



# 4PSA VoipNow ® 1.6.4 Call API Guide

For more information about 4PSA VoipNow, check: http://www.voipnow.com
Copyrights 2006-2008 Rack-Soft, LLC.VoipNow is a registered trademark of Rack-Soft, LLC.

## Call API Guide

Manual Version 41930.21 at 2008/09/24 18:23:13

For suggestions regarding this manual contact: docs@4psa.com

Copyright © 2002-2008 Rack-Soft, LLC

All rights reserved

Distribution of this work or derivative of this work is prohibited unless prior written permission is obtained from the copyright holder.

VoipNow is a Registered Trademark of Rack-Soft, LLC.

4PSA is a Registered Trademark of Rack-Soft, LLC.

Asterisk is a Trademark of Digium, Inc.

Linux is a Registered Trademark of Linus Torvalds.

All other trademarks and copyrights are property of their respective owners.

# Table of Contents

Chapter 1. What is Call API	4
Chapter 2. How to Use Call API	
The Call API Functions	
Encrypted Authentication	9

## Chapter 1

# What is Call API

Call API is an application that allows users to handle 4PSA VoipNow calls remotely, using HTTP requests. Users can initiate or terminate calls and track active calls for a particular extension. The application is located at:

/usr/local/voipnow/admin/htdocs/callapi/callapi.php

Call API is available to users with administrator, client or extension accounts as follows:

- Administrators and clients will have to use their Call API password setup in 4PSA VoipNow authenticated interface
- For extension accounts, one will have to use the telephone's password or, if such a password hasn't been set, the web interface password



## Note

In order to use the Call API functions for a particular extension, you first need to activate Call API access. To activate the Call API access, click the chosen extension

link in the Extension Management page. Next, click the Phone Terminal Setup icon available in the Tools area. In the new page that opens, select the Call API access checkbox available in the Basic settings area and click OK.



#### Note

In order to use the Call API functions for a particular extension, both the extension's client and administrator need to have Call API enabled.

## Chapter 2

## How to Use Call API

## The Call API Functions

The Call API is a work in progress. For now, only four basic functions are available. Development is under way and the number of features will increase in the future. The available functions are:

1. newcall - This function allows users to initiate phone calls from an extension. For this purpose, one will use the following typical URL:

```
<server_ip>/callapi/callapi.php?
request=newcall&user=<mylogin>&pass=<mypass>&phone=<phonenumbertocall>
[&callerid=<calleridnum>]&interactive=<yes|no>[&from=<number>]
[&timeout=<seconds>][&style=<stylesheet>]
```

### where:

- <server\_ip> is 4PSA VoipNow server's IP address
- <mylogin> is the user's login name.
- <mypass> is the user's login password.
- <phonenumbertocall> is the called party's telephone number.
- <calleridnum> is an optional parameter representing the caller's ID. It
  enables the called party to see the name of the caller. When not specified,
  the caller's phone number will be displayed instead.
- The interactive parameter allows the user to monitor how the function performs in real time. For instance:
  - when set to yes, the user will be able to see when the dialing is initiated, when the other party answers, for how long has the conversation been holding on or when one of the callers hangs up
  - when set to no, the user will only see in retrospect whether the call took place or not
- The from parameter is required only if the user is an administrator or a client, in which case he will have to specify the dialing extension. The phone number <number> must be entered using the format XXXX\*XXX (the extension's long number). If the user is an extension, this parameter is not required.

- <seconds> indicates for how may seconds the 4PSA VoipNow server will
  wait for the caller to pick up the phone before hanging up. The minimum
  value is 5 seconds. If not specified, this will be defaulted at 25 seconds.
- <stylesheet> is the name of the .css file used to display the callapi.php page. The style parameter is optional. Should you decide to customize your page layout, you can write your own .css files and save them in the Call API directory (<server\_ip>/callapi/).

For instance, the following URL:

```
http://192.168.14.128/callapi/callapi.php?
request=newcall&user=John&pass=jpassword&phone=001234
&callerid=John&interactive=yes&timeout=10&style=style.css
```

allows user John who uses password jpassword to call another user (say Mary) at phone number 001234 while sending her his ID (her phone screen will display she has a phone call from John).

2. newextcall - This function allows users to initiate phone calls that connect two numbers on the public phone network. These phone numbers are connected toghether using a callback extension.



## Note

Note: Technically any two phone numbers can be connected toghether, based on the settings of the callback extension, however it is recommended to setup a callback extension that calls only users from authorized CallerIDs. In this way the request will involve a fixed external number (the authorized CallerID) and any other phone number.

To initiate a call, one will use the following typical URL:

```
<server_ip>/callapi/callapi.php?
request=newextcall&user=<mylogin>&pass=<
mypass>&ccbext=<ccbextension>&caller=<callerphonenumber>&phonenumber>&phonenumber>
[&callerid=<calleridnum>][&callduration=<callduration>]
[&from=<number>][&timeout=<seconds>] [&style=<stylesheet>]
```

#### where:

- <server\_ip> is 4PSA VoipNow server's IP address
- <mylogin> is the user's login name.
- <mypass> is the user's login password.
- <ccbextension> is the callback type extension under which account the call is made, in XXXX\*XXX format.

