the IP address to verify that it is from TringMe.

3.10 PHP Helper Functions

For convenience of PHP developers, we have released helper PHP functions for most of the commonly used operations. Helper functions are bundled together in the sample source code available at <http://tringme.com/samples.html> . If helper function is used, the above Click 2 Call sample code reduces to:

```
// TringMe Call using helper function
<?php

include_once ("tringmehelper.php");

$response = TringMeCall($privatekey, $publickey, $callsrc
, $calldest);
```

4 Adobe Flash API

TringMe uses standard NetStream and NetConnection interfaces which have been available since Flash 7. This allows even an inexperienced flash developer to develop new widgets by writing few lines of code. Since NetConnection and NetStream are widely used and tons of documentation and help already are available on the web, readers are requested to refer those resources and we will not cover it here. Instead, the following gives the steps and specifications for making a call using TringMe. For exact usage, refer to TringMe widget source code.

4.1 Making Call

Making a Call is as simple as:

1. Create Flash NetConnection object
2. Two NetStream Objects (A & B) attached them to NetConnection
3. Create Microphone Object and attach it to a NetStream A
4. Publish NetStream A
5. Play NetStream B

Complete working source code of a minimal phone widget is below.

```
nc = new NetConnection();

// handle asynchronous notification and failures
nc.onStatus = ConnectStatus;
nc.connect(TringMeURI); // Refer section 4.2 below

//Create NetStream A and attach it with Microphone
nsA = new NetStream(nc);
mic = Microphone.get();
mic.setRate(8);
mic.setSilenceLevel(0);
nsA.attachAudio(mic);
nsA.publish(randomstreamname, "record");

// Create NetStream B and play
nsB = new NetStream(nc);
nsB.play("tring", -1);
```

4.2 NetConnection targetURI Specification

To make a Flash phone call, the URI string should be formatted in one of the below mentioned specifications:

```
rtmp://sip.tringme.com/<cookie>/<destination>[/<tag>][/<dtmf>] [/<callerid>][/<duration>]
```

For the 'cookie' parameter, cookie or channel can be passed, as obtained from 'login' or 'setchannel' API. For more information on channel and 'setchannel' API, refer to Section 5.

'tag' [Optional] - User supplied parameter to associate a call in the Call Data Records (obtained by 'getcdr'). 'tag' is alphanumeric only (a-z, A-Z, 0-9), and all other characters will be replaced by 'X'.

'callerid' [Optional] – Caller ID – caller id is passed only if destination is SIP. To use callerid on TringMe's network requires special activation due to the privacy & security issues [contact bizdev@tringme.com]

'dtmf'[Optional] – DTMF

'duration' [Optional] – Maximum duration of the call

4.3 Ending Call

To end the call, close NetStream by calling close method.

```
// close stream A, it optional to close stream B as it will be automatically closed by TringMe  
nsA.close();
```

If the call is terminated by called party, TringMe will close the connection. Notification will be posted to onStatus handler of class NetConnection

```
function ConnectStatus( pInfo:Object ) {  
    if ( pInfo.code == "NetConnection.Connect.Success" ) {  
        // do something to indicate successful connection with TringMe  
    } else if ( pInfo.code == "NetConnection.Connect.Closed" ||  
        pInfo.code == "NetConnection.Connect.Rejected" )  
    {
```

```
        // close streams and clean up

    }

}
```

4.4 Call Status Notifications in Flash

To receive call status notification in Flash itself (in addition to the real-time notification described in 3.9), register a status handler as shown below. Currently only following status notifications are supported:

TringMe.Status.00200 – Call Connected Status

```
//Create NetConnection and connect

nc = new NetConnection();

// handle async notification and failures
nc.onStatus = ConnectStatus;
nc.connect(TringMeURI);

//Create NetStream A and attach it with Microphone

nsA = new NetStream(nc);
mic = Microphone.get();
mic.setRate(8);
mic.setSilenceLevel(0);
nsA.attachAudio(mic);
nsA.publish(randomstreamname, "record");

// Create NetStream B and play

nsB = new NetStream(nc);
nsB.onStatus = NSOnStatusHandler;

nsB.play("tring", -1);

...

// Declare the OnStatus handler for receiving status updates.

// It should be registered as shown below
```

```

NSOnStatusHandler = function (info:Object) {

    trace(info.code);

    if(info.code == "TringMe.Status.00200") {

        // Call is connected ...

    }

}

```

4.5 Using TringMe's Premade Flash Widgets

While you can build your own Flash phone widgets as described above, you can also use TringMe's premade widgets easily. Integrating these readymade widgets is as simple as adding few lines of HTML code.

