XML / JSON API Reference

MAINTENANCE RELEASE 80

www.portaone.com

Copyright Notice & Disclaimers

PortaSIP® Media Applications API Reference, September 2019
Maintenance Release 80
V1.80.01

Please address your comments and suggestions to: Sales Department, PortaOne, Inc. Suite #408, 2963 Glen Drive, Coquitlam BC V3B 2P7 Canada.

Table of Contents

Preface

This document provides information for developers who want to interface their applications with PortaSIP® media applications via XML and JSON API. The PortaBilling® XML and JSON API is described in the **PortaBilling XML / JSON API Reference**.

Where to get the latest version of this guide

The hard copy of this guide is updated upon major releases only, and does not always contain the latest material on enhancements that occur in-between minor releases. The online copy of this guide is always up to date, and integrates the latest changes to the product. You can access the latest copy of this guide at: **www.portaone.com/support/documentation/**.

Conventions

This publication uses the following conventions:

Exclamation mark draws your attention to important actions that must be taken for proper configuration.

NOTE: Notes contain additional information to supplement or accentuate important points in the text.

Trademarks and Copyrights

PortaBilling®, PortaSIP® and PortaSwitch® are registered trademarks of PortaOne, Inc.

1. XML / JSON API Overview

Security

Connection to the XML / JSON API interface is provided via HTTPS. Authentication is done using authentication pairs (login-password or login-API token). Each subsequent request to the API should contain the **auth_info** structure.

Note that we strongly recommend using the *session_id* property (which is received during the authorization via the **LoginRequest**) in the **auth_info** structure for all session requests. Otherwise, if you use login-password authentication pairs for every request, new sessions will be created and will cause additional load to the database.

XML API

XML (SOAP) API has its own advantages and drawbacks as compared with JSON API. Among the benefits are the following:

- There is a wide range of reusable software available to programmers to handle XML so they do not have to re-invent code.
- XML (SOAP) is more verbose compared with JSON, but because of this, the data encoding result is typically larger than the equivalent encoding in JSON API.

Access to XML API

Proxy (URL): **https://web-server.yourdomain.com:8443/soap/soap.fcgi**
SOAP URI (namespace): **https://web-server.yourdomain.com/UM/SOAP/**

Please replace the **web-server.yourdomain.com** with the actual hostname of your PortaSwitch® web server.

Error Handling

SOAP faults are used to carry error information within a SOAP message. If the actual response has a SOAP fault element as the body entry, then an error has occurred. In this case, the accuracy of any other fields in the response cannot be guaranteed, and you should only use the fault sub-elements to identify the error. Currently, these sub-elements are as follows:

- **faultcode** is intended for use by the client software and provides an algorithmic mechanism for identifying a fault.

- **faultstring** provides a human-readable explanation of a fault, and is not intended for algorithmic processing.

JSON API

As an alternative to XML API, PortaSwitch® supports JSON API, thus providing your development department with a choice of Web Application Services that can be used. Among the advantages of JSON API are the following:

- Simple data structures that can be easily read and written.
- JSON format is faster in parsing and generating data due to simple syntax, thus there is little influence on web server performance.
- Supports the same methods as those in the SOAP.
- Simplifies the creation of front-end web sites that receive and modify data with minimum impact on performance.

Access to JSON API

All JSON requests to PortaSIP® Media Server API must be sent to the following URL: **https://<web-server.yourdomain.com>:<port>/rest/<service>/<method>/**

Please replace the **web-server.yourdomain.com** with the actual hostname of your PortaSwitch® web server.

Replace <port> with the required port. The JSON interface is available for administrators on port 443, the interface for customers is available on port 8444, the interface for resellers is available on port 8442 and the interface for accounts is available on port 8445.

Replace <**service**> with the API service that contains the required method (e.g. specify the **SMPreferences** service to manage voice mailbox preferences.)

Replace <**method**> with the required API method (e.g. specify **set_folder_preferences** method in order to change mailbox folder preferences.)

Here is an example of the URL the POST request to be sent to:

```
https://demo.portaone.com:8443/rest/SMPreferences/set_folder_preferences/
```

Sending an HTTP request

For HTTP requests you must include the following parameters (in JSON format) in the POST request body:

- `auth_info` – The mandatory authentication. Not used with the methods to establish the API session information (see the **Security** section).
- `params` – A set of method parameters (in JSON format) that depend on a method structure. Note that method parameters and their structures are the same as those in the SOAP.

The Content-Type header field used with a HTTP POST request must have one of the following values:

- application/x-www-form-urlencoded
- multipart/form-data

Please note that special characters must be escaped.
For example, if you want to send the HTTP GET request:

```
https://111.111.11.11:8443/rest/AutoAttendant/set_menu_transiti
on/{"login":"000111222","password":"mysEcReTp@ss","session_id":
null,"domain":"111.111.11.11"}/{"i_menu":12,"transition_info":"
event":"#","action":"Transfer","target_i_menu":0,"destination":
"1","play_prompt":"Y"}}
```

it must be transformed into the following form:

```
https://111.111.11.11:8443/rest/AutoAttendant/set_menu_transiti
on/%7B%22login%22%3A%22000111222%22,%22password%22%3A%22
mysEcReTp@ss%22,%22session_id%22%3Anull,%22domain%22%3A%22111.1
11.11.11%22%7D/%7B%22i_menu%22%3A12%22transition_info%22%3A%7B%
22event%22%3A%22%2523%22%2C%22action%22%3A%22Transfer%22%2C%22t
arget_i_menu%22%3A0%2C%22destination%22%3A%221%22%2C%22play_pro
mpt%22%3A%22Y%22%7D%7D
```

You can run JSON requests in the dry run mode. The dry run mode does not execute the method itself. Instead, it checks input arguments according to the schema validation rules and returns validation results. To run a request in the dry run mode, add the `aux_info` structure into the request. The structure has the following fields:

- `dry_run` – indicates that the method must be run in the dry run mode.

Examples of API requests

The examples below are given with the use of cURL command line tool.

establish API session

Request:

```
curl https://demo.portaone.com:8444/UM/rest/Session/login
-d auth_info='{}'
-d params='{"login":"SIPAccounts", "password":"123password"}'
```

Response:
{"session_id":"f1ab18fe5a3decf0ba828e56a3d9e982"}

Error handling

If the server returns the '500 Internal Server Error' status code in the HTTP response, then the response body contains a JSON object which includes two elements (keys) that carry error information:

- **faultcode**, that is intended for use by the client software and provides an algorithmic mechanism for identifying the fault.
- **faultstring**, that provides a human readable explanation of the fault, and is not intended for algorithmic processing.

WSDL

Each PortaSIP® Media Server has its own set of WSDL documents available for download from the web server. These documents can be downloaded from:

- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Session.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Types.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Voicemail.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=SMPreferences.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=DialDirectory.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=AutoAttendant.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Conference.xsd**

All requests to PortaSIP® Media Server API are handled via an SSL connection. By default, PortaSIP® Media Server installations contain a self-signed certificate that provides the means to encrypt data. However, since this certificate's authenticity cannot be validated, you may experience some problems when connecting to an SSL site. In that case, it may be necessary to obtain a certificate from a genuine certificate authority. Another option is to generate your own certificate authority and have certificates deployed to all API clients. However, this goes beyond the scope of the present document.

2. Reference

Notation conventions

The following typographic conventions apply throughout this chapter:

- * – A value can be entered for this property only when inserting new records and cannot be changed later.
- ** – This property is read-only, and its value cannot be changed.
- Mandatory properties (whose value must be entered during insertion, and cannot be set to an empty value later) are underlined.
- n – This property can be used with the **nil** attribute to indicate that it is blank (has no content):
    - In the *Request* message the **xsi:nil="true"** attribute can be used to clear the property (set value to NULL in the database).
    - In the *Response* message a property has the **xsi:nil="true"** attribute if it is blank (has the NULL value in the database).

Establishing an authenticated session

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/Session

Methods

login

Parameters: **LoginRequest**
Return value: **LoginResponse**

Checks the validity of login and password and returns session_id on success. An API fault is generated on failure.

logout

Parameters: **LogoutRequest**
Return value: **LogoutResponse**

Terminates the session. You should call logout() to terminate the session properly.

Type reference

LoginRequest structure

Property	Type	Description
login	string, 32 char max	Account ID specified on web interface
domain	string	Media Server Domain corresponding to billing environment that the account belongs to
password	string, 16 chars max	Password specified on web interface

LoginResponse structure

Property	Type	Description
session_id	string, 32 chars max	ID of newly opened session

LogoutRequest structure

Property	Type	Description
-	-	-

LogoutResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

Global methods and types

Type reference

The structure below is used to pass authentication data to the API method. There are two possible ways to authenticate an API request: create a session and pass session_id in auth_info, or pass all the required credentials (login/domain/password) in every API request.

auth_info structure

Property	Type	Description
login	string, 32 chars max	Account ID specified on web interface

domain	string	Media Server Domain corresponding to current billing environment
password	string, 16 chars max	Account's password for web self-care interface
or alternatively:		
session_id	string, 32 chars max	The unique ID of previously opened API session

Voicemail Settings

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/Voicemail

Methods

get_vm_settings

Parameters: **GetVMSettingsRequest**
Return value: **GetVMSettingsResponse**

This method enables an API user (account) to get general voicemail settings from the PortaSIP® Media Server database.

set_vm_settings

Parameters: **SetVMSettingsRequest**
Return value: **SetVMSettingsResponse**

This method enables an API user (account) to set general voicemail settings in the PortaSIP® Media Server database.

get_vm_greeting

Parameters: **GetVMGreetingRequest**
Return value: **GetVMGreetingResponse**

This method enables an API user (account) to get the sound prompt for a specified greeting from the PortaSIP® Media Server database. The sound file is returned in a MIME attachment.

set_vm_greeting

Parameters: **SetVMGreetingRequest**
Return value: **SetVMGreetingResponse**

This method enables an API user (account) to set the sound prompt for a specified greeting type. The sound file is sent in a MIME attachment.

Type reference

GetVMSettingsRequest structure

This method doesn't have any parameters.

GetVMSettingsResponse structure

Property	Type	Description
vm_settings	VMSettings	Complete information about general voicemail settings

SetVMSettingsRequest structure

May include **any** of the following properties:

Property	Type	Description
vm_settings	VMSettings	Complete information about general voicemail settings

SetVMSettingsResponse structure

Property	Type	Description
vm_settings_saved	int	1 in case of success

VMSettings structure

Property	Type	Description
password	string	Password for accessing voicemail via IVR
password_ask	string	• yes – ask for password when accessing voicemail via IVR; • no – don't ask for password when accessing voicemail via IVR
prompt_levels	string	PortaSIP® Media Server offers three voice prompt settings in each supported language: • standard • extended • rapid
announce_dt	string	Announce the date and time when each voicemail was sent.

		Values: • yes • no
auto_play	string	Auto-play new voicemail(s) when a call to voicemail is established. Values: • yes • no
greetings	string	Type of greeting for users wishing to leave a voicemail. Values: • standard • extended • personal; • name
fax_file	string	Format for received faxes: • multi_png • multi_tiff • pdf • tiff
ext_email	string, max 128 chars	External email for forwarding, copying, and notifying
ext_email_action	string	Action for external email: • none • forward • notify • copy • fwd_as_attachment
ext_email_vm_fmt	string	Voice message audio format: • au • mp3 (default) • wav
enable_disa	string (Yes/No)	Enable DISA functionality for customer's voicemail
disa_password	string	Password for using DISA functionality

GetVMGreetingRequest structure

Property	Type	Description
greeting_type	string	Values: • standard • extended • personal

		• name

GetVMGreetingResponse structure

Property	Type	Description
filename	string	Filename of greeting attached to SOAP response in a MIME attachment

SetVMGreetingRequest structure

Property	Type	Description
greeting_info	GreetingInfo structure	Complete information about general greeting's settings

GreetingInfo structure

Property	Type	Description
greeting_type	string	Values: • extended • personal • name
filename	string	Filename of greeting attached to SOAP request in a MIME attachment

SetVMGreetingResponse structure

Property	Type	Description
success	int	1 in case of success
i_audio_file	int	The ID of the audio file in the CodecConverter conversion queue

Folder preferences and Mailbox and message display options

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/SMPreferences

Methods

get_folder_preferences

Parameters: GetFolderPreferencesRequest
Return value: GetFolderPreferencesResponse

This method enables an API user (account) to get the preferences of his mailbox.

set_folder_preferences

Parameters: SetFolderPreferencesRequest
Return value: SetFolderPreferencesResponse

This method enables an API user (account) to set the preferences of his mailbox.

get_display_preferences

Parameters: GetDisplayPreferencesRequest
Return value: GetDisplayPreferencesResponse

This method enables an API user (account) to get the display preferences of his mailbox and messages.

set_display_preferences

Parameters: SetDisplayPreferencesRequest
Return value: SetDisplayPreferencesResponse

This method enables an API user (account) to set the display preferences of his mailbox and messages.