- <callerphonenumber> is the number of person first called, usually a known number and maybe an authorized CallerID.
- <calleridnum> is an optional parameter representing the caller's ID. It enables the <callerphonenumber> to see the name of the <phonenumbertocall>. When not specified, the <callerphonenumber> will be displayed instead.
- <callduration> is the maximum duration of the call, can be setup dinamically.
- <from> is required only if the user is an administrator or a client, in which
  case he will have to specify the dialing extension. The phone number
  <number> must be entered using the format XXXX\*XXX (the extension's
  long number). If the user is an extension, this parameter is not required.
- <seconds> indicates for how may seconds the 4PSA VoipNow server will wait for the <callerphonenumber> to pick up the phone before hanging up. The minimum value is 5 seconds. If not specified, this will be defaulted at 25 seconds.
- <stylesheet> is the name of the .css file used to display the callapi.php page. The style parameter is optional. Should you decide to customize your page layout, you can write your own .css files and.
- 3. hangup This function allows users to hang up all active calls for an extension. For this purpose, one will use the following typical URL:

```
<server_ip>/callapi/callapi.php?
request=hangup&user=<mylogin>&pass=<mypass>&interactive=<yes|
no> &exten=<exten_number>[&style=<stylesheet>]
```

#### where:

- <server\_ip> is 4PSA VoipNow server's IP address
- <mylogin> is the user's login name.
- <mypass> is the user's login password.
- The interactive parameter allows the user to monitor how the function performs in real time.
- <exten\_number> is the extension's number. If the user logs in as an extension, this parameter is ignored.
- <stylesheet> is the name of the .css file used to display the callapi.php page. The style parameter is optional. Should you decide to customize your page layout, you can write your own .css files and save them in the Call API directory (<server\_ip>/callapi/).

For instance, the following URL:

```
http://192.168.14.128/callapi/callapi.php?
request=hangup&user=admin&pass=adminpass
&interactive=no&exten=0241*043&style=hangup.css
```

the admin user with the adminpass password will shut down all the active calls for extension 0241\*043

4. listexten - This function allows users to monitor the statuses of a particular user's calls. For this purpose, one will use the following typical URL:

```
<server_ip>/callapi.php?
request=listexten&user=<mylogin>&pass=<mypass>&interactive=<yes|
no> &showtime=<time>[&style=<stylesheet>]
```

#### where:

- <server\_ip> is 4PSA VoipNow server's IP address
- <mylogin> is the user's login name.
- <mypass> is the user's login password.
- The interactive parameter allows the user to monitor how the function performs in real time.
- <time> is the refresh interval for the data displayed in the interactive mode. The minimum value is 10. Any value below 10 will be accounted as 10.
- <stylesheet> is the name of the .css file used to display the callapi.php page. The style parameter is optional. Should you decide to customize your page layout, you can write your own .css files and save them in the Call API directory (<server\_ip>/callapi/).

If the user is an extension, the function will display the extension's active calls. If the user is a client or administrator, the function will display all the client's active calls or all the calls on the server respectively.

For instance, the following URL:

```
http://192.168.14.128/callapi/callapi.php?
request=listexten&user=Mary&pass=mpassword&interactive=yes
&showtime=15&style=mystyle.css
```

allows user Mary using password mpassword to see the active calls on her phone. The page displaying the active calls will be refreshed every 15 seconds using stylesheet mystyle.css.

5. test - This function allows users to check whether Call API is enabled for a particular user or not. Call API might be unavailable for a number of reasons: it might not be activated in 4PSA VoipNow (see above for details),

the username/password might be wrong or the local PBX interface might be unavailable. To access the test function, use the following typical URL:

```
<server_ip>/callapi/callapi.php?
request=test&user=<mylogin>&pass=<mypass>[&style=<stylesheet>]
[&xml=1]
```

#### where:

- <server\_ip> is 4PSA VoipNow server's IP address
- <mylogin> is the user's login name.
- <mypass> is the user's login password.
- <stylesheet> is the name of the .css file used to display the callapi.php page. The style parameter is optional. Should you decide to customize your page layout, you can write your own .css files and save them in the Call API directory (<server\_ip>/callapi/).
- Adding the optional parameter &xml at the end of your request allows you to display the results in XML format. When set to 1, the function is activated. Otherwise it is deactivated.

For instance, the following URL:

```
http://192.168.14.128/callapi/callapi.php?
request=test&user=John&pass=jpassword&style=style.css&xml=1
```

tests whether user John with password jpassword is able to use the Call API functionality. The information will be generated in XML format.

## **Encrypted Authentication**

When using Call API, you can use OpenSSL or SHA256 data encryption to protect your private data:

 If you would like to use OpenSSL to encrypt your username and password when using the Call API functions described <u>above</u>, replace the following URL sequence:

```
user=<mylogin>&pass=<mypass>
with
auth=<login_auth>
where <login_auth> is the base64 encrypted conversion of sequence:
user=<mylogin> pass=<mypass>
```

Encrypt the above sequence using the public API key that was configured in 4PSA VoipNow. For this purpose, access the Encryption keys management

page, by clicking the Encryption keys icon, located in the PBX features page. You can recognize the API key by the icon (if all icons in the API column are grayed out, there is no API key set.). Click the certificate's public key icon to download the public key, or click on the certificate's name to access all its features, including its public key.

• If you would like to use SHA256 hashing algorithm for your password when using the Call API functions described <u>above</u>, replace the following URL sequence:

```
&pass=<mypass>
with
&shapass=<SHA256(password)>
where <SHA256(password)> is the SHA256 encrypted user password.
```