4.5.1 Embedding TringPhone widget

TringPhone is a Flash widget and can be embedded into your website like any other flash file (.swf) file. TringPhone takes cookie or channel (as described in Section 5) as a parameter. Use 'login' or 'setchannel' API to obtain the cookie or the channel respectively. Following are the details of flash object:

Sample URL:

<http://login.tringme.com/swf/tringphone.swf?channel=<cookie or channel>>

Suggested Flash Parameters:

- **Width:** 280
- **Height:** 400
- **majorversion:** "6"
- **wmode:** "transparent"

4.5.2 Embedding Push-n-talk widget

TringMe Push-n-talk is a Flash widget can be used for fixed dialing and can be embedded into your website like any other flash file (.swf) file. Any user visiting your website can just push and start talking to you without dialing any number.

Push-n-talk takes following parameters. **All the parameters except 'channel' & 'name' are OPTIONAL:**

Parameter	Description
'channel'	Channel (as described in Sec 5)
'dtmf'	DTMF
'callerid'	Caller ID

'tag'	Call tag
'duration'	Max duration of the call
'name'	Name to be displayed on Widget
'greeting'	URL of mp3 file to be played before placing the call
'postgreeting'	URL of mp3 file to be played after the call
'ctxt'	Call Text - Replaces default call text on the widget. Default 'Click to Call'. For Localization & customization.
'htxt'	Hangup Text - Replaces default hangup text on the widget.
'ttx'	Call Termination Text – Replaces default call termination text on the widget
'background'	URL of Background image

Sample Widget URL:

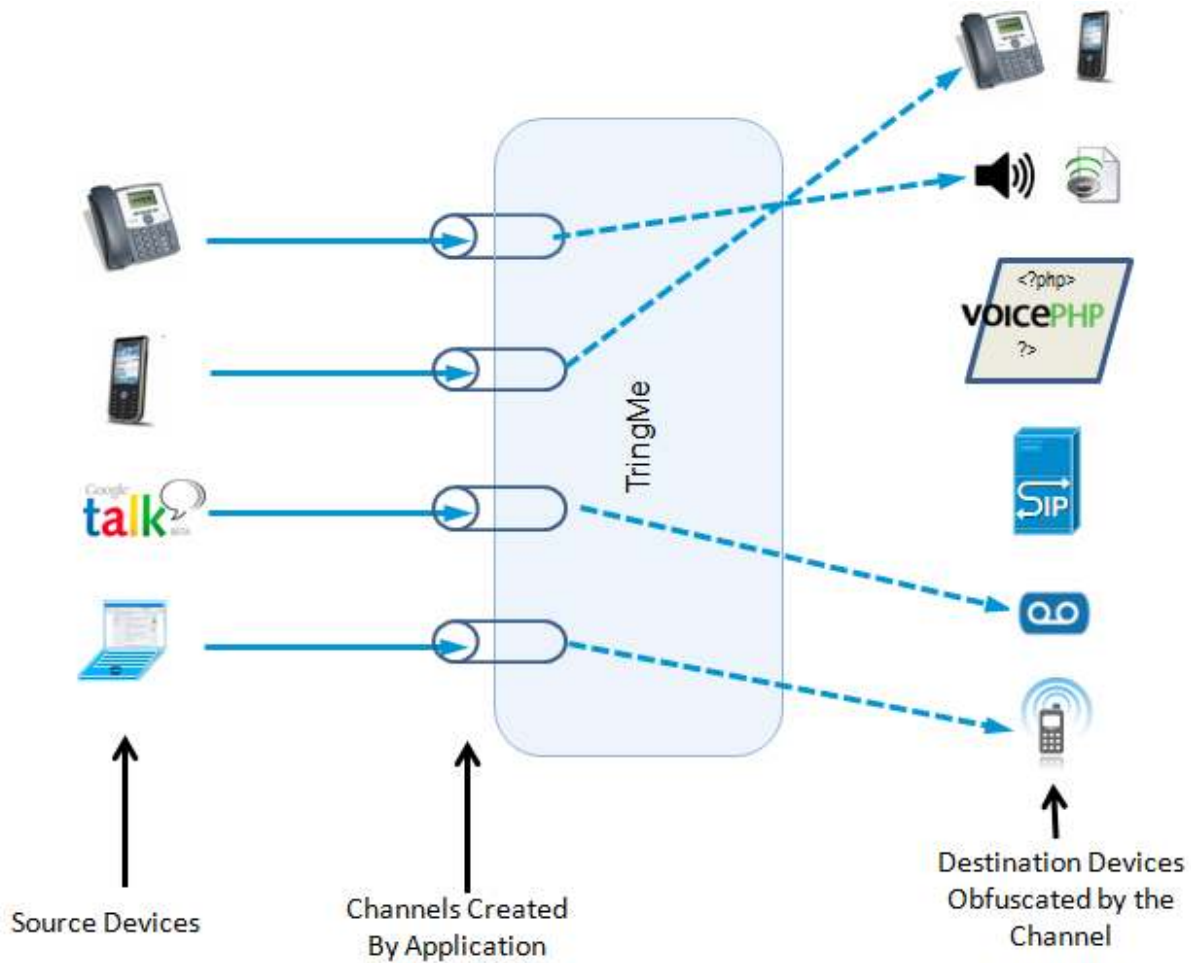
`http://login.tringme.com/swf/tringmewidget.swf?channel=<channel>&name=My+Name&greeting=http://mywebsite.com/greeting.mp3`