Type reference

GetFolderPreferencesResponse structure

Property	Type	Description
folder_prefs	FolderPreferences structure	Complete information about the folder preferences; for more information, see below

FolderPreferences structure

Property	Type	Description
trash_folder	string	An IMAP folder where messages are moved on deletion. The messages are deleted completely if this field is set to "none"
draft_folder	string	An IMAP folder where the user can save a message in progress as a draft. The messages aren't saved if this field is set to "none"
unseen_type	int	The Unread Message Notification Type: • 1 – Only Unseen

		• 2 – Unseen and Total
unseen_notify	int	Enable Unread Message Notification: • 1 – No Notification • 2 – Only INBOX • 3 – All Folders
sent_folder	string	An IMAP folder messages are copied to after they are sent. The messages aren't copied if this field is set to "none"
unseen_cumulative	int	Enable the Cumulative Unread Message Notification. This controls the behavior of the message counter displayed next to each folder in the folder list. When enabled, if the folder contains sub-folders and is collapsed, then the message count includes all messages within all the sub-folders of that folder.
search_memory	int	Memory Search options. If the user searches the mailbox, the search can be saved for quick access later on. This option defines how many mailbox searches will be saved. Allowed Values: 0 (disabled), 1, 2, 3, 4, 5, 6, 7, 8, 9

SetFolderPreferencesRequest structure

Property	Type	Description
folder_prefs	FolderPreferences structure	Complete information about the folder preferences; for more information, see below

SetFolderPreferencesResponse structure

Property	Type	Description
success	int	1 in case of success

GetDisplayPreferencesRequest structure

GetDisplayPreferencesResponse structure

Property	Type	Description
display_prefs	DisplayPreferences structure	Complete information about the display preferences; for more information, see below

DisplayPreferences structure

Property	Type	Description
wrap_at	int	Defines how many characters to allow before wrapping text
truncate_sender	int	Specifies the lenght of the From / To fields (0 for full)
show_xmailer_default	int	When viewing a message, this displays which email service or client the sender used
editor_height	int	Specifies the height of the Editor Window
mdn_user_support	int	Specifies whether to enable the Mail Delivery Notification
truncate_subject	int	Specifies the lenght of the Subject Field (0 for full)
body_quote	string	Prefix each line of the original message with this symbol when replying or forwarding an email message
include_self_reply_all	int	Specifies whether to include user's address in CC when he chooses Reply All
sig_first	int	Specifies whether to prepend signature before Reply/Forward text
pf_cleandisplay	int	Specifies whether to display the View Printable Version link in a message
editor_size	int	Specifies the width of the Editor Window.
show_html_default	int	Specifies what version to show by default if a received message is sent in both text and HTML formats: 0 – Text version 1 – HTML version
page_selector_max	int	Specifies the maximum number of pages that will be shown at one time
internal_date_sort	int	Specifies whether to sort messages by Received Date
page_selector	int	Specifies whether to show Page Selector. When enabled, message pages will be shown above and below the list of messages, allowing the user to quickly jump to a specific message page
addrsrch_fullname	string	Specifies the format of addresses added from the address book: • "Noprefix" – No prefix, address only

		• "Nickname" – Nickname and address • "Fullaname" – Full name and address
show_num	int	Specifies the number of messages that will be shown on one page
show_images	int	Specifies whether to display attached images with the message

SetDisplayPreferencesRequest structure

Property	Type	Description
display_prefs	DisplayPreferences structure	Complete information about the display preferences; for more information, see below

SetDisplayPreferencesResponse structure

Property	Type	Description
success	int	1 in case of success

Auto attendant configuration

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/AutoAttendant

Methods

get_menu_list

Parameters: GetMenuListRequest
Return value: GetMenuListResponse

This method enables an API user (account) to get a list of all configured auto attendant menus.

update_menu

Parameters: UpdateMenuRequest
Return value: UpdateMenuResponse

This method enables an API user (account) to update the settings of a separate auto attendant menu.

create_menu

Parameters: CreateMenuRequest

Return value: CreateMenuResponse

This method enables an API user (account) to create an auto attendant menu.

del_menu

Parameters: DelMenuRequest
Return value: DelMenuResponse

This method enables an API user (account) to delete an auto attendant menu.

del_menu_transition

Parameters: DelMenuTransitionRequest
Return value: DelMenuTransitionResponse

This method enables an API user to delete an auto attendant menu transition.

set_menu_prompt

Parameters: SetMenuPromptRequest
Return value: SetMenuPromptResponse

This method enables an API user (account) to set (record) separate promt for selected auto attendant menu. The sound file is sent in a MIME attachment to the API request.

get_menu_prompt

Parameters: GetMenuPromptRequest
Return value: GetMenuPromptResponse

This method enables an API user (account) to get a separate prompt from the selected auto attendant menu. The sound file is sent in a MIME attachment to the API request.

get_menu_transition_list

Parameters: GetMenuTransitionListRequest
Return value: GetMenuTransitionListResponse

This method enables an API user (account) to get a list of auto attendant menu transitions.

set_menu_transition

Parameters: SetMenuTransitionRequest

Return value: SetMenuTransitionResponse

This method enables an API user (account) to set auto attendant menu transitions. The transition prompt should be sent in a MIME attachment.

get_menu_transition_prompt

Parameters: GetMenuTransitionPromptRequest
Return value: GetMenuTransitionPromptResponse

This method enables an API user (account) to get an auto attendant menu transition prompt. The prompt is sent in a MIME attachment.

set_menu_transition_promt

Parameters: SetMenuTransitionPromptRequest
Return value: SetMenuTransitionPromptResponse

This method enables an API user to set an auto attendant menu transition prompt. The transition prompt should be sent in a MIME attachment.

Type reference

GetMenuListRequest structure

Property	Type	Description
-	-	-

GetMenuListResponse structure

Property	Type	Description
menu_list	array of MenuInfo structures	The list of auto attendant menus

UpdateMenuRequest structure

Property	Type	Description
menu_info	MenuInfo	Auto attendant menu data

UpdateMenuResponse structure

Property	Type	Description
i_menu	int	The unique ID of updated menu record

CreateMenuRequest structure

Property	Type	Description
menu_info	MenuInfo	Auto attendant menu data

CreateMenuResponse structure

Property	Type	Description
i_menu	int	The unique ID of created menu record

DelMenuRequest structure

Property	Type	Description
i_menu	int	The unique ID of deleted menu record

DelMenuResponse structure

Property	Type	Description
i_menu	int	The unique ID of deleted menu database record

DelMenuTransitionRequest structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record

DelMenuTransitionResponse structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record

MenuInfo structure

Property	Type	Description
i_menu*	int	The unique ID of menu record (required for the update_menu and del_menu methods)
name	string, max 64 chars	**The unique** within one account menu name; 'ROOT' name is reserved for the root menu, which always exists
period	string, max 255 chars	Period in special format (see the **How to Define a Time Period** section of this guide).
period_desc	string, max 255 chars	Description of period in a form understandable by end-users
msg_disabled_type	string	'Unavailable' prompt type – standard or recorded by user. Values: • standard • custom
msg_timeout_type	string	'Timeout' prompt type –

		standard or recorded by user. Values: • standard • custom
msg_intro_set	int	1 if 'Into' prompt recorded; otherwise 0
msg_menu_set	int	1 if 'Menu' prompt recorded; otherwise 0
msg_disabled_set	int	1 if 'Unavailable' prompt recorded; otherwise 0
msg_timeout_set	int	1 if 'Timeout' prompt recorded; otherwise 0
msg_intro_type	string	'Intro' prompt type – standard or recorded by user. Values: • standard • custom
msg_menu_type	string	'Menu' prompt type – standard or recorded by user. Values: • standard • custom
replay_menu_times	int	The number of times to repaly the menu prompts
first_digit_timeout n	int	The timeout in seconds to wait while the first digit is entered
next_digit_timeout	int	The maximum timeout in seconds between collected digits. Default: 5
direct_dial_enabled	string (Y/N)	If set to Y, allow dialing extension from the menu directly. If enabled, the "DirectDial" value for the *action* attribute will be forbidden. Default: N

SetMenuPromptRequest structure

Property	Type	Description
i_menu	int	The unique ID of updated menu record
prompt_type	string	Prompt type: • intro • menu

		• disabled • timeout
prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API request

SetMenuPromptResponse structure

Property	Type	Description
i_menu	int	The unique ID of updated menu record
i_audio_file	int	The ID of the audio file in the CodecConverter conversion queue

GetMenuPromptRequest structure

Property	Type	Description
i_menu	int	The unique ID of menu record
prompt_type	string	Prompt type: • intro • menu • disabled • timeout

GetMenuPromptResponse structure

Property	Type	Description
prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API response

GetMenuTransitionListRequest structure

Property	Type	Description
i_menu	int	The unique ID of menu record

GetMenuTransitionListResponse structure

Property	Type	Description
transition_list	array of TransitionInfo structures	Set of transitions for specified auto attendant menu

SetMenuTransitionRequest structure

Property	Type	Description
i_menu	int	The unique ID of the menu record
transition_info	TransitionInfo	Properties of the menu transition

SetMenuTransitionResponse structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record
i_audio_file	int	The ID of the audio file in the CodecConverter conversion queue

GetMenuTransitionPromptRequest structure

Property	Type	Description
event	string	Transition event; see allowed values in **TransitionInfo** structure
i_menu	int	The unique ID of menu record
i_menu_transition	int	The unique ID of the menu transition record

GetMenuTransitionPromptResponse structure

Property	Type	Description
prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API request

SetMenuTransitionPromptRequest structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record
prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API request

SetMenuTransitionPromptResponse structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record
i_audio_file	int	The ID of the audio file in the CodecConverter conversion queue

TransitionInfo structure

Property	Type	Description
action	string	Performed action.

		Possible values: • Disabled – No action. • Directory – Launch the 'Dial Directory' IVR. • Queue – Transfer to the call queue specified in the *target_i_menu* property. • Transfer – Transfer to the preconfigured number specified in the *destination* property. • TransferE164 – Transfer to the E164 number specified in the *destination* property. • Voicemail – Launch voicemail recording. • Menu – Go to the auto attendant menu specified in *target_i_menu* property. • Extension – Transfer to the extension dialed by a user. Note that at the voice prompt request, the user must input a menu item first and then the extension number. • DISA – Make a call. • DirectDial – Transfer to the extension dialed by a user. Note that the first number of the extension must coincide with the current action digit. • DisconnectCall – Disconnect a call.
announce_ext_numbers	string	Specifies whether to announce the external

		number. Possible values: • Y – Announce the external number. • N – Don not announce the external number.
destination	string, max. 32 chars	Destination for 'Transfer,' 'TransferE164' action
event	string	Transition event. Possible values: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '*', '#', 'Timeout', 'Not Active', 'f'.
i_menu_transition	int	The unique ID of the menu transition record
max_size	int	The maximum allowed number of digits that a user can input as an extension (applicable only for the Extension *action*)
play_prompt	string	Play or do not play user-recorded prompt before action. Possible values: • Y • N
prompt	string	Filename of a user-recorded prompt that is being sent in a MIME attachment (only for the set_menu_transition method)
prompt_set	int	1 if user-recorded prompt set
target_i_menu	int	The unique ID of the auto attendant menu record
target_i_queue	int	The unique ID of the call queue to which the call must be transferred.

Conference configuration

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/Conference

Methods

get_conf_info

Parameters: **GetConfInfoRequest**
Return value: **GetConfInfoResponse**
Realm: account

This method enables an API user to obtain conference settings by i_conf or name.

get_conf_list

Parameters: **GetConfListRequest**
Return value: **GetConfListResponse**
Realm: account

This method enables an API user to obtain a list of all his conferences and their settings.

create_conf

Parameters: **CreateConfRequest**
Return value: **CreateConfResponse**
Realm: account

This method enables an API user to create a new conference entity.

update_conf

Parameters: **UpdateConfRequest**
Return value: **UpdateConfResponse**
Realm: account

This method enables an API user (account) to update a conference entity.

del_conf

Parameters: **DelConfRequest**
Return value: **DelConfResponse**
Realm: account

This method enables an API user to delete a certain conference.

set_conf_prompt

Parameters: SetConfPromptRequest
Return value: SetConfPromptResponse

This method enables an API user (account) to set (record) separate prompts for conferences. The sound file is sent in a MIME attachment to the API request.

get_conf_prompt

Parameters: GetConfPromptRequest
Return value: GetConfPromptResponse

This method enables an API user (account) to get a prompt recorded for a conference. The sound file is sent in a MIME attachment to the API request.

