

SOAP and REST/JSON API - Documentation

API Documentation

This Document is a Complete Reference Guide for the use of VoIP.ms API.
[Example Codes](#) can be downloaded from [here](#)

INTRODUCTION

With the VoIP.ms API, you can easily integrate VoIP.ms functionality on your website, intranet, extranet or customer portal.

The VoIP.ms API provides the basic functions needed to manage, purchase and cancel DID numbers, create and edit sub-accounts, consult CDR, balance and much more.

This document is aimed to help you setup and integrate VoIP.ms API properly into your website. It will also provide a complete list of functions, parameters needed, response from functions and possible error codes with their meaning.

VoIP.ms also supports Application-to-Person (A2P) delivery for business text messaging. However, requirements and fees vary due to industry regulations. The default limit is 100 SMS per day, which can be raised upon request. Contact us at messaging@voip.ms to start your verification process.

UPDATE API PASSWORD

You need to set the API Password before you enable the API Access for your account.
Please note that this password is only used for the API in order to give access to its functions.

To Update your Password please follow these steps:

1. Log-in to your VoIP.ms account
2. Go to "Main Menu" -> "SOAP & REST/JSON API"
3. Introduce the Password you want to use to access the API
4. Click on button [Save API Password] to Update the API Password

API Password has been saved

API Configuration

API Password	New Password: • Password must be at least 8 characters long • Password must contain at least one upper case character • Password must contain at least one lower case character • Password must contain at least one number • Password allows alphanumeric and the following characters: ! # \$ % & / () = ? * [] _ : . , { } + - Confirm Password: 	Save API Password
Enable API	Enabled	Enable/Disable API
Enable IP Addresses		Save IP Addresses

Feedback

The VoIP.ms API is not enabled by default, you have to allow the API Access from within your account following these steps:

- 1. Log-in to your VoIP.ms account
- 2. Go to "Main Menu" -> "SOAP & REST/JSON API"
- 3. Click on button [Enable/Disable API] to Enable / Disable the API Access

API has been Enabled

API Configuration

API Password ?	New Password: • Password must be at least 8 characters long • Password must contain at least one upper case character • Password must contain at least one lower case character • Password must contain at least one number • Password allows alphanumeric and the following characters: ! # \$ % & / () = ? * [] _ : . , { } + - Confirm Password: 	Save API Password
Enable API ?	Enabled	Enable/Disable API
Enable IP Addresses ?		Save IP Addresses

ENABLE IP ADDRESSES

By default no IP Address is able to consume services from VoIP.ms API.
IP Addresses must be enabled on Per Account Basis following these steps:

- 1. Log-in to your VoIP.ms account
- 2. Go to "Main Menu" -> "SOAP & REST/JSON API"
- 3. Introduce the IP Address where the API is going to be consumed from
- 4. Click on [Save IP Addresses] button to Enable the IP Addresses

IP Addresses have been saved

API Configuration

API Password ?	New Password: • Password must be at least 8 characters long • Password must contain at least one upper case character • Password must contain at least one lower case character • Password must contain at least one number • Password allows alphanumeric and the following characters: ! # \$ % & / () = ? * [] _ : . , { } + - Confirm Password: 	Save API Password
Enable API ?	Enabled	Enable/Disable API
Enable IP Addresses ?	174.132.203.114	Save IP Addresses

You can add various IP Addresses by separating them with a comma (example: 192.168.1.1, 192.168.1.2)

You can also add the following type of values:

- IP Ranges (example: 192.168.1.0/24, 192.168.2.0/24)
- Wildcards (example: 192.168.1.*, 192.168.2.*)
- DNS (example: sub1.domain.tld, sub2.domain.tld)

If you don't know what IP to enable, try using the function "getIP"
- Shows the IP used by the client application requesting information from the API
* this is the only function not using the IP for authentication.
* the IP returned should be the one used in the API Configuration.

FUNCTIONS LIST

VoIP.ms API has over 100 functions to help you integrate our services into your website.