Suggested Flash Parameters:

- **Width:** 215
- **Height:** 138
- **majorversion:** "6"
- **wmode:** "transparent"

5 Channels

So far we have used cookie and destination combination to make the calls. Channel offers finer control over call destinations. In a way, you can perceive a channel as restricted version of cookie having defined rules on what is allowed over channel. For example, a channel could allow to call worldwide, or it could be restricted to call a particular destination or a set of destinations only. In other words, voice channel defines the complete configuration for the destination leg.



Channels are ideally suited for applications with recurrently tasks (say bulk calls or conferencing) or where you cannot expose the cookie (say, Flash Push & Talk applications). In such cases, you can define channel with fixed destination and use where you have used cookie earlier (e.g. call or Flash API).

Channel allows developer to control some aspects of the destination that will be connected – type of device, duration of the call and so on. Most importantly, developer may want to set these channels and re-use them from thereon without. Channels make it easy for one to create destination specific configuration and there onwards refer to them using just a single identifier. There are two key parts to using Voice Channels.

1. Channel Creation – This will show you how to create a channel, what are the available options and the capabilities that a channel offers.
2. Using Channels – This will show you how a channel can be used instead of cookie.

5.1 Channel Creation

Before we plunge into the details of creating channels, let’s understand some key concepts of channels:

- Channels are associated with a user. It can be associated with API owner or the user whose cookie is passed during the channel creation.
- Channels can specify duration, validity or usage to control the ‘expiry’ of the channel. Please refer to the ‘setchannel’ API for precise list of parameters.
- Channels can be created, modified or deleted.

Since channels control how the call is ‘routed’ to the destination, refer to the destination section 3.8. Once the channel is created, it is available till it expires or is deleted. Expiration can happen when duration, validity or uses expires. It is recommended that developers delete a channel explicitly if they don’t need it anymore. TringMe will send notification when any channel is deleted due to usage.

5.2 Channels APIs

Channel can be used exactly same as cookie in REST telephony API, or in Flash API. Here are the basic channel related APIs in REST

5.2.1 Create or Modify a Channel

5.2.1.1 Description

This API creates a channel. Several parameters can be specified to define the behavior of the channel.

5.2.1.2 Parameters

op	‘setchannel’
cookie	Cookie which was obtained using ‘login’.
channel [optional]	Channel. If a channel is specified, then that channel is modified. If none is specified, then a new channel is created.

dest [optional]	<p>Destination for this channel. Please refer to the section 3.8 for all the supported types of destination.</p> <p>For Phone and SIP types of destination, wildcards can be specified.</p> <ul style="list-style-type: none"> • When wildcards are specified in Phone, only phone numbers which match the criteria would be allowed to go through this channel. *, ? should be used for wildcards. • In SIP, wildcard can be specified as well. <ul style="list-style-type: none"> ○ sip:@mysipserver.com – to call all supported users/numbers at mysipserver.com SIP. ○ sip: @<sip-id> - to call any number using the sip-id.
duration [optional]	<p>Call duration after which the call should automatically disconnect. The duration is the time from the moment call is answered till the time when the call disconnects</p>
validity [optional]	<p>Validity of this channel in seconds.</p>
maxusage [optional]	<p>Maximum number of call attempts.</p>
tag [optional]	<p>User supplied parameter to associate a call in the Call Data Records.</p> <p>This should be alphanumeric only (a-z, A-Z, 0-9). All other characters will be replaced by 'X'. It can be a maximum of 16 characters</p>
dtmf [optional]	<p>DTMF that should be sent on this channel.</p>
flag [optional]	<p>One or more combinations of the flags below:</p> <p>0 – Disables all notifications. 1 – Send call completed and DTMF input notifications. This is the default value. 2 – Send call-in-progress notifications (e.g. SRCANSWERED)</p> <p>For more details on real-time notifications, refer to section 3.9.</p>

5.2.1.3 Return Value

If successful, it will return 'OK <channel>'. The <channel> is the identifier that can be used in APIs which require channel as an input.