Type reference

ConfInfo structure

Property	Type	Description
i_conf	int	The unique ID for a conference entity
name	string	A conference name
pin_host	string	PIN for administrator to log into the conference
pin_user	string	PIN for user to join the conference
max_call_duration	int	Maximum conference duration
max_session_time	int	Maximum session time (exclusive with start / end time)
max_participants	int	Maximal participants
wait_host	string	Whether the administrator should log in first. Allowed values: Y, N
play_announce	string	Specifies whether announcements should be played. Allowed values: Y, N
play_moh	string	Specifies whether MOH should be played. Allowed values: Y, N
start_time	string	Specifies when the conference will start (For permanent conference use max_session_time). Note that the time is defined in UTC

msg_intro_set	string	1 if an 'Intro' prompt recorded; otherwise 0
moh_set video_conf	string string	1 if a 'MOH' prompt recorded; otherwise 0 Specifies whether video conference is enabled. Allowed values: Y, N

GetConfInfoRequest structure

Property	Type	Description
i_conf	int	The unique ID for the conference
name	string	The conference name

GetConfInfoResponse structure

Property	Type	Description
conf_info	ConfInfo structure	General conference settings

GetConfListRequest structure

Property	Type	Description
-	-	-

GetConfListResponse tructure

Property	Type	Description
conf_list	array of ConfInfo structures	The list of conferences and their settings

CreateConfRequest structure

Property	Type	Description
conf_info	ConfInfo structure	General conference settings

CreateConfResponse structure

Property	Type	Description
i_conf	int	The unique ID for a new conference

UpdateConfRequest structure

Property	Type	Description
conf_info	ConfInfo structure	General conference settings

UpdateConfResponse structure

Property	Type	Description
i_conf	int	The unique ID for the updated conference

DelConfRequest structure

Property	Type	Description
i_conf	int	The unique ID for the conference to be deleted

DelConfResponse structure

Property	Type	Description
i_conf	int	The unique ID for deleted conference

SetConfPromptRequest structure

Property	Type	Description
i_conf	int	The unique ID for a conference record
prompt_type	string	Prompt type: • intro • moh
prompt	string	Filename for a prompt that is being sent in a MIME attachment to the API request

SetConfPromptResponse structure

Property	Type	Description
i_conf	int	The unique ID for the updated conference record
i_audio_file	int	The ID of the audio file in the CodecConverter conversion queue

GetConfPromptRequest structure

Property	Type	Description
i_conf	int	The unique ID for a conference record
prompt_type	string	Prompt type: • intro • moh

GetConfPromptResponse structure

Property	Type	Description
prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API response

3. Call control API

Overview

The Call control API permits to begin, answer and terminate a call, retrieve a list of currently established calls and subscribe to notifications about call state changes for the whole IP Centrex environment as well as for individual extensions. Together with already existing API methods (e.g. to retrieve customer information), these help to build a full-grown CTI solution.

The Call control API is accessible via WebSockets. WebSocket connections are processed by workers. Each worker can process up to 100 concurrent connections. The actual maximum number of connections possible, however, depends on the capacity and general configuration of the Apache server.

Access to JSON-RPC API

All JSON-RPC requests to the API must be sent to the following URL:
wss://<web-server.yourdomain.com>:<port>/ws

Please replace the **web-server.yourdomain.com** with the actual hostname of your web server.

Replace <**port**> with the required port. The JSON-RPC interface is available for administrators on port 443, the interface for customers is available on port 8444, the interface for resellers is available on port 8442 and the interface for accounts is available on port 8445.

Here is an example of the URL the POST request to be sent to:

Sending a JSON-RPC request

For JSON-RPC requests you must include the following parameters in the POST request body:

- `cseq` – Since the WebSocket protocol is asynchronous, this value is used to match the response with the request (the same value is present in the response). If no value is passed in the request, no response is expected and none will be returned.
- `auth_info` – The mandatory authentication information (see the **Security** section).
- `service` – The API service that contains the required method.
- `method` – The name of the required API method.
- `params` – A set of method parameters (in JSON format) that depend on a method structure.

The example below illustrates the login request:

```
{
    "cseq": 2,
"service": "Session",
"method": "login",
"params": {
"login": "demo",
"password": "p@ssw0rd"
    }
}
```

The response contains the session ID value:

```
{
    "cseq":2,
"result":{ "session_id":"07491b94c0025f464c388ff387c7265a"
         },
    "success":1
}
```

Error handling

In case a request could not be executed or had errors in its structure, the response contains the following error information:

- **code**, that is intended for use by the client software and provides an algorithmic mechanism for identifying the fault.
- **message**, that provides a human readable explanation of the fault, and is not intended for algorithmic processing.
- **details**, that complement the **message** and contain the erroneous object.

Error codes are listed in the table below:

Service	Error code	Error message	Details
CallControl	sip.unsupported_method	Unsupported method	The specified method name is unknown
CallControl	sip.wrong_parameters	Bad parameters	Incorrect parameter list - required parameters are missing or contain incorrect values
CallControl	sip.internal_server_error	Internal processing	Unspecified processing

		error	error prevents the correct execution of a method
CallControl	sip.call_not_found	Not found	A call (call part) specified by session_id (dialog_id) was not found
CallControl	sip.call_control_disabled	Disabled	The used method is disabled
CallControl	sip.service_not_enabled	The service is not enabled	The conferencing service is not enabled for this account
all	internal_error	Internal server error	

Call state notification management

URL (namespace): wss://portabilling-web.yourdomain.com/ws/CallControl

These methods enable an agent to monitor calls in progress (outgoing and incoming) and receive notifications about call state changes. This helps in manipulating calls (e.g. redirect the call to another party if the extension is busy).

Use these methods together with the **Voice API** methods to build your CTI solutions such as attendant console or click-to-dial application for your CRM system.

The call states can be monitored for:

- an individual extension. It is represented as an account in PortaSwitch® and is identified by the `i_account` key;
- an entire IP Centrex environment. It is represented as a customer in PortaSwitch® and is identified by the `i_customer` key;
- an access number of your custom IVR application. It is identified by the `i_ivr_an` key.
- an external number. Subscribe the app (e.g. a switchboard app) to call state notifications for the account within your IP Centrex in the usual way. When an account transfers, forwards or makes a

call, the app receives call state notifications from external numbers via that account's channel.

To retrieve the internal ID value of either entity, use **PortaBilling API**. For example, to find `i_account` of an office extension you wish to receive notifications for, call the `get_account_list` API method:

```
{
    "cseq": 2,
    "service": "Account",
    "method": "get_account_list",
    "auth_info":{
      "session_id":"b11c226be16aa179b0d6b2fa0fd1394c"
      },
      "params": {
     "id":"12065558954"
    }
}
```

You can find the information about required PortaBilling® API methods in the descriptions to the method attributes.

Methods

enable_api_notifications

This method enables an agent to subscribe and receive call state notifications for:

- an individual extension by passing the i_account value in the API request;
- the entire IP Centrex environment by passing the i_customer value in the API request;
- the main or a branch office within the IP Centrex environment. This applies when a company has independent offices (i.e. branches) linked to the main one. Such offices are identified by the i_main_office value.
- the access number of your custom IVR application by passing the i_ivr_an value in the API request.

When a call state changes, the sip.call_control_notifications event is sent. It contains the current call state.

Parameters: **EnableApiNotificationsRequest**
Return value: **EnableApiNotificationsResponse**
Realm: administrator, reseller, retail customer, account

Request example:

```
{
    "cseq": 2,
    "auth_info": {
        "session_id": "f9d7eab82631b385fbecff9b65883076"
    },
    "service": "CallControl",
    "method": "enable_api_notifications",
    "params": {
        "event": "sip.call_control_notifications",
        "i_customer": 30
    }
}
```


Response example:


```
{
    "cseq":2,
    "result":{
        "success":1
    },
    "success":1
}
```


Example of a sip.call_control_notifications event


```
{
    "action":"update",
    "event":"sip.call_control_notifications",
    "result":{
        "call_info":{
            "call":{
                "id":"b0fc4957-6fc3a86f@192.168.233.134",
                "tag":"qdef43kz9zxym4lz.o"
            },
            "callee":{
                "account_id":"123007",
                "centrex_id":"30",
                "display_id":"123007",
                "forwarder_list":[

                ],
                "id":"123007"
            },
            "caller":{
                "account_id":"123002",
                "centrex_id":"30",
                "display_id":"123002",
                "forwarder_list":[

                ],
                "id":"123002"
            },
            "duration":10,
            "reason":"Call terminated by API",
            "reason_code":null,
            "state":"terminated",
            "type":"incoming"
        }
    }
}
```

Example of a sip.call_control_notifications event for auto attendant account.


```
{
        "connect_time": "2018-12-28 13:12:25.245",
        "dialog_id": [
            "7e033351-4e65ab85@192.168.64.196",
            "112347"
        ],
        "direction":
"recipient",
        "local_party": {
            "account_id": "123010",
            "centrex_id": "30",
            "display_id": "123010",
            "forwarder_list":[
     {
    "account_id": "123010",
     "centrex_id": "30",
     "display_id": "123010",
     "forwarders": [],
     "id": "123010"
     }
    ],
            "id": "123010",
            "net": "0"
        },
        "queue_info": {
            "i_c_queue": 25,
            "operators": 2,
            "position": 0
        },
        "remote_party": {
            "account_id": "11198700001",
            "centrex_id": "10",
            "display_id": "11198700001",
            "forwarders": [],
            "id": "11198700001",
            "name": "11198700001",
            "net": "0"
        },
        "session_id": 192123,
        "sip_transport_id": "192.168.67.30:5070",
        "start_time": "2018-12-28 13:12:25.146",
        "state": "queued"
    }
```


Example of a sip.call_control_notifications event for an account that forwards a call to an external number.


```
{
"action":"update",
"event":"sip.call_control_notifications",
"result":{
"call_info":{
"call":{
"id":"b0fc4957-6fc3a86f@192.168.233.134",
"tag":"qdef43kz9zxym4lz.o"
},
"callee":{
"account_id":"123007",
```

```
"centrex_id":"30",
"display_id":"123007",
"forwarder_list":[
{
    "account_id": "123008",
     "centrex_id": "30",
     "display_id": "123008",
     "forwarders": [],
     "id": "123008"
},
{
"account_id": "123009",
"centrex_id": "30",
"display_id": "123009",
"forwarders": [],
    "id": "123009"
}
],
"id":"123007"
},
"caller":{
"account_id":"123002",
"centrex_id":"30",
"display_id":"123002",
"forwarder_list":[

],
"id":"123002"
},
"duration":10,
"reason":"Call terminated by API",
"reason_code":null,
"state":"terminated",
"type":"incoming"
}
}
```


disable_api_notifications

This method enables an agent to unsubscribe from call state notifications for:

- an individual extension by passing the i_account value in the API request;
- the entire IP Centrex environment by passing the i_customer value in the API request;
- the main or a branch office within the IP Centrex environment. This applies when a company has independent offices (i.e. branches) linked to the main one. Such offices are identified by the i_main_office value.
- the access number of an IVR application by passing the i_ivr_an value in the API request.

Parameters: **DisableApiNotificationsRequest**
Return value: **DisableApiNotificationsResponse**
Realm: administrator, reseller, retail customer, account

Request example:

```
{
    "cseq": 2,
    "auth_info": {
        "session_id": "f9d7eab82631b385fbecff9b65883076"
    },
    "service": "CallControl",
    "method": "disable_api_notifications",
    "params": {
        "event": "sip.call_control_notifications",
        "i_customer": 30
    }
}
```

Response example:

```
{
   "cseq":2,
   "result":{
      "success":1
   },
   "success":1
}
```

Type reference

EnableApiNotificationsRequest structure

The request must contain at least one attribute that is mentioned in the structure below.

Property	Type	Description
i_account	unsignedLong	The unique ID of the account record. The account represents a phone line or an office extension. To get the account ID, call the **get_account_list** method. The *i_account* is returned in the AccountInfo structure. See **https://www.portaone.com/docs/PortaBilling_API.html#AccountInfo**.
i_ivr_an	unsignedLong	The unique ID of the access number associated with the custom IVR application (it has the User application type on the PortaBilling® web interface). In PortaBilling® the IVR access

		number is also associated with the account record. This is required to apply charges for using this number. Call the **obtain_access_number** API method to assign the access number to the account and receive its ID in the response. See **https://www.portaone.com/docs/PortaBilling_API.html** and **https://www.portaone.com/docs/PortaBilling_API.html#ObtainAccessNumberResponse**. If you operate under the customer, reseller or the administrator realm, first retrieve the *i_account* to which you wish to assign the access number and pass it in the API request.
i_customer	unsignedLong	The unique ID of the customer record. To get the customer ID, call the **get_customer_info** method. The *i_customer* is returned in the CustomerInfo structure. See **https://www.portaone.com/docs/PortaBilling_API.html#CustomerInfo**.
i_main_office	unsignedLong	The unique ID of the main office (customer record with office type 3). To get the main office ID and headquarters office type, call the **get_customer_info** method. Possible values for the *i_office_type* attribute in the response are the following: • 1 – none • 2 – branch office • 3 – main office If the office type is 1 (none), leave this attribute empty. If the office type is 2 (branch office), specify the office ID from the

		i_main_office attribute. If the office type is 3 (main office), specify the main office ID from the *i_customer* attribute. See **https://www.portaone.com/docs/PortaBilling_API.html#CustomerInfo**.

EnableApiNotificationsResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

DisableApiNotificationsRequest structure

The request must contain at least one attribute that is mentioned in the structure below.

Property	Type	Description
i_account	unsignedLong	The unique ID of the account record. The account represents a phone line or an office extension. To get the account ID, call the **get_account_list** method. The *i_account* is returned in the AccountInfo structure. See **https://www.portaone.com/docs/PortaBilling_API.html#AccountInfo**.- *If you operate under Customer*
i_ivr_an	unsignedLong	The unique ID of the access number associated with the custom IVR application (it has the User application type on the PortaBilling® web interface). In PortaBilling® the IVR access number is also associated with the account record. This is required to apply charges for using this number. Call the **obtain_access_number** API method to assign the access number to the account and receive its ID in the response.