All of these functions expect a minimum of two parameters:

- api_username (email used to login to VoIP.ms portal)
- api_password (password created for the API)

The functions are separated in the following categories:

General	
getBalance	- Retrieves Balance for your Account if no additional parameter is provided. - Retrieves Balance and Calls Statistics for your Account if "advanced" parameter is true.
getConference	- Retrieves a list of Conferences if no additional parameter is provided. - Retrieves a specific Conference if a conference code is provided.
getConferenceMembers	- Retrieves a list of Member profiles if no additional parameter is provided. - Retrieves a specific member if a member code is provided.
getConferenceRecordings	- Retrieves a list of recordings of a specific conference.
getConferenceRecordingFile	- Retrieves a specific Recording File data in Base64 format.
getSequences	- Retrieves a list of Sequences if no Sequence ID is provided. - Retrieves a specific Sequence if a Sequence ID is provided.
getCountries	- Retrieves a list of Countries if no additional parameter is provided. - Retrieves a specific Country if a country code is provided.
getIP	- Shows the IP used by the client application requesting information from the API * this is the only function not using the IP for authentication. * the IP returned should be the one used in the API Configuration.
getLanguages	- Retrieves a list of Languages if no additional parameter is provided. - Retrieves a specific Language if a language code is provided.
getLocales	- Retrieves a list of locale codes if no additional parameter is provided. - Retrieves a specific locale code if a language code is provided.
getServersInfo	- Retrieves a list of Servers with their info if no additional parameter is provided. - Retrieves a specific Server with its info if a Server POP is provided.
getTransactionHistory	- Retrieves the Transaction History records between two dates.
Accounts	
createSubAccount	- Adds a new Sub Account entry to your Account
delSubAccount	- Deletes a specific Sub Account from your Account
getAllowedCodecs	- Retrieves a list of Allowed Codecs if no additional parameter is provided. - Retrieves a specific Allowed Codec if a codec code is provided.
getAuthTypes	- Retrieves a list of Authentication Types if no additional parameter is provided. - Retrieves a specific Authentication Type if an auth type code is provided.
getDeviceTypes	- Retrieves a list of Device Types if no additional parameter is provided. - Retrieves a specific Device Type if a device type code is provided.
getDTMFModes	- Retrieves a list of DTMF Modes if no additional parameter is provided. - Retrieves a specific DTMF Mode if a DTMF mode code is provided.
getInvoice	- Retrieves a URL to download the invoice in a PDF file.
getLockInternational	- Retrieves a list of Lock Modes if no additional parameter is provided. - Retrieves a specific Lock Mode if a lock code is provided.
getMusicOnHold	- Retrieves a list of Music on Hold Options if no additional parameter is provided. - Retrieves a specific Music on Hold Option if a MOH code is provided.
delMusicOnHold	- Deletes a specific custom Music on Hold.
getNAT	- Retrieves a list of NAT Options if no additional parameter is provided. - Retrieves a specific NAT Option if a NAT code is provided.
getProtocols	- Retrieves a list of Protocols if no additional parameter is provided. - Retrieves a specific Protocol if a protocol code is provided.
getRegistrationStatus	- Retrieves the Registration Status of all accounts if no account is provided.
getReportEstimatedHoldTime	- Retrieves a list of 'ReportEstimateHoldTime' Types if no additional parameter is provided. - Retrieves a specific 'ReportEstimateHoldTime' Type if a type code is provided.
getRoutes	- Retrieves a list of Route Options if no additional parameter is provided. - Retrieves a specific Route Option if a route code is provided.
getSubAccounts	- Retrieves all Sub Accounts if no additional parameter is provided. - Retrieves Reseller Client Accounts if Reseller Client ID is provided. - Retrieves a specific Sub Account if a Sub Account is provided.
addMemberToConference	- Add Member to a Conference
setSubAccount	- Updates Sub Account information.
Call Detail Records	
getCallAccounts	- Retrieves all Sub Accounts if no additional parameter is provided. - Retrieves Reseller Client Accounts if Reseller Client ID is provided.
getCallBilling	- Retrieves a list of Call Billing Options.
getCallTypes	- Retrieves a list of Call Types and All DIDs if no additional parameter is provided. - Retrieves a list of Call Types and Reseller Client DIDs if a Reseller Client ID is provided.