5.2.1.4 Example

<http://api.tringme.com/api.php?op=setchannel&cookie=1234567890123456789012345678901&dest=gtalk%3AJoe.dirt%40gmail.com&apikey=b12345678901234567890123456789012345&sig=f1234567890123456789012345678901>

5.2.2 Get Channel Information

5.2.2.1 Description

This API retrieves all the call records for a particular user. It returns the data in JSON format.

5.2.2.2 Parameters

op	'getchannel'
channel	Channel whose information is being requested.

5.2.2.3 Return Value

If successful, it will return 'OK <validity>'.

5.2.3 Delete Channel

5.2.3.1 Description

This API retrieves all the call records for a particular user. It returns the data in JSON format.

5.2.3.2 Parameters

op	'deletechannel'
channel	Channel to be deleted.

5.3 Accessing Channels using SIP

Channel can also be accessed by outside world using SIP. Each channel has its own unique SIP address – **<channel>@sip.tringme.com** which is accessible from any SIP capable device or phone in the world. This allows you to create your own phone service, IVR or conference service and accessible using SIP.

5.3.1 Mapping Channel to DID (Direct Inward Dialing)

DID vendor maps DIDs to sip address. You can map a DID to channel just by mapping channel's SIP address as described in 5.3

6 Billing and Charging

TringMe platform including VoicePHP is completely free. In most cases, TringMe only charges for the phone calls. However, if none of the leg is over phone, we need to at least charge minimum just to meet the infrastructure cost and resource utilization. All such calls will be charged at the rate 1 sipcredits/minute. SIP credits can be purchased from TringMe website or in bulk by contacting support@tringme.com.

All Conference call and SIP legs will always be charged at the rate 1 sipcredit/minute irrespective of other legs.

6.1 Simultaneous Calls

TringMe's platform allows placing unlimited simultaneous calls; only limited by credit balance. By default, TringMe reserves credits for 90 minutes before placing the call. These 'reserved' credits are freed once the call ends and only the charges for the actual duration of the call are applied to the user. The default 'reserved' credits can be over-ridden by specifying the 'duration' parameter in 'call' or Flash APIs in which case the credits reserved will be only for the specified duration only.

6.2 Conference Charges

Conference charges are as follows:

- Each Conference Charge: \$1 USD for first hour, 0.25USD per hour thereafter.
- Usage charge: 1 sip credit per participant per minute.
- If used with callback facility, phone charges as actuals.

If you plan to use for larger quantity, we recommend our private switch. Please contact bizdev@tringme.com for further details on this.

7 How To

In this section, you will find common techniques to use APIs for specific purpose. Given the capability of the API, there are several different ways to use them to your benefit or application requirement. Our intent here is to give you basic building blocks which can then enable you to use them for more complex requirements as needed.

7.1 Create a Conference

Setting up a conference is quick and easy. Here are the steps to follow:

1. Use 'createconference' API to create a conference. You can specify optional parameters like duration and when to control scheduling aspects of the conference. If successful, it will return a <conf-id>.
2. Once the <conf-id> is available, conference can be invoked in multiple ways:
 - a. Using [conf:<conf-id>:<flag>](#) as a 'dest' parameter in 'call' API. The 'src' parameter can contain the phone numbers of the users as a comma separated list.
 - b. Creating a channel and specify [conf:<conf-id>:<flag>](#) in the 'dest'.
 - c. Use that channel created in step above (b) to invoke calls from Flash, 'call' API.
 - d. Use SIP client and invoke [sip:<channel>@sip.tringme.com](#) to join the conference.
 - e. If you would like to assign a DID number to your conference, map the DID number to channel as described in section 5.3.1

7.2 Best Practices

7.2.1 Calculating available credits

When placing multiple calls or sending multiple SMS, ensure that sufficient credits are available to place the calls and SMS to all the numbers upfront. If credits are insufficient, TringMe will place the call (or send SMS) to as many number as possible and ignore remaining numbers. Call Records will identify the calls placed or SMS sent.

For conferencing or bulk calling, it is recommended to specify a duration and then calculate the credits required for that duration.

8 Support

If you have any questions or have encountered a bug, please send a mail to support@tringme.com.

If you have specific requirements which cannot be met using these APIs, send us a mail at bizdev@tringme.com with your requirement details and we will get back to you.

For latest TringMe news and updates, follow us on Twitter: @tringme

Visit TringMe Blog: <http://blog.tringme.com> for up-to-date information & articles.