		See **https://www.portaone.com/docs/PortaBilling_API.html** and **https://www.portaone.com/docs/PortaBilling_API.html#ObtainAccessNumberResponse**. If you operate under the customer, reseller or the administrator realm, first retrieve the *i_account* to which you wish to assign the access number and pass it in the API request.
i_customer	unsignedLong	The unique ID of the customer record. To get the customer ID, call the **get_customer_info** method. The *i_customer* is returned in the CustomerInfo structure. See **https://www.portaone.com/docs/PortaBilling_API.html#CustomerInfo**.
i_main_office	unsignedLong	The unique ID of the main office (customer record with office type 3). To get the main office ID and headquarters office type, call the **get_customer_info** method. Possible values for the *i_office_type* attribute in the response are the following: • 1 – none • 2 – branch office • 3 – main office If the office type is 1 (none), leave this attribute empty. If the office type is 2 (branch office), specify the office ID from the *i_main_office* attribute. If the office type is 3 (main office), specify the main office ID from the *i_customer* attribute. See **https://www.portaone.com/docs/Porta**

		Billing_API.html#CustomerInfo.

DisableApiNotificationsResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

Voice API

URL (namespace): wss://portabilling-web.yourdomain.com/ws/CallControl

These API methods enable an agent to control call flow so that they can make, receive or redirect calls from their applications or web browsers.

The agent is represented as an account in PortaSwitch® and is charged for making calls.

Methods

get_sip_calls_list

This method enables an agent to receive a list of calls in progress for an individual extension or for the whole IP Centrex environment. For this, subscribe to API notifications using the **enable_api_notifications** method.

Parameters: **GetSipCallsListRequest**
Return value: **GetSipCallsListResponse**
Realm: administrator, reseller, retail customer, account
Standalone mode support: Yes

Request example:

```
{
   "cseq":2,
   "auth_info":{
      "session_id":"ba6596e4b60919f8695033a20519d6af"
   },
   "service":"CallControl",
   "method":"get_sip_calls_list",
   "params": {
        "i_customer": 30
    }
}
```

Response example:

```
{
    "cseq":2,
    "result":{
        "calls_list":[
            {
                "call":{
                    "id":"30108b5e-b29bdab0@192.168.233.134",
                    "tag":"ba6783868a4100a8o1"
                },
                "callee":{
                    "account_id":"123007",
                    "display_id":"123007",
                    "forwarder_list":[

                    ],
                    "id":"123007"
                },
                "caller":{
                    "account_id":"123002",
                    "centrex_id":"30",
                    "display_id":"123002",
                    "forwarder_list":[

                    ],
                    "id":"123002"
                },
                "start_time":"2018-11-29 13:41:04",
                "state":"ringing",
                "type":"outgoing"

            },
            {
                "call":{
                    "id":"30108b5e-b29bdab0@192.168.233.134",
                    "tag":"qdkr5cjc9cfyzxyb.o"
                },
                "callee":{
                    "account_id":"123007",
                    "centrex_id":"30",
                    "display_id":"123007",
                    "forwarder_list":[

                    ],
                    "id":"123007"
                },
                "caller":{
                    "account_id":"123002",
                    "centrex_id":"30",
                    "display_id":"123002",
                    "forwarder_list":[

                    ],
                    "id":"123002"
                },
                "start_time":"2018-11-29 13:41:04",
                "state":"ringing",
                "type":"incoming"
            }
        ]
    },
    "success":1
```

```
}
```

originate_advanced_call

This method enables an agent to initiate a callback call to a phone number or an extension and then connect it with the desired destination.

The number to which the callback should be established is specified in the **caller_id** attribute. The destination number is defined in the **callee_id** attribute. The **bill_id** attribute contains the ID of the agent's account in PortaSwitch® to charge for this call.

PortaSIP® first places a call to the **caller_id** destination according to the routing plan (leg A). When the first UA answers the call, PortaSIP® places a second call to the destination specified as **callee_id** (leg B).

For calls to go through, the product configuration for the account specified in the **bill_id** attribute must include the rating entry with the INCOMING access code.

Parameters: **OriginateAdvancedCallRequest**
Return value: **OriginateAdvancedCallResponse**
Realm: administrator, reseller, retail customer, account

Request example:

```
{
   "cseq":3,
   "auth_info":{
      "session_id":"48e8eedee6a6a520bd5e0380b83ffeb7"
   },
   "service":"CallControl",
   "method":"originate_advanced_call",
   "params":{
      "bill_id":"123007",
      "callee_id":"123003",
      "caller_id":"123007"
   },
}
```

Response example:

```
{
   "cseq":3,
   "result":{
      "call":{
         "id":"BBfc!rA3EV9JNsDxdhRoPvXGW4xR@192.168.243.133"
      },
      "success":1
   },
   "success":1
}
```

Notification example for call leg A:

```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"BBfc!rA3EV9JNsDxdhRoPvXGW4xR@192.168.243.133
",
            "tag":"tz+d-zju2fxikkor.o"
         },
         "callee":{
            "account_id":"123007",
            "centrex_id":"30",
            "display_id":"123007",
            "forwarder_list":[

            ],
            "id":"123007"
         },
         "caller":{
            "display_id":"123003",
            "forwarder_list":[

            ],
            "id":"123003"
         },
         "start_time":"2018-12-03 13:43:01",
         "state":"trying",
         "type":"incoming"
      }
   }
}

{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"BBfc!rA3EV9JNsDxdhRoPvXGW4xR@192.168.243.133
",
            "tag":"tz+d-zju2fxikkor.o"
         },
         "callee":{
            "account_id":"123007",
            "centrex_id":"30",
            "display_id":"123007",
            "forwarder_list":[

            ],
            "id":"123007"
         },
         "caller":{
            "display_id":"123003",
            "forwarder_list":[

            ],
            "id":"123003"
         },
         "start_time":"2018-12-03 13:43:01",
         "state":"ringing",
```

```
            "type":"incoming"
         }
      }
   }

   {
      "action":"update",
      "event":"sip.call_control_notifications",
      "result":{
         "call_info":{
            "call":{
               "id":"BBfc!rA3EV9JNsDxdhRoPvXGW4xR@192.168.243.133
   ",
               "tag":"tz+d-zju2fxikkor.o"
            },
            "callee":{
               "account_id":"123007",
               "centrex_id":"30",
               "display_id":"123007",
               "forwarder_list":[

               ],
               "id":"123007"
            },
            "caller":{
               "display_id":"123003",
               "forwarder_list":[

               ],
               "id":"123003"
            },
            "connect_time":"2018-12-03 13:43:04",
            "state":"connected",
            "type":"incoming"
         }
      }
   }
```


Notification example for call leg B:


```
   {
      "action":"update",
      "event":"sip.call_control_notifications",
      "result":{
         "call_info":{
            "call":{
               "id":"BBfc!rA3EV9JNsDxdhRoPvXGW4xR@192.168.243.133
   ~1o",
               "tag":"tz+d-zju2fw2o4ws.o"
            },
            "callee":{
               "account_id":"123003",
               "centrex_id":"30",
               "display_id":"123003",
               "forwarder_list":[

               ],
               "id":"123003"
            },
            "caller":{
               "account_id":"123007",
```

```
            "centrex_id":"30",
            "display_id":"123007",
            "forwarder_list":[

            ],
            "id":"123007"
         },
         "start_time":"2018-12-03 13:43:05",
         "state":"trying",
         "type":"incoming"
      }
   }
}

{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"BBfc!rA3EV9JNsDxdhRoPvXGW4xR@192.168.243.133
~1o",
            "tag":"tz+d-zju2fw2o4ws.o"
         },
         "callee":{
            "account_id":"123003",
            "centrex_id":"30",
            "display_id":"123003",
            "forwarder_list":[

            ],
            "id":"123003"
         },
         "caller":{
            "account_id":"123007",
            "centrex_id":"30",
            "display_id":"123007",
            "forwarder_list":[

            ],
            "id":"123007"
         },
         "start_time":"2018-12-03 13:43:05",
         "state":"ringing",
         "type":"incoming"
      }
   }
}

{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"BBfc!rA3EV9JNsDxdhRoPvXGW4xR@192.168.243.133
~1o",
            "tag":"tz+d-zju2fw2o4ws.o"
         },
         "callee":{
            "account_id":"123003",
```

```
                "centrex_id":"30",
                "display_id":"123003",
                "forwarder_list":[

                ],
                "id":"123003"
             },
             "caller":{
                "account_id":"123007",
                "centrex_id":"30",
                "display_id":"123007",
                "forwarder_list":[

                ],
                "id":"123007"
             },
             "connect_time":"2018-12-03 13:43:08",
             "state":"connected",
             "type":"incoming"
          }
       }
    }
```

answer_call

This method enables an agent to answer incoming calls using the application instead of picking up a handset.

Parameters: AnswerCallRequest
Return value: AnswerCallResponse
Realm: administrator, reseller, retail customer, account

Request example:

```
{
   "cseq":2,
   "auth_info":{
      "session_id":"9dc0afbab375071f4d132fa82502025c"
   },
   "service":"CallControl",
   "method":"answer_call",
   "params":{
      "call":{
         "id":"30108b5e-b29bdab0@192.168.233.134",
         "tag":"qdkr5cjc9cfyzxyb.o"
      },
      "callee_answer_mode":"notify"
   }
}
```

Response example:

```
{
   "cseq":2,
   "result":{
      "success":1
   },
   "success":1
```

```
}
```


Notification example from the party that answers the call


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"30108b5e-b29bdab0@192.168.233.134",
            "tag":"qdkr5cjc9cfyzxyb.o"
         },
         "callee":{
            "account_id":"123007",
            "centrex_id":"30",
            "display_id":"123007",
            "forwarder_list":[

            ],
            "id":"123007"
         },
         "caller":{
            "account_id":"123002",
            "centrex_id":"30",
            "display_id":"123002",
            "forwarder_list":[

            ],
            "id":"123002"
         },
         "connect_time":"2018-11-29 13:41:25",
         "state":"connected",
         "type":"incoming"
      }
   }
}
```


terminate_call

This method enables an agent to disconnect a call from the application.

Parameters: **TerminateCallRequest**
Return value: **TerminateCallResponse**
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq":2,
   "auth_info":{
      "session_id":"9dc0afbab375071f4d132fa82502025c"
   },
   "service":"CallControl",
   "method":"terminate_call",
   "params":{
      "call":{
         "id":"b0fc4957-6fc3a86f@192.168.233.134",
```

```
            "tag":"qdef43kz9zxym4lz.o"
         },
      }
   }
```

Response example:

```
   {
      "cseq":2,
      "result":{
         "success":1
      },
      "success":1
   }
```

Notification example from the terminated party:

```
   {
      "action":"update",
      "event":"sip.call_control_notifications",
      "result":{
         "call_info":{
            "call":{
               "id":"b0fc4957-6fc3a86f@192.168.233.134",
               "tag":"qdef43kz9zxym4lz.o"
            },
            "callee":{
               "account_id":"123007",
               "centrex_id":"30",
               "display_id":"123007",
               "forwarder_list":[

               ],
               "id":"123007"
            },
            "caller":{
               "account_id":"123002",
               "centrex_id":"30",
               "display_id":"123002",
               "forwarder_list":[

               ],
               "id":"123002"
            },
            "duration":10,
            "reason":"Call terminated by API",
            "reason_code":null,
            "state":"terminated",
            "type":"incoming"
         }
      }
   }
```

hold_call

This method enables an agent to put a call on hold from the application, without dialing the Hold key on the UA.

The UA must support NOTIFY request with "Event: hold" (see BroadWorks Remote Control Talk Event Package) to initiate hold.

If UA doesn't support the event package, it replies with 400, 489 error code. In this case PortaSIP® places both call parties on hold and plays its own MOH.

Parameters: **HoldCallRequest**
Return value: **HoldCallResponse**
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq":2,
   "auth_info":{
      "session_id":"9dc0afbab375071f4d132fa82502025c"
   },
   "service":"CallControl",
   "method":"hold_call",
   "params":{
      "call":{
         "id":"402a8bb-e4e23269@192.168.233.134",
         "tag":"3481849322414871o1"
      },
   }
}
```


Response example:


```
{
   "cseq":2,
   "result":{
      "success":1
   },
   "success":1
}
```


Notification example from the party that places the call on hold:


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"402a8bb-e4e23269@192.168.233.134",
            "tag":"3481849322414871o1"
         },
         "callee":{
            "account_id":"123007",
            "centrex_id":"30",
            "display_id":"123007",
            "forwarder_list":[

            ],
            "id":"123007"
```

```
            },
            "caller":{
               "account_id":"123002",
               "centrex_id":"30",
               "display_id":"123002",
               "forwarder_list":[

               ],
               "id":"123002"
            },
            "state":"holding",
            "type":"outgoing"
         }
      }
   }
```


Notification example from the party that is placed on hold:


```
   {
      "action":"update",
      "event":"sip.call_control_notifications",
      "result":{
         "call_info":{
            "call":{
               "id":"402a8bb-e4e23269@192.168.233.134",
               "tag":"6zje7kwsqio4htum.o"
            },
            "callee":{
               "account_id":"123007",
               "centrex_id":"30",
               "display_id":"123007",
               "forwarder_list":[

               ],
               "id":"123007"
            },
            "caller":{
               "account_id":"123002",
               "centrex_id":"30",
               "display_id":"123002",
               "forwarder_list":[

               ],
               "id":"123002"
            },
            "connect_time":"2018-11-29 14:20:39",
            "state":"held",
            "type":"incoming"
         }
      }
   }
```


unhold_call

This method enables an agent to release a call from hold from the application. The UA must support NOTIFY request with "Event: talk" (see BroadWorks Remote Control Talk Event Package) to initiate unhold.