getCDR	- Retrieves the Call Detail Records of all your calls.
getRates	- Retrieves the Rates for a specific Package and a Search term.
getTerminationRates	- Retrieves the Rates for a specific Route (Premium, Value) and a Search term.
getResellerCDR	- Retrieves the Call Detail Records for a specific Reseller Client.
getResellerSMS	- Retrieves a list of SMS messages for a specific Reseller Client. by: date range, sms type, DID number, and contact
getResellerMMS	- Retrieves a list of MMS messages for a specific Reseller Client. by: date range, mms type, DID number, and contact
Call Parking	
getCallParking	- Retrieves all Call Parking entries if no additional parameter is provided. - Retrieves a specific Parking entry if a Call Parking ID is provided.
setCallParking	- Updates a specific Call Parking entry if a Call Parking ID is provided. - Adds a new Call Parking entry if no Call Parking ID is provided.
delCallParking	- Deletes a specific Call Parking entry from your Account.
Call Recordings	
getCallRecordings	- Retrieves all call recordings related to account.
getCallRecording	- Retrieves one especific call recording information, including the recording file on mp3 format.
sendCallRecordingEmail	- Send information and audio file to email account.
delCallRecording	- Delete specific call recording, audio file and information related.
Clients	
addCharge	- Adds a Charge to a specific Reseller Client
addPayment	- Adds a Payment to a specific Reseller Client
assignDIDvPRI	- Assigns a Per Minute DID to a VPRI (Flat Rate DIDs can't be assigned)
getBalanceManagement	- Retrieves a list of Balance Management Options if no additional parameter is provided. - Retrieves a specific Balance Management Option if a code is provided.
getCharges	- Retrieves Charges made to a specific Reseller Client.
getClientPackages	- Retrieves a list of Packages for a specific Reseller Client.
getClients	- Retrieves a list of all Clients if no additional parameter is provided. - Retrieves a specific Reseller Client if a Reseller Client ID is provided. - Retrieves a specific Reseller Client if a Reseller Client e-mail is provided.
getClientThreshold	- Retrieves the Threshold Information for a specific Reseller Client.
getDeposits	- Retrieves Deposits made for a specific Reseller Client.
getPackages	- Retrieves a list of Packages if no additional parameter is provided. - Retrieves a specific Package if a package code is provided.
getResellerBalance	- Retrieves Balance and Calls Statistics for a specific Reseller Client for the last 30 days and current day.
setClient	- Updates Reseller Client information.
setClientThreshold	- Update the Threshold Amount for a specific Reseller Client.- Update the Threshold notification e-mail for a specific Reseller Client if the e-mail address is provided.
setConference	- Updates a specific Conference if a conference code is provided. - Adds a new Conference entry if no conference code is provided.
setConferenceMember	- Updates a specific Member profile if a member code is provided. - Adds a new Member profile entry if no member code is provided.
setSequences	- Updates a specific Sequence if a Sequence ID is provided. - Adds a new Sequence entry if no Sequence ID is provided.
signupClient	- Signs a new Reseller Client to your Reseller Account.
DIDs	
backOrderDIDUSA	- Backorder DID (USA) from a specific ratecenter and state.
backOrderDIDCAN	- Backorder DID (CANADA) from a specific ratecenter and province.
cancelDID	- Deletes a specific DID from your Account.
connectDID	- Connects a specific DID to a specific Reseller Client Sub Account
delCallback	- Deletes a specific Callback from your Account.
delCallerIDFiltering	- Deletes a specific CallerID Filtering from your Account.
delCallHunting	- Deletes a specific Call Hunting from your Account.
delConference	- Deletes a specific Conference from your Account.
delConferenceMember	- Deletes a specific Member profile from your Account.
delSequences	- Deletes a specific Sequence from your Account.
delClient	- Deletes a specific reseller client from your Account.
delDISA	- Deletes a specific DISA from your Account.
deleteSMS	- Deletes a specific SMS from your Account.
deleteMMS	- Deletes a specific MMS from your Account.
delForwarding	- Deletes a specific Forwarding from your Account.
delIVR	- Deletes a specific IVR from your Account.
delPhonebook	- Deletes a specific Phonebook from your Account.

delPhonebookGroup	- Deletes a specific Phonebook group from your Account.
delQueue	- Deletes a specific Queue from your Account.
delRecording	- Deletes a specific Recording from your Account.
delRingGroup	- Deletes a specific Ring Group from your Account.
delSIPURI	- Deletes a specific SIP URI from your Account.
delStaticMember	- Deletes a specific Static Member from Queue.
delTimeCondition	- Deletes a specific Time Condition from your Account.
getCallbacks	- Retrieves a list of Callbacks if no additional parameter is provided. - Retrieves a specific Callback if a Callback code is provided.
getCallerIDFiltering	- Retrieves a list of CallerID Filterings if no additional parameter is provided. - Retrieves a specific CallerID Filtering if a CallerID Filtering code is provided.
getCallHuntings	- Retrieves a list of Call Huntings if no additional parameter is provided. - Retrieves a specific Call Huntings if a Call Hunting code is provided.
getCarriers	- Retrieves a list of Carriers for Vanity Numbers if no additional parameter is provided. - Retrieves a specific Carrier for Vanity Numbers if a carrier code is provided.
getDIDCountries	- Retrieves a list of Countries for International DIDs if no country code is provided. - Retrieves a specific Country for International DIDs if a country code is provided.
getDIDsCAN	- Retrives a list of Canadian DIDs by Province and Ratecenter.
getDIDsInfo	- Retrieves information from all your DIDs if no additional parameter is provided. - Retrieves information from Reseller Client's DIDs if a Reseller Client ID is provided. - Retrieves information from Sub Account's DIDs if a Sub Account is provided. - Retrieves information from a specific DID if a DID Number is provided. - Retrieves SMS information from a specific DID if the SMS is available.
getDIDsInternationalGeographic	- Retrieves a list of International Geographic DIDs by Country.
getDIDsInternationalNational	- Retrieves a list of International National DIDs by Country.
getDIDsInternationalTollFree	- Retrieves a list of International TollFree DIDs by Country.
getDIDsUSA	- Retrives a list of USA DIDs by State and Ratecenter.
getDIDvPRI	- Retrives the list of DIDs assigned to the VPRI.
getDISAs	- Retrieves a list of DISAs if no additional parameter is provided. - Retrieves a specific DISA if a DISA code is provided.
getForwardings	- Retrieves a list of Forwardings if no additional parameter is provided. - Retrieves a specific Forwarding if a fwd code is provided.
getInternationalTypes	- Retrieves a list of Types for International DIDs if no additional parameter is provided. - Retrieves a specific Types for International DIDs if a type code is provided.
getIVRs	- Retrieves a list of IVRs if no additional parameter is provided. - Retrieves a specific IVR if a IVR code is provided.
getJoinWhenEmptyTypes	- Retrieves a list of 'JoinWhenEmpty' Types if no additional parameter is provided. - Retrieves a specific 'JoinWhenEmpty' Types if a type code is provided.
getMMS	- Retrieves a list of MMS messages by: date range, mms type, DID number, and contact.
getMediaMMS	- Retrieves media files from the message.
getPhonebook	- Retrieves a list of Phonebook entries if no additional parameter is provided. - Retrieves a list of Phonebook entries if a name is provided. - Retrieves a specific Phonebook entry if a Phonebook code is provided. - Retrieves a list of Phonebook entries if a phonebook group name is provided. - Retrieves a list of Phonebook entries if a phonebook group code is provided.
getPhonebookGroups	- Retrieves a list of Phonebook groups if no additional parameter is provided. - Retrieves a list of Phonebook groups if a name is provided. - Retrieves a specific Phonebook group if a group ID is provided.
getPortability	- Shows if a DID Number can be ported into our network. - Display plans and rates available if the DID Number can be ported into our network.
getProvinces	- Retrieves a list of Canadian Provinces.
getQueues	- Retrieves a list of Queue entries if no additional parameter is provided. - Retrieves a specific Queue entry if a Queue code is provided.
getRateCentersCAN	- Retrieves a list of Canadian Ratecenters by Province.
getRateCentersUSA	- Retrieves a list of USA Ratecenters by State.
getRecordings	- Retrieves a list of Recordings if no additional parameter is provided. - Retrieves a specific Recording if a Recording code is provided.
getRecordingFile	- Retrieves a specific Recording File data in Base64 format.
getRingGroups	- Retrieves a list of Ring Groups if no additional parameter is provided. - Retrieves a specific Ring Group if a ring group code is provided.
getRingStrategies	- Retrieves a list of Ring Strategies if no additional parameter is provided. - Retrieves a specific Ring Strategy if a ring strategy code is provided.
getSIPURIs	- Retrieves a list of SIP URIs if no additional parameter is provided. - Retrieves a specific SIP URI if a SIP URI code is provided.
getSMS	- Retrieves a list of SMS messages by: date range, sms type, DID number, and contact.
getStates	- Retrieves a list of USA States.
getStaticMembers	- Retrieves a list of Static Members from a queue if no additional parameter is provided. - Retrieves a specific Static Member from a queue if Queue ID and Member ID are provided
getTimeConditions	- Retrieves a list of Time Conditions if no additional parameter is provided. - Retrieves a specific Time Condition if a time condition code is provided.
getVoicemailSetups	- Retrieves a list of Voicemail Setup Options if no additional parameter is provided. - Retrieves a specific Voicemail Setup Option if a voicemail setup code is provided.

getVoicemailAttachmentFormats	- Retrieves a list of Email Attachment Format Options if no additional parameter is provided. - Retrieves a specific Email Attachment Format Option if a format value is provided.
orderDID	- Orders and Adds a new DID Number to the Account.
orderDIDInternationalGeographic	- Orders and Adds new International Geographic DID Numbers to the Account.
orderDIDInternationalNational	- Orders and Adds new International National DID Numbers to the Account.
orderDIDInternationalTollFree	- Orders and Adds new International TollFree DID Numbers to the Account.
orderDIDVirtual	- Orders and Adds a new Virtual DID Number to the Account.
orderTollFree	- Orders and Adds a new Toll Free Number to the Account.
orderVanity	- Orders and Adds a new Vanity Toll Free Number to the Account.
removeDIDvPRI	- Removes a DID from a VPRI
searchDIDsCAN	- Searches for Canadian DIDs by Province using a Search Criteria.
searchDIDsUSA	- Searches for USA DIDs by State using a Search Criteria.
searchTollFreeCanUS	- Searches for USA/Canada Toll Free Numbers using a Search Criteria. - Shows all USA/Canada Toll Free Numbers available if no criteria is provided.
searchTollFreeUSA	- Searches for USA Toll Free Numbers using a Search Criteria. - Shows all USA Toll Free Numbers available if no criteria is provided.
searchVanity	- Searches for Vanity Toll Free Numbers using a Search Criteria.
sendSMS	- Send a SMS message to a Destination Number.
sendMMS	- Send a MMS message to a Destination Number.
setCallback	- Updates a specific Callback if a callback code is provided. - Adds a new Callback entry if no callback code is provided.
setCallerIDFiltering	- Updates a specific Caller ID Filtering if a filtering code is provided. - Adds a new Caller ID Filtering if no filtering code is provided.
setCallHunting	- Updates a specific Call Hunting if a Call Hunting code is provided. - Adds a new Call Hunting if no Call Hunting code is provided.
setDIDBillingType	- Updates the Billing Plan from a specific DID.
setDIDInfo	- Updates the information from a specific DID.
setDIDPOP	- Updates the POP from a specific DID.
setDIDRouting	- Updates the Routing from a specific DID.
setDIDVoicemail	- Updates the Voicemail from a specific DID.
setDISA	- Updates a specific DISA if a disa code is provided. - Adds a new DISA entry if no disa code is provided.
setForwarding	- Updates a specific Forwarding if a fwd code is provided. - Adds a new Forwarding entry if no fwd code is provided.
setIVR	- Updates a specific IVR if an IVR code is provided. - Adds a new IVR entry if no IVR code is provided.
setPhonebook	- Updates a specific Phonebook entry if a phonebook code is provided. - Adds a new Phonebook entry if no phonebook code is provided.
setPhonebookGroup	- Updates a specific Phonebook group if a phonebook code is provided. - Adds a new Phonebook group if no phonebook group code is provided. - Assigns or modifies group members if a member list is provided
setQueue	- Updates a specific Queue entry if a queue code is provided. - Adds a new Queue entry if no queue code is provided.
setRecording	- Updates a specific Recording File if a Recording ID is provided. - Adds a new Recording file entry if no Recording ID is provided.
setRingGroup	- Updates a specific Ring Group if a ring group code is provided. - Adds a new Ring Group entry if no ring group code is provided.
setSIPURI	- Updates a specific SIP URI if a SIP URI code is provided. - Adds a new SIP URI entry if no SIP URI code is provided.
setSMS	- Enable/Disable the SMS Service for a DID - Change the SMS settings for a DID
setStaticMember	- Updates a specific Member from queue if a Member code is provided. - Adds a new Member to Queue if no Member code is provided.
setTimeCondition	- Updates a specific Time Condition if a time condition code is provided. - Adds a new Time Condition entry if no time condition code is provided.
unconnectDID	- Unconnects specific DID from Reseller Client Sub Account.
Fax	
connectFAX	- Connects a specific FAX DID to a specific Reseller Client Sub Account
unconnectFAX	- Unconnects specific FAX DID from Reseller Client Sub Account.
cancelFaxNumber	- Deletes a specific Fax Number from your Account.
deleteFaxMessage	- Deletes a specific Fax Message from your Account.
delEmailToFax	- Deletes a specific "Email to Fax configuration" from your Account.
delFaxFolder	- Deletes a specific Fax Folder from your Account.
getBackOrders	- Retrieves a list of backorder DIDs if no additional parameter is provided. - Retrieves a specific backorder DID if a backorder DID code is provided.
getFaxProvinces	- Retrieves a list of Canadian Fax Provinces if no additional parameter is provided. - Retrieves a specific Canadian Fax Province if a province code is provided.