If the UA does not support this event package, it replies with 400, 489 error code. PortaSIP® unholds both call parties and stops playing its own MOH.

Parameters: **UnholdCallRequest**
Return value: **UnholdCallResponse**
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq":2,
   "auth_info":{
      "session_id":"9dc0afbab375071f4d132fa82502025c"
   },
   "service":"CallControl",
   "method":"unhold_call",
   "params":{
      "call":{
         "id":"402a8bb-e4e23269@192.168.233.134",
         "tag":"34818493224148710l"
      },
   }
}
```


Response example:


```
{
   "cseq":2,
   "result":{
      "success":1
   },
   "success":1
}
```


Notification example from the released party:


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"402a8bb-e4e23269@192.168.233.134",
            "tag":"34818493224148710l"
         },
         "callee":{
            "account_id":"123007",
            "centrex_id":"30",
            "display_id":"123007",
            "forwarder_list":[

            ],
            "id":"123007"
         },
         "caller":{
            "account_id":"123002",
            "centrex_id":"30",
```

```
            "display_id":"123002",
            "forwarder_list":[

            ],
            "id":"123002"
         },
         "connect_time":"2018-11-29 15:07:18",
         "state":"connected",
         "type":"outgoing"
      }
   }
}
```


transfer_call

This method enables an agent to perform blind call transfer to another extension or an external number from the application.

Parameters: TransferCallRequest
Return value: TransferCallResponse
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq":2,
   "auth_info":{
      "session_id":"9dc0afbab375071f4d132fa82502025c"
   },
   "service":"CallControl",
   "method":"transfer_call",
   "params":{
      "call":{
         "id":"9fb147fc-4f045471@192.168.233.134",
         "tag":"9dwxscwxpshymxri.o"
      },
      "cld": "123003"
   }
}
```


Response example:


```
{
   "cseq":2,
   "result":{
      "success":1
   },
   "success":1
}
```


Notification example for the transferring party to be terminated:


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
```

```
            "call":{
               "id":"9fb147fc-4f045471@192.168.233.134",
               "tag":"9dwxscwxpshymxri.o"
            },
            "callee":{
               "account_id":"123007",
               "centrex_id":"30",
               "display_id":"123007",
               "forwarder_list":[

               ],
               "id":"123007"
            },
            "caller":{
               "account_id":"123002",
               "centrex_id":"30",
               "display_id":"123002",
               "forwarder_list":[

               ],
               "id":"123002"
            },
            "duration":5,
            "reason":"blind transfer",
            "reason_code":null,
            "state":"terminated",
            "type":"incoming"
         }
      }
   }
```


Notification example for the transfer target:


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"9fb147fc-4f045471@192.168.233.134",
            "tag":"9dwxscwxpshymuf6.o"
         },
         "callee":{
            "account_id":"123003",
            "centrex_id":"30",
            "display_id":"123003",
            "forwarder_list":[

            ],
            "id":"123003"
         },
         "caller":{
            "account_id":"123002",
            "centrex_id":"30",
            "display_id":"123002",
            "forwarder_list":[

            ],
            "id":"123002"
         },
         "start_time":"2018-11-29 15:33:05",
```

```
            "state":"trying",
            "type":"incoming"
         }
      }
   }

   {
      "action":"update",
      "event":"sip.call_control_notifications",
      "result":{
         "call_info":{
            "call":{
               "id":"9fb147fc-4f045471@192.168.233.134",
               "tag":"9dwxscwxpshymuf6.o"
            },
            "callee":{
               "account_id":"123003",
               "centrex_id":"30",
               "display_id":"123003",
               "forwarder_list":[

               ],
               "id":"123003"
            },
            "caller":{
               "account_id":"123002",
               "centrex_id":"30",
               "display_id":"123002",
               "forwarder_list":[

               ],
               "id":"123002"
            },
            "connect_time":"2018-11-29 15:33:08",
            "state":"connected",
            "type":"incoming"
         }
      }
   }
```


Notification example for the transferee:


```
   {
      "action":"update",
      "event":"sip.call_control_notifications",
      "result":{
         "call_info":{
            "call":{
               "id":"9fb147fc-4f045471@192.168.233.134",
               "tag":"e8592a80272833bdo1"
            },
            "callee":{
               "account_id":"123003",
               "centrex_id":"30",
               "display_id":"123003",
               "forwarder_list":[

               ],
               "id":"123003"
```

```
            },
            "caller":{
               "account_id":"123002",
               "centrex_id":"30",
               "display_id":"123002",
               "forwarder_list":[

               ],
               "id":"123002"
            },
            "connect_time":"2018-11-29 15:33:00",
            "state":"connected",
            "type":"outgoing"
         }
      }
   }
```

join_calls

This method enables an agent to join calls when performing attended call transfer.

PortaSIP® disconnects the transferor identified by the **call** and **to_call** attributes in the call dialogs established with it and joins the remaining parties.

Parameters: JoinCallsRequest
Return value: JoinCallsResponse
Realm: administrator, reseller, retail customer, account

Request example:

```
{
   "cseq":4,
   "auth_info":{
      "session_id":"e6816cb2c29d4bc9db622f5ddc0796fe"
   },
   "service":"CallControl",
   "method":"join_calls",
   "params":{
      "call":{
         "id":"1c4cef19-e035cfcd@192.168.233.134",
         "tag":"670946a915d2dd7do1"
      },
      "to_call":{
         "id":"hDSjA!Ei3bMIfb6OxfLndUwXqqF9@192.168.243.133~1o
",
         "tag":"kc3kqksjw670th8h.o"
      },
   },
}
```

Response example:

```
{
   "cseq":4,
   "result":{
```

```
        "success":1
    },
    "success":1
}
```

Type Reference

GetSipCallsListRequest structure

Property	Type	Description
i_account	unsignedLong	The unique ID of the account record. The account represents a phone line or an office extension. To get the account ID, call the **get_account_list** method. The *i_account* is returned in the AccountInfo structure. See **https://www.portaone.com/docs/PortaBilling_API.html#AccountInfo**.
i_main_office	unsignedLong	The unique ID of the main office (customer record with office type 3). To get the main office ID and headquarters office type, call the **get_customer_info** method. Possible values for the *i_office_type* attribute in the response are the following: • 1 – none • 2 – branch office • 3 – main office If the office type is 1 (none), leave this attribute empty. If the office type is 2 (branch office), specify the office ID from the *i_main_office* attribute. If the office type is 3 (main office), specify the main office ID from the *i_customer* attribute. See **https://www.portaone.com/docs/PortaBilling_API.html#CustomerInfo**.

i_customer	unsignedLong	The unique ID of the customer record. To get the customer ID, call the **get_customer_info** method. The *i_customer* is returned in the CustomerInfo structure. See **https://www.portaone.com/docs/PortaBilling_API.html#CustomerInfo**.
i_ivr_an	unsignedLong	The unique ID of the access number associated with the custom IVR application (it has the User application type on the PortaBilling® web interface). In PortaBilling® the IVR access number is also associated with the account record. This is required to apply charges for using this number. Call the **obtain_access_number** API method to assign the access number to the account and receive its ID in the response. See **https://www.portaone.com/docs/PortaBilling_API.html** and **https://www.portaone.com/docs/PortaBilling_API.html#ObtainAccessNumberResponse**. If you operate under the customer, reseller or the administrator realm, first retrieve the *i_account* to which you wish to assign the access number and pass it in the API request.

GetSipCallsListResponse structure

Property	Type	Description
calls_list	Array of **SIPCallInfo** structure	The list of calls

SIPCallInfo structure

Property	Type	Description
Common		
call	SipCallIdentifier structure	The unique ID of separate parts of the call.
callee	SipCallerInfo structure	The information about who the caller is calling to
caller	SipCallerInfo structure	The information about the caller.
state	string	Defines the current state of the call. Possible values: • **trying** – a call is initiated and an outgoing request is sent; • **ringing** – a phone is ringing; • **early** – early media is played; • **terminated** – a call is disconnected; • **connected** – a call is answered / taken-from-hold and the remote side is connected; • **held** – a call party is connected and is put on hold. This state is returned to the party which is placed on hold. • **holding** - call part is connected and is put on hold. This state is returned to the party who places the call on hold. • **queued** – a caller is placed to a call queue; • **dequeued** - a caller is

		removed from the call queue. • **started** – the conference call has started • **terminated** – the conference call has been terminated
transport_id	string	The SIP address of the call defined in the format IP:port
transfer_success	int	A non-zero value means that the call transfer has been successfully performed.
type	string	Defines the type of the call dialog direction in terms of UA. Possible values: • **outgoing** - outgoing call dialog. UA is initiator. • **incoming** - incoming call dialog. UA is recipient.
State specific information		
connect_time	dateTime	The date and time when the call was connected or put on hold. It is sent for "**connected**", "**held**", "**holding**" call states.
duration	int	The call length (in seconds) from the moment when the call was connected. It is sent for "**terminated"** call state.
reason	string	Describes the reason for terminated or not established calls. Is sent for "**terminated**" call state.
queue_info	SipCallQueueStateInfo structure	The information about a call queue status. Is sent for "queued" and "**dequeued**" call states.
reason_code	int	The code of the reason the calls ended. Is sent for "**terminated**" state.
start_time	dateTime	The date and time when the call was initiated. Is sent for "**trying**", "**ringing**", "**early**", "**started**" call states.

terminate_time	dateTime	The date and time when the call was finished. Ii is sent for "**terminated**" call states
IVR notification specific, should never happen in regular API calls		
digit	string	The detected DTMF digit. Possible values: • "0" • "1" • "2" • "3" • "4" • "5" • "6" • "7" • "8" • "9" • "*" • "#"
dtmf_duration	int	DTMF duration in milliseconds
event	string	The event related to asynchronous IVR notification. • **play_prompt_completed** – prompt playback finished; • **dtmf_digit_detected** – single DTMF digit received.
order	int	The prompt playback order (2 or 3). Represents relative position of one prompt to another (if any) in playback stack. Possible values:

		• "2" • "3"

SipCallerInfo structure

SipCallerInfo structure reflects information passed in the `PortaOne-Calling-Party`, `PortaOne-Redirecting-Party`, `PortaOne-Called-Party` RADIUS attributes. See the **External System Interfaces Guide** for details.

Property	Type	Description
account_id	string	The phone number (PIN) of the party who is making the call. It is represented as an account in PortaSwitch® and is unique in the environment associated with call participant. See ID in **https://www.portaone.com/docs/PortaBilling_API.html#AccountInfo**.
centrex_id	string	The ID of the IP Centrex environment the party who is making the call belongs to. It is represented as a customer (the main office) in PortaSwitch®. See **https://www.portaone.com/docs/PortaBilling_API.html#CustomerInfo**
display_id	string	The display number provided by the callee/caller. It is taken from the `From` header and typically displayed on the called party's phone display.
display_name	string	The display name provided by the callee/caller. It is taken from the `CLN` field.
extension_id	string	The extension number configured on the PBX / within the IP Centrex environment and associated with the phone line (account ID).
huntgroup_id	string	The huntgroup number, on behalf of which the call happened.
id	string	The phone number of the calling / called party depending on whether it is an incoming or an outgoing call
forwarder_list	Array of SipForwarderAccountI	The list of account IDs that initiated the forward.

	nfo structure	
access_number	string	The IVR access number that receives the call.

SipCallIdentifier structure

Property	Type	Description
tag	string	The call remote tag
id	string	The unique call identifier

SipCallQueueStateInfo structure

Property	Type	Description
i_c_queue	usingnedLong	The unique ID of a call queue record. To retrieve the *i_c_queue*, call the **get_callqueue_list** method. See **https://www.portaone.com/docs/PortaBilling_API.html#CQInfo**
position	int	A caller's position in the queue.
operators	int	The number of active operators.

SipForwarderAccountInfo structure

Property	Type	Description
id	string	The phone number of the party who initiated the forward.
display_id	string	The Display ID provided by the forwarder.
display_name	string	The display name.
centrex_id	string	The Centrex identifier.
extension_id	string	The extension ID.
huntgroup_id	string	The ID of the huntgroup involved in establishing the call.

AnswerCallRequest structure

Property	Type	Description
callee_answer_mode	string	Specifies the exact method to use for call answering. Possible values: • Notify – PortaSIP® sends NOTIFY request with "Event: talk" (see BroadWorks Remot

		e Control Talk Event Package). • Invite – PortaSIP® reconnects the call using new dialog and "auto-answer" header for INVITE: `Alert-Info: <sip:127.0.0.1>;info=AutoAnswer Call-Info: <sip:127.0.0.1>;answer-after=0`
call	SipCallIdentifier structure	The unique IDs of separate parts of the call.

AnswerCallResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed.

TerminateCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The unique IDs of separate parts of the call

TerminateCallResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

HoldCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The unique ID of separate parts of the call.

HoldCallResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed.

UnholdCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The unique ID of separate parts of the call

UnholdCallResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed.

TransferCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The unique ID of the call party that initiates transfer (acts as transferor)
cld	string	The phone number of the transfer target
sip_headers	SipHeaderInfo structure	The set of SIP headers.

TransferCallResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

SipHeaderInfo structure

Property	Type	Description
name	string	The header name.
value	string	The header value.

JoinCallsRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The identifier of the call party to be joined
to_call	SipCallIdentifier structure	The identifier of the call party to be joined with

JoinCallsResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed.

OriginateAdvancedCallRequest structure

Property	Type	Description
bill_id	string	The account number to be charged for a call

callee_auto_pickup	string	This flag indicates whether to request auto-answer functionality from the caller's user agent. When enabled, PortaSIP® sends the INVITE request to establish the call, which contains "auto-answer" header fields:
		
		`Alert-Info: <sip:127.0.0.1>;info=AutoAnswer`
		`Call-Info: <sip:127.0.0.1>;answer-after=0`
		
		The UA should support this functionality.
		
		Possible values:
		• Y - enable auto-answer functionality;
		• N - disable auto-answer functionality (default value).

callee_id	string	The phone number to be called
caller_id	string	The phone number of the calling party

OriginateAdvancedCallResponse structure

Property	Type	Description
call	SipCallIdentifier structure	The unique ID of the originated call.
success	int	A non-zero value means that the operation was completed.

Call control API for IVR

URL (namespace): wss://portabilling-web.yourdomain.com/ws/CallControl

These API methods enable development and operation of custom IVR applications. Thus, users can manage the incoming calls and automate their processing (e.g. to play specific prompt upon user DTMF input).

Methods

play_prompt

Use this method to play an IVR prompt to a user for a call to the IVR access number. The prompts are stored on the remote server (separately or together with the IVR application). PortaSIP® must have access to this server.

Parameters: PlayPromptRequest
Return value: PlayPromptResponse
Realm: administrator, reseller, retail customer, account

Request example:

```
{
   "cseq":5,
   "auth_info":{
      "session_id":"0160ec8ffb4a394908d8be40e5b013d4"
   },
   "service":"CallControl",
   "method":"play_prompt",
   "params":{
      "call":{
         "id":"3300f0b3-7a54a006@192.168.233.134",
         "tag":"100021"
      },
      "order":3,
      "repeat":-1,
      "url":"http://192.168.233.137:8080/files/welcome",
   },
}
```

Response example:

```
{
   "cseq":5,
   "result":{
      "success":1
   },
   "success":1
}
```

stop_play_prompt

Use this method to stop playing prompts for a call to User Application access number.

Parameters: StopPlayPromptRequest
Return value: StopPlayPromptResponse
Realm: administrator, reseller, retail customer, account

Request example:

```
{
   "cseq":7,
   "auth_info":{
      "session_id":"0160ec8ffb4a394908d8be40e5b013d4"
   },
   "service":"CallControl",
   "method":"stop_play_prompt",
   "params":{
      "call":{
         "id":"3300f0b3-7a54a006@192.168.233.134",
         "tag":"100021"
      },
      "order":3,
   },
}
```

Response example:

```
{
   "cseq":7,
   "result":{
      "success":1
   },
   "success":1
}
```

Notification example for the IVR access number:

```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{
            "id":"3300f0b3-7a54a006@192.168.233.134",
            "tag":"100021"
         },
         "callee":{
            "access_number":"4567890"
         },
         "caller":{
            "account_id":"123002",
            "centrex_id":"30",
            "display_id":"123002",
            "forwarder_list":[

            ],
```

```
            "id":"123002"
         },
         "connect_time":"2018-12-05 08:51:16",
         "event":"play_prompt_completed",
         "order":2,
         "state":"connected",
         "type":"incoming"
      }
   }
}
```


start_dtmf_detect

Use this method to start detecting user DTMF inputs for a call to the IVR access number.

PortaSIP® supports the following DTMF modes: inband, RFC2833 and SIP INFO.

Parameters: StartDtmfDetectRequest
Return value: StartDtmfDetectResponse
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq":4,
   "auth_info":{
      "session_id":"0160ec8ffb4a394908d8be40e5b013d4"
   },
   "service":"CallControl",
   "method":"start_dtmf_detect",
   "params":{
      "call":{
         "id":"3300f0b3-7a54a006@192.168.233.134",
         "tag":"100021"
      },
   }
}
```


Response example:


```
{
   "cseq":4,
   "result":{
      "success":1
   },
   "success":1
}
```


Notification example for the DTMF received from the user:


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
```

```
         "call_info":{
            "call":{
               "id":"3300f0b3-7a54a006@192.168.233.134",
               "tag":"100021"
            },
            "callee":{
               "access_number":"4567890"
            },
            "caller":{
               "account_id":"123002",
               "centrex_id":"30",
               "display_id":"123002",
               "forwarder_list":[

               ],
               "id":"123002"
            },
            "connect_time":"2018-12-05 08:51:16",
            "digit":"1",
            "dtmf_duration":220,
            "event":"dtmf_digit_detected",
            "state":"connected",
            "type":"incoming"
         }
      }
   }
```


stop_dtmf_detect

Use this method to stop detecting DTMF inputs for a call to User Application access number.

Parameters: StopDtmfDetectRequest
Return value: StopDtmfDetectResponse
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq":17,
   "auth_info":{
      "session_id":"0160ec8ffb4a394908d8be40e5b013d4"
   },
   "service":"CallControl",
   "method":"stop_dtmf_detect",
   "params":{
      "call":{
         "id":"e0c8e7c-80b037c9@192.168.233.134",
         "tag":"100022"
      },
   },
}
```


Response example:


```
{
   "cseq":17,
   "result":{
```

```
      "success":1
   },
   "success":1
}
```


ring_call

Use this method to inform the caller about the call's progress. PortaSIP® sends the "180 Ringing" response.

Parameters: RingCallRequest
Return value: RingCallResponse
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq": 2,
   "auth_info": {
      "session_id": "649d127e95ca1ae62ec735df99ebf01c"
   },
   "service": "CallControl",
   "method": "ring_call",
   "params": {
      "call": {
         "id": "1202668471@192.168.233.48",
         "tag": "100007"
      }
   }
}
```


Response example:


```
{
   "cseq":2,
   "result":{
      "success":1
   },
  "success":1
}
```


Notification example for the IVR application:


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
         "call":{"id":"1202668471@192.168.233.48",
         "tag":"100007"
      },
         "callee":{
            "access_number":"*51"
         },
         "caller":{
            "account_id":"123003",
            "centrex_id":"30",
            "display_id":"123003",
```

```
                "display_name":"123003",
                "forwarder_list":[],
                "id":"123003"
            },
            "start_time":"2019-06-07 10:57:12",
            "state":"ringing",
            "type":"incoming"
        }
    }
}
```


progress_call

Use this method to inform the caller about the call's progress. PortaSIP® sends the "183 Early media" response. The caller can hear a custom ring tone.

Parameters: **ProgressCallRequest**
Return value: **ProgressCallResponse**
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq": 2,
   "auth_info": {
      "session_id": "649d127e95ca1ae62ec735df99ebf01c"
   },
   "service": "CallControl",
   "method": "progress_call",
   "params": {
      "call": {
         "id": "499624331@192.168.233.48",
         "tag": "100010"
      }
   }
 }
```


Response example:


```
{
   "cseq":2,
   "result":{
      "success":1
   },
  "success":1
}
```


Notification example for the IVR application:


```
{
   "action":"update",
   "event":"sip.call_control_notifications",
   "result":{
      "call_info":{
```

```
            "call":{"id":"499624331@192.168.233.48",
            "tag":"100010"
         },
            "callee":{
               "access_number":"*51"
            },
            "caller":{
               "account_id":"123003",
               "centrex_id":"30",
               "display_id":"123003",
               "display_name":"123003",
               "forwarder_list":[],
               "id":"123003"
            },
            "start_time":"2019-06-07 11:07:08",
            "state":"early",
            "type":"incoming"
         }
      }
   }
```


Type Reference

PlayPromptRequest structure

Property	Type	Description
call	**SipCallIdentifier** structure	The unique ID of separate parts of the call
url	string	The URL path to prompt files. Supported protocols are HTTP, HTTPS. Supported audio formats are: au, g729. The .au file must be in 8-bit G.711 u-law data encoding format. The URL must contain path to the file without file extension. E.g. if prompts are 'http://myhost.com/prompt.au', 'http://myhost.com/prompt.g729', url is 'https://myhost.com/prompt'. It is **not** mandatory to have prompts in all formats.
order	int	Prompt playback order (2 or 3). Represents relative position of one prompt to another (if any) in playback stack.

		Default value: 2
repeat	int	The number of times to repeat the prompt. "-1" - repeat playback forever. Default value: 1

PlayPromptResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

StartDtmfDetectRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The unique ID of separate parts of the call

StartDtmfDetectResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

StopDtmfDetectRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The unique IDs of separate parts of the call

StopDtmfDetectResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

StopPlayPromptRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The unique ID of separate parts of the call
order	int	Prompt playback order (2 or 3). Represents relative position of one prompt to another (if any) in playback stack.

		Default value: 2

StopPlayPromptResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

RingCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The identifier of the call

RingCallResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation has been successfully completed

ProgressCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The identifier of the call

ProgressCallResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation has been successfully completed

Conferencing API

These API methods enable agents to effectively manage their calls by converting active calls into conferences on the fly. While in a conference, agents can add / remove, mute and / or hold participants (e.g. to have a private talk with some participants while others remain connected). An agent can leave and join the conference they created at any moment while the call for the remaining participants continues.

An agent is represented as an account in PortaSwitch®. When the API session is established under the account realm, the agent who creates a conference automatically becomes its owner and can manage only this conference. When the API session is established under the customer realm, the agent has access to all conferences in the IP Centrex environment and can manage them (e.g. to appoint another account as the conference owner.) A conference owner is charged for the conferencing service.

To create conference rooms and / or convert calls into conferences, the agent's product configuration must include both the Voice calls and Conferencing services.

Methods

join_on_spot_conference

Use this method to create a conference room or to join to an active conference. When called under the account realm, this method enables the agent to join only the conference they have created. When called under the customer realm, this method enables creating conferences on behalf of any account under the customer and joining any active conference in the IP Centrex environment.

Parameters: JoinOnSpotConferenceRequest
Return value: JoinOnSpotConferenceResponse
Realm: administrator, reseller, retail customer, account

Request example:


```
{
   "cseq": 2,
   "auth_info": {
      "session_id": "637efe9391d6a9b10961052e86119655"
   },
   "service": "CallControl",
   "method": "join_on_spot_conference",
   "params": {
      "conference_info": {
          "name":"test",
          "owner_id": "12065550021"
      },
      "call_info": {
          "caller_id":"test's conference",
          "sip_auth_username": "12065550021",
          "callee_id":"12065550019"
       },
      "participant_info": {
            "mute": "N",
            "lang": "en",
             "quiet_mode": "N",
             "wait_for_moderator": "N",
             "moderator": "Y",
             "max_participants": "15"
       }
   }
}
```


Response example:

```
{
  "cseq":2,
   "success": 1,
  "call": {
    "id": "cSsJ!DNMIEZfxJrZlKJja12vKcb4@192.168.233.134"
  }
}
```


Notification example about the conference sent to the conference owner:


```
{
  "action":"update",
  "event": "sip.call_control_notifications",
  "result": {
     "conference_info": {
       "call": {
         "id": "cSsJ!DNMIEZfxJrZlKJja12vKcb4@192.168.233.134",
         "tag": 100028
      },
    "conference_name": "test's conference",
    "conference_owner_id": "12065550021",
    "event": "ConferenceInfo",
    "start_time": "2019-02-01 13:15:55",
    "status": "started"
      }
  }
}
```


Notification example about the participant sent to the conference owner:


```
{
  "action":"update",
  "event": "sip.call_control_notifications",
  "result": {
     "participant_info": {
       "call": {
         "id":
"cSsJ!DNMIEZfxJrZlKJja12vKcb4@192.168.233.134",
         "tag": 100028
      },
    "conference_name": "test's conference",
    "conference_owner_id": "12065550021",
    "event": "ParticipantInfo",
    "hold": "N",
    "join_time": "2019-02-01 13:15:58"
   "leave_time": None,
   "moderator": "Y",
   "mute": "Y",
   "participant_id": 1
      }
  }
}
```

convert_to_on_spot_conference

Use this method to convert an active call to a conference call. The agent's product configuration in PortaBilling® must include the Conferencing service.

Parameters: **ConvertToOnSpotConferenceRequest**
Return value: **ConvertToOnSpotConferenceResponse**
Realm: administrator, reseller, retail customer, account

Request example:


```
  {
    "cseq":2,
     "service": "CallControl",
     "method": "convert_to_on_spot_conference",
    "params": {
      "callee": {
        "moderator": "N",
        "max_participants": "15",
        "lang": "en",
        "quiet_mode": "N",
        "music_on_hold_url":
"http://91.212.119.34:5687/8_bit",
        "announced_conference_name_url": "
http://91.212.119.34:5687/test_conf",
      "wait_for_moderator": "Y"
    },
     "caller": {
       "moderator": "Y",
       "max_participants": "15",
       "lang": "en",
       "quiet_mode": "N",
       "music_on_hold_url": "http://91.212.119.34/8_bit",
       "announced_conference_name_url":
"http://192.168.233.134:5687/test_conf"
    },
    "conference_info": {
      "name": " test "
    },
    "call_info": {
      "tag": "fd15be61f2813a4eo2",
      "id": "f424a21d-908b49c2@192.168.***.***"
    }
  },
  "auth_info": {
    "session_id": "637efe9391d6a9b10961052e86119655"
  }
}
```


Response example:


```
{
   "cseq":2,
   "result":{
      "success":1
   },
   "success":1
}
```

get_on_spot_conference_list

Use this method to query the list of currently active conferences and receive the information about them. For this, subscribe to API notifications using the **enable_api_notifications** method.

Parameters: **GetOnSpotConferenceListRequest**
Return value: **GetOnSpotConferenceListResponse**
Realm: administrator, reseller, retail customer, account

Request example:

```
{
   "cseq": 3,
   "auth_info": {
      "session_id": "637efe9391d6a9b10961052e86119655"
   },
   "service": "CallControl",
   "method": "get_on_spot_conference_list",
   "params": {
      "conference_info": {}
}
```

Response example:

```
{"cseq": 3,
 "result": {"conference_list": [
             {"i_customer": 4871,
              "name": "test_conference",
              "owner_id": "12065550021 "}
             ]
           },
 "success": 1
}
```

get_on_spot_conference_participant_list

Use this method to query information about participants in a given conference. The response contains all participants: those who are in the conference and those who left.

Parameters: **GetOnSpotConferenceParticipantListRequest**
Return value: **GetOnSpotConferenceParticipantListResponse**
Realm: administrator, reseller, retail customer, account

Request example:

```
{
   "cseq": 3,
   "auth_info": {
      "session_id": "637efe9391d6a9b10961052e86119655"
   },
   "service": "CallControl",
   "method": "get_on_spot_conference_participant_list",
   "params": {
```

```
        "conference_info": {
            "name":"test_conference",
             "owner_id": "12065550021"
        }
  }
```

Response example:

```
  {"cseq": 3,
   "result": "participant_list": [
      {
        "caller_id": "12065550021",
        "moderator": "N",
        "participant_id": 1,
        "mute": "N",
        "hold": "N",
        "join_time": "2019-02-08 14:29:07"
      },
      {
        "callee_id": "12065550019",
        "moderator": "N",
        "participant_id": 2,
        "mute": "N",
        "hold": "N",
        "join_time": "2019-02-08 14:29:16"
      }
    ]
  }
```

update_on_spot_conference_participant

Use this method to update the participant status. This method enables you to mute and unmute participants, put them on hold and release them from hold.

Parameters: UpdateOnSpotConferenceParticipantRequest
Return value: UpdateOnSpotConferenceParticipantResponse
Realm: administrator, reseller, retail customer, account

Request example (to mute a participant):

```
  {
    "cseq":2,
     "service": "CallControl",
     "method": "update_on_spot_conference_participant",
    "params": {
      "participant_info": {
        "participant_id": "1",
        "mute": "Y",
        "music_on_hold_url":
"http://91.212.119.34:5687/8_bit",
        "hold": "N",
    },
    "conference_info": {
      "name": "test"
      "owner_id":"12065550021"
    },
    "auth_info": {
      "session_id": "637efe9391d6a9b10961052e86119655"
```

```
  }
}
```

Response example:


```
{
  "cseq":2,
    "result":{
     "success":1
  },
  "success":1
}
```


delete_on_spot_conference_participant

Use this method to delete participants from on the spot conference by disconnecting them.

Parameters: DeleteOnSpotConferenceParticipantRequest
Return value: DeleteOnSpotConferenceParticipantResponse
Realm: administrator, reseller, retail customer, account

Request example:

```
{
    "cseq": 2,
    "auth_info": {
       "session_id": "637efe9391d6a9b10961052e86119655"
    },
    "service": "CallControl",
    "method": "delete_on_spot_conference_participant",
    "params": {
       "conference_info": {
           "name":"test",
           "owner_id": "12065550021"
       },
        "participant_info": {
             "participant_id": "1",
             "play_announce": "Y",
        }
    }
}
```

Response example:


```
{
  "cseq":2,
    "result":{
     "success":1
  },
  "success":1
}
```

Type reference

JoinOnSpotConferenceRequest structure

Property	Type	Description
call_info	SipOnSpotConferenceCallInfo structure	The information about the call
conference_info	SipOnSpotConferenceInfo structure	The information about the conference
participant_info	SipOnSpotConferenceParticipantInfo structure	The information about the conference participant

JoinOnSpotConferenceResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed
call	SipCallIdentifier structure	The unique ID of the originated call.

SipOnSpotConferenceCallInfo structure

Property	Type	Description
caller_id	string	The phone number used to establish a call to a new conference participant
callee_id	string	The phone number of a conference participant to be called
id	string	The call identifier
sip_auth_username	string	The SIP username used for authorization and charges for a call to a new conference participant. By default, the conference room owner is charged for a voice call to a conference participant and the time spent by the participant in the conference room. It is equal to the *owner_id* attribute if the method is executed under the account realm. If the method is executed

		under the customer realm, this attribute's value must correspond to IDs of one of the accounts of this customer
tag	string	The call remote tag
transport_id	string	The SIP transport ID

SipOnSpotConferenceInfo structure

Property	Type	Description
name	string (max 64 chars)	The name of the conference room. It must be unique among the rooms created by the agent or within the IP Centrex environment. This means that any user who joins "room3121" under specific account/customer credentials will end up in the same conference room, while the user connecting through different accounts/customers will not
owner_id	string	The unique ID of the conference room owner's account. By default, and if the *sip_auth_username* attribute is not defined, the conference room owner is charged for voice calls to conference participants and the time spent by every participant in the conference room. If the method is executed under account realm, this value is not mandatory as it is equal to this account ID. If executed under the customer realm, this attribute's value must correspond to IDs of one of the accounts of this customer
i_customer	int	The unique ID of the customer record. To get the customer ID, call the **get_customer_info** method. The *i_customer* is

		returned in the CustomerInfo structure. See **https://www.portaone.com/docs/PortaBilling_API.html#CustomerInfo**.

SipOnSpotConferenceParticipantInfo structure

Property	Type	Description
announced_conference_name_url	string	Specifies a URL for the sound file with the name of the conference which will be announced to people joining it. Supported protocols are HTTP, HTTPS. Supported audio formats are: au, g729. The .au file must be in the 8-bit G.711 u-law data encoding format. The URL must contain the path to the file without the file extension. E.g. if the prompts are 'http://myhost.com/prompt.au', 'http://myhost.com/prompt.g729', the URL is 'https://myhost.com/prompt'. It is **not** mandatory to have prompts in all formats.
caller_id	string	The phone number used to establish a call to a new conference participant
hold	string	This flag indicates whether the participant is on hold in the conference. Possible values: Y, N
join_time	date Time	The date and time when the participant joined the on-the-spot conference
teave_time	date	The date and time when

	Time	the participant left the on-the-spot conference
play_announce	string	The flag shows whether to play an announcement before removing a participant from the conference
lang	string	The ISO 639-1 language the user prefers when interacting with the conferencing IVR application. Find the list of supported languages in Appendices section of the **PortaSIP® Media Applications Guide**. If the specified language is unsupported, PortaSIP® plays prompts in English. Default value: en
max_participants	int	The maximum number of concurrent participants allowed in the conference room. This value must not exceed the maximum number of simultaneous participants defined for the conference room owner within their account or product configuration
moderator	string	This flag indicates whether the participant is the moderator. A moderator is the person who is responsible for arranging the conference and is usually the host or leader of the conference. Possible values: Y, N Default value: N
music_on_hold_url	string	Specifies a URL for the sound file with the name of the conference which will be announced to people joining it.

		Supported protocols are HTTP, HTTPS. Supported audio formats are: au, g729. The .au file must be in the 8-bit G.711 u-law data encoding format. The URL must contain the path to the file without the file extension. E.g. if the prompts are 'http://myhost.com/prompt.au', 'http://myhost.com/prompt.g729', the URL is 'https://myhost.com/prompt'. It is **not** mandatory to have prompts in all formats.
mute	string	The flag indicates whether the participant can speak in the conference. Possible values: Y, N Default value: N
quiet_mode	string	This flag indicates whether PortaSIP® plays the welcome message and enter / leave sounds to conference participants. When enabled, the participant joins the conference without others being aware of that (e.g. to monitor the conversation.) Possible values: Y, N Default value: N
participant_id	int	The unique identifier of the conference participant
wait_for_moderator	string	This flag indicates that the conference participants will not be able to communicate with each

		other until a moderator joins the room. When this flag is enabled and the last moderator leaves the conference room, the conference ends and all other participants are disconnected. Possible values: Y, N. Default value: N

ConvertToOnSpotConferenceRequest structure

Property	Type	Description
conference_info	**SipOnSpotConferenceInfo** structure	The information about the conference
call_info	**SipOnSpotConferenceCallInfo** structure	The information about the call
callee	**SipOnSpotConferenceParticipantInfo** structure	The information about the participant who the caller was calling to
caller	**SipOnSpotConferenceParticipantInfo** structure	The information about the participant who was calling

ConvertToOnSpotConferenceResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

GetOnSpotConferenceListRequest structure

Property	Type	Description
conference_info	**SipOnSpotConferenceInfo** structure	The information about the conference

GetOnSpotConferenceListResponse structure

Property	Type	Description
conference_list	Array of SipOnSpotConferenceInfo structures	The list of active conference rooms

GetOnSpotConferenceParticipantListRequest structure

Property	Type	Description
conference_info	SipOnSpotConferenceInfo structure	The information about the conference

GetOnSpotConferenceParticipantListResponse structure

Property	Type	Description
participant_list	Array of SipOnSpotConferenceParticipantInfo structures	The information about the conference participants

UpdateOnSpotConferenceParticipantRequest structure

Property	Type	Description
conference_info	SipOnSpotConferenceInfo structure	The information about the conference
participant_info	SipOnSpotConferenceParticipantInfo structure	The information about the conference participant

UpdateOnSpotConferenceParticipantResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

DeleteOnSpotConferenceParticipantRequest structure

Property	Type	Description
conference_info	SipOnSpotConferenceInfo structure	The information about the conference
participant_info	SipOnSpotConferenceParticipantInfo structure	The information about the conference participant

DeleteOnSpotConferenceParticipantResponse structure

Property	Type	Description
success	int	A non-zero value means that the operation was completed

4. Appendices

Appendix A. Sample script for PortaSIP® media server SOAP communication

```
#!perl -w
use strict;
# to enable client-side script debugging uncomment the line below
# and comment the one next to it
#use SOAP::Lite   +trace => 'debug';

use SOAP::Lite;
use MIME::Entity;
use Data::Dumper;

# If the server certificate is not trusted (e.g. it was not issued by a
# trusted certificate authority), then ignore it.
$ENV{PERL_LWP_SSL_VERIFY_HOSTNAME}=0;

##### Preparing ########################################################

my $soap_sess = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/Session')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $soap_test = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/Voicemail')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $soap_dial_dir = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/DialDirectory')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $soap_aa = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/AutoAttendant')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $authInfo = $soap_sess->login({
        'login'     => '88881',
        'domain'    => 'pum.somedomain.com',
        'password'  => 'test123'})->result();
$authInfo = SOAP::Header->name( 'auth_info' => {
                'session_id' => $authInfo->{'session_id'}
                });
my $authInfo_nosess = SOAP::Header->name( 'auth_info' => {
        'login'     => '88881',
        'domain'    => 'pum.somedomain.com',
        'password'  => 'test123'
                });
```

```
########################################################################

my $res;
#example of accessing SOAP module without establishing session
$res = $soap_test->get_vm_settings($authInfo_nosess)->result();
print Dumper($res);
$res = $soap_test->set_vm_settings($authInfo,
        {'vm_settings' =>
            {
                'password'  => '777',
                'ext_email' => 'sergey.pavlov@gmail.com',
                'auto_play' => 'no',
                'announce_dt'=> 'no'
            } })->result();
print "set_vm_settings done\n";
my $ent =  MIME::Entity->build(
                'Filename'      => 'wellcome.au',
                'Type'   => 'audio/basic',
                'Encoding'  => 'base64',
                'Path'      =>
'/var/lib/psmsc/prompts/en/personal_ivr/frw_select_order.au',
                );

my @parts = ($ent);
$res = $soap_test->parts([ $ent ])->set_vm_greeting($authInfo,

        {'greeting_info' =>
            {
                'greeting_type'  => 'name',
                'filename'=> 'wellcome.au'
            } })->result();
print "set_vm_greeting done\n";

$res = $soap_test->get_vm_greeting($authInfo,
            {
                'greeting_type'  => 'name',
            })->result();
print "get_vm_greeting done\n";

######## Dial Directory ############################

$res = $soap_dial_dir->get_dir_info($authInfo)->result();
print "get_directory_list done\n";
print Dumper($res);

$res = $soap_dial_dir->parts(@parts)->create_dir_entry($authInfo,
    {
    'dir_entry_info'  => {
    'active'            => 'Y',
    'abbreviated_number'=> '1787896',
    'number_to_dial'    => '111111',
    'lastname'          => 'LName',
    'description'       => 'desc foo',
    'prompt'            => 'wellcome.au'
    }
    } )->result();
print "create_dir_entry done\n";
print Dumper($res);
my $i_entry = $res->{'i_entry'};
$res = $soap_dial_dir->parts(@parts)->update_dir_entry($authInfo,
    {
    'dir_entry_info'  => {
    'i_entry'    => $i_entry,
    'active'            => 'Y',
    'abbreviated_number'=> '99',
    'number_to_dial'    => '565656',
    'lastname'          => 'LastName',
    'description'       => 'desc333',
    'prompt'            => 'wellcome.au'
    }
    } )->result();
```

```
print "update_dir_entry done\n";
$res = $soap_dial_dir->get_dir_entry($authInfo, {'i_entry' => $i_entry}
)->result();
print "get_dir_entry done\n";
print Dumper($res);

$res = $soap_dial_dir->del_dir_entry($authInfo,
    {'i_entry'  => $res->{'dir_entry_info'}->{'i_entry'}})->result();
print "del_dir_entry done\n";
print Dumper($res);

######## Auto Attendant ############################

$res = $soap_aa->get_menu_list($authInfo)->result();
if (!$res) {
    print "get_menu_list failed\n";
}
print "get_menu_list done\n";
print Dumper($res);
my $root_i_menu;
foreach my $menu (@{$res->{'menu_list'}}) {
    if ($menu->{'name'} eq 'ROOT') {
        $root_i_menu = $menu->{'i_menu'};
        last;
    }
}

$res = $soap_aa->parts(@parts)->set_menu_prompt($authInfo,
    {
        'i_menu'        => $root_i_menu,
        'prompt_type'   => 'intro',
        'prompt'        => 'wellcome.au'
    })->result();
print "set_menu_prompt done\n";
print Dumper($res);

$res = $soap_aa->set_menu_transition($authInfo,
    {
        'transition_info' => {
        'i_menu'    => $root_i_menu,
        'event'     => '0',
        'action'    => 'Transfer',
        'destination'=> '5555',
        }
    })->result();
print "set_menu_transition done\n";
print Dumper($res);

$res = $soap_aa->get_menu_transition_list($authInfo,
    {
        'i_menu'    => $root_i_menu,
    })->result();
print "get_menu_transition_list done\n";
print Dumper($res);

$res = $soap_aa->get_menu_prompt($authInfo,
    {
        'i_menu'        => $root_i_menu,
        'prompt_type'   => 'intro',
    })->result();
print "get_menu_prompt done\n";
print Dumper($res);

$res = $soap_aa->create_menu($authInfo,
    { 'menu_info' => {
        'name'          => 'AABBBCCC',
        'period'        => 'hr{0-11}',
        'period_desc'   => 'Some period',
        'msg_timeout_type' => 'standard'
        }
    })->result();
print "create_menu done\n";
```

```
print Dumper($res);
my $new_i_menu=$res->{'i_menu'};

$res = $soap_aa->update_menu($authInfo,
    { 'menu_info' => {
        'i_menu'        => $new_i_menu,
        'name'          => 'DDDEEFF',
        'period'        => 'hr{0-2}',
        'period_desc'   => 'New period',
        }
    })->result();
print "update_menu done\n";
print Dumper($res);

$res = $soap_aa->get_menu_list($authInfo)->result();
print "get_menu_list done\n";
print Dumper($res);

$res = $soap_aa->del_menu($authInfo,
    { 'i_menu' => $new_i_menu })->result();
print "del_menu done\n";
print Dumper($res);

$res = $soap_sess->logout($authInfo)->result();
print "logout done\n";
print Dumper($res);
```

Appendix B. Call control API usage examples

Below you will finds scripts written in Python and JavaScript that show the communication with PortaSwitch® via the Call control API. Both scripts initiate a WebSocket connection and enable the receipt of call state notifications for the IP Centrex environment.

JavaScript

This is the .html page with the options to connect to PortaSwitch via WebSocket and start receiving call state notifications and close the connection. The action log is shown in the output field.

```
<!DOCTYPE html>

<html>
<head>
    <meta charset="utf-8">

    <title>PortaBilling Call Controll API example</title>
    <script language="javascript" type="text/javascript">
      // web socket server URL
      // replace <web-server.yourdomain.com> with the actual hostname
of your web server
      // replace <port> with the required port
      var wsUri = "wss:///<web-server.yourdomain.com>:<port>/ws";

      // request body for enabling notifications receiving for the IP
Centrex
      // replace <session id> with authenticated session id (can be
obtained from Session.login response)
      // replace <i_customer> with id of a customer which notifications
you want to receive
```

```
      var enable_notif_rqst = '{"cseq": 2, "service": "Customer",
"method": "enable_api_notifications", "auth_info":
{"session_id":"<session id>"}, "params": { "i_customer":
"<i_customer>", "event": "sip.call_control_notifications" }}';

      // request body for disabling notifications receiving
      var disnable_notif_rqst = '{"cseq": 3, "service": "Customer",
"method": "disable_api_notifications", "auth_info":
{"session_id":"<session id>"}, "params": { "i_customer":
"<i_customer>", "event": "sip.call_control_notifications" }}';

      var output;
      var websocket;

      function connect() {
          output = document.getElementById("output");
          startReceivingNotifications();
      }

      function startReceivingNotifications() {
          // create new WebSocket instance
          websocket = new WebSocket(wsUri);
          websocket.onopen = function(evt) {
              // socket opening event handler
              onOpen(evt);
          };
          websocket.onclose = function(evt) {
              // socket closing event handler
              onClose(evt);
          };
          websocket.onmessage = function(evt) {
              // message receiving event handler
              onMessage(evt);
          };
          websocket.onerror = function(evt) {
              // error event handler
              onError(evt);
          };
      }

      function onOpen(evt) {
          writeToScreen("CONNECTED");
          //send request to start notifications receiving
          doSend(enable_notif_rqst);
      }

      function onClose(evt) {
          writeToScreen("DISCONNECTED");
      }

      function onMessage(evt) {
          // display received response
          writeToScreen('<span style="color: blue;">RECEIVED: ' +
evt.data + '<\/span>');
      }

      function onError(evt) {
          // display received error message
          writeToScreen('<span style="color: red;">ERROR:<\/span> ' +
evt.data);
      }

      function doSend(message) {
          // display sent request
          writeToScreen("SENT: " + message);
          // send request to the web socket server
          websocket.send(message);
      }

      function writeToScreen(message) {
          var pre = document.createElement("p");
          pre.style.wordWrap = "break-word";
```

```
            pre.innerHTML = message;
            output.appendChild(pre);
        }

        function disconnect() {
            //send request to stop notifications receiving
            doSend(disnable_notif_rqst);
            // close web socket
            websocket.close();
        }

    </script>
</head>

<body>
    <h2>PortaBilling Call Controll API example</h2>

    <input type="button" name="connect" value="Connect"
onclick=connect()>
    <input type="button" name="disconnect" value="Disconnect"
onclick=disconnect()>

    <div id="output" style="height:200px; border: 1px solid black;
overflow: scroll; margin-top: 10px;"></div>
</body>
</html>
```

Python

This script connects to the WebSocket server, enables the receipt of notifications, waits for 10 seconds for notifications. After that the script disables the receipt of notifications and closes the WebSocket connection.

```
import websocket
try:
    import thread
except ImportError:
    import _thread as thread
import time

# web socket server URL
# replace <web-server.yourdomain.com> with the actual hostname of your
web server
ws_uri = "wss://<web-server.yourdomain.com>/ws"

# request body for enabling notifications receiving for the IP Centrex
# replace <session id> with authenticated session id (can be obtained
from Session.login response)
# replace <i_customer> with id of a customer which notifications you
want to receive
enable_receiving = '{"cseq": 3, "service": "Customer", "method":
"enable_api_notifications", "auth_info": {"session_id": "<session
id>"}, "params": { "i_customer": "<i_customer>", "event":
"sip.call_control_notifications" }}'

# request body for disabling notifications receiving
disable_receiving = '{"cseq": 3, "service": "Customer", "method":
"disable_api_notifications", "auth_info": {"session_id": "<session
id"}, "params": { "i_customer": "<i_customer>", "event":
"sip.call_control_notifications" }}'

def on_message(ws, message):
    print("RECEIVED: %s" % message)

def on_error(ws, error):
    print(error)

def on_close(ws):
```

```
    print("DISCONNECTED")

def on_open(ws):
    def run(*args):
        print("CONNECTED")
        print("SENT: %s" % enable_receiving)
        # send request to start notifications receiving
        ws.send(enable_receiving)

        # waiting for notifications
        time.sleep(10)

        # send request to stop notifications receiving
        ws.send(disable_receiving)
        # wait before closing websocket
        time.sleep(5)
        # close websocket
        ws.close()

    thread.start_new_thread(run, ())


if __name__ == "__main__":
    ws = websocket.WebSocketApp(ws_uri,
                              on_message = on_message, # message
receiving event handler
                              on_error = on_error,     # error event
handler
                              on_close = on_close)     # socket closing
event handler
    ws.on_open = on_open
    ws.run_forever()
```

Appendix C. How to define a time period

A time period is specified as a string in the following format:

```
sub-period[, sub-period...]
```

A sub-period takes the following form:

```
scale {range [range ...]} [scale {range [range ...]}]
```

The scale must be one of the nine different options (or their equivalent codes):

Scale	Scale Code	Valid Range Values
year	yr	n – where n is an integer 0<=n<=99 or n>=1970
month	mo	1-12 or jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec
week	wk	1-6
yday	yd	1-365
mday	md	1-31
wday	wd	1-7 or su, mo, tu, we, th, fr, sa
hour	hr	0-23 or 12am 1am-11am 12noon 12pm 1pm-11pm

minute	min	0-59
second	sec	0-59

The same scale type may be specified multiple times. Additional scales simply extend the range defined by previous scales of the same type. The range of a given scale must be a valid value in the form:

`v`

or

`v-v`

In the range specification ***v-v***, if the second value is larger than the first, the range wraps around unless the scale specification is “year”. Year does not wrap because a year is never really reset, rather it just changes by increments.

Ignoring that fact that led to the dreaded Y2K nightmare, when a year rolls over from 99 to 00, it has really rolled over one century, not gone back a century. Time period supports the ambiguous two digit year notation because it is so widespread.

However, two-digit notation is converted to four digits by prepending the first two digits of the current year. In the case of 99-1972, the 99 is translated to whatever the current century is (probably the 20th), and so the range 99-1972 is treated as 1972-1999. For the 21st century, the range would then be 1972-2099.

In any case, if `v-v` is 9-2, and the scale is month, September, October, November, December, January, and February are the months specified by the range (9-2 is the same as Sep-Feb).

If `v-v` is 2-9, then the valid months are February, March, April, May, Jun, July, August, and September.

`v` is not a point in time. For the hour scale, 9 specifies the time period from 9:00:00 am to 9:59:59 am. This is what most people would call 9-10.

In other words, `v` is discrete in its time scale. 9 changes to 10 when 9:59:59 changes to 10:00:00, but 9 is the period from 9:00:00 to 9:59:59. Just before 9:00:00, `v` was 8.

Note that there can be a white space anywhere, and case is unimportant. Note also that scales must be specified either in long form (year, month, week, etc.) or in code form (yr, mo, wk, etc.). Scale forms in a period statement may be mixed.

Furthermore, when using letters to specify ranges, only the first two (for weekdays) or the first three (for months) are significant. January is a valid specification for Jan, and Sunday is a valid specification for su. Sun is also valid for su.

Period Examples

Example 1

To specify a time period from Monday through Friday, 9 a.m. to 5 p.m., use the following period:

```
wd {Mon-Fri} hr {9am-4pm}
```

When specifying a range using "-", it is best to think of "-" as meaning "through", i.e. 9 a.m. through 4 p.m., which is the time interval ending just before 5 p.m.

Example 2

To specify a time period from 9 a.m. to 5 p.m. on Monday, Wednesday, and Friday and from 9 a.m. to 3 p.m. on Tuesday and Thursday, use the following period:

```
wd {Mon Wed Fri} hr {9am-4pm}, wd{Tue Thu} hr {9am-2pm}
```

Example 3

To specify a time period that extends from Monday to Friday, 9 a.m. to 5 p.m., but alternates the weeks in a month, use the following period:

```
wk {1 3 5} wd {Mon Wed Fri} hr {9am-4pm}
```

Example 4

For a period that specifies the winter:

```
mo {Nov-Feb}
```

The next example is equivalent to the previous one:

```
mo {Jan-Feb Nov-Dec}
```

as is:

```
mo {jan feb nov dec}
```

or also:

```
mo {Jan Feb}, mo {Nov Dec}
```

and this, too:

```
mo {Jan Feb} mo {Nov Dec}
```

Example 5

To specify a period of every other half-hour, use something like this:

```
minute {0-29}
```

Example 6

To specify the morning, use the following period definition:

```
hour {12am-11am}
```

Please note that '11 a.m.' here is not the 11:00:00 a.m. time point but the 11:00:00 a.m.–11:59:59 a.m. interval.

Example 7

To specify the period that consists of several 5-second blocks:

```
sec {0-4 10-14 20-24 30-34 40-44 50-54}
```

Example 8

To specify every first half-hour on alternating weekdays, and the second half-hour during the rest of the week, use the following period:

```
wd {1 3 5 7} min {0-29}, wd {2 4 6} min {30-59}
```