getFaxStates	- Retrieves a list of American Fax States if no additional parameter is provided. - Retrieves a specific American Fax State if a state code is provided.
getFaxRateCentersCAN	- Retrieves a list of Canadian Ratecenters by Province.
getFaxRateCentersUSA	- Retrieves a list of USA Ratecenters by State.
getFaxNumbersInfo	- Retrieves a list of Fax Numbers.
getFaxNumbersPortability	- Shows if a Fax Number can be ported into our network
getFaxMessages	- Retrieves a list of Fax Messages. - Retrieves a specific Fax Message if a Fax Message ID is provided.
getFaxMessagePDF	- Retrieves a Base64 code of the Fax Message to create a PDF file.
getFaxFolders	- Retrieves a list of Fax Folders from your account.
getEmailToFax	- Retrieves a list of "Email to Fax configurations" from your account if no additional parameter is provided. - Retrieves a specific "Email to Fax configuration" from your account if a ID is provided.
mailFaxMessagePDF	- Send a Fax Message attached as a PDF file to an email destination.
moveFaxMessage	- Moves a Fax Message to a different folder.
orderFaxNumber	- Orders and Adds a new Fax Number to the Account.
setFaxFolder	- Create or update the information of a specific Fax Folder.
setEmailToFax	- Create or update the information of a specific "Email to Fax configuration".
searchFaxAreaCodeCAN	- Retrieves a list of Canadian Ratecenters searched by Area Code.
searchFaxAreaCodeUSA	- Retrieves a list of USA Ratecenters searched by Area Code.
setFaxNumberInfo	- Updates the information from a specific Fax Number.
setFaxNumberEmail	- Updates the email configuration from a specific Fax Number.
setFaxNumberURLCallback	- Updates the url callback configuration from a specific Fax Number.
sendFaxMessage	- Send a Fax message to a Destination Number.
e911	
e911AddressTypes	- Retrieves a list of e911 Address Types if no additional parameter is provided. - Retrieves a specific e911 Address Type if an Address code is provided.
e911Cancel	- Cancel the e911 Service from a specific DID.
e911Info	- Retrieves the e911 information from a specific DID.
e911Provision	- Subscribes your DID to the e911 Emergency Services.
e911ProvisionManually	- Subscribes your DID to the e911 Emergency Services. - All e911 information will be validated by the VoIP.ms staff.
e911Update	- Updates the Information from your e911 Emergency Services Subscription.
e911Validate	- Validates your e911 information in order to start your e911 Emergency Services Subscription.
Local Number Portability (LNP)	
addLNPPort	- Add one or more numbers to start a portability process.
addLNPPFile	- Add an invoice file to a portability process.
getLNPPStatus	- Retrieve the current status of a given portability process.
getLNPNNotes	- Retrieve the list of notes from the given portability process.
getLNPListStatus	- Retrieve the list of possible status of a portability process.
getLNPList	- Retrieve the full list of all your portability processes.
getLNPDDetails	- Retrieve the details of a given portability process.
getLNPAAttachList	- Retrieve the list of invoice (attached) files from a given portability process.
getLNPAAttach	- Retrieve the details of an attached invoice.
Voicemail	
createVoicemail	- Adds a new Voicemail entry to your Account
delMessages	- Deletes all messages in all servers from a specific Voicemail from your Account
delMemberFromConference	- Removes a member profile from a specific Conference from your Account
delVoicemail	- Deletes a specific Voicemail from your Account
getPlayInstructions	- Retrieves a list of Play Instructions modes if no additional parameter is provided. - Retrieves a specific Play Instructions mode if a play code is provided.
getTimezones	- Retrieves a list of Timezones if no additional parameter is provided. - Retrieves a specific Timezone if a timezone code is provided.
getVoicemails	- Retrieves a list of Voicemails if no additional parameter is provided. - Retrieves a specific Voicemail if a voicemail code is provided.
getVoicemailFolders	- Retrieves a list of default Voicemail Folders if no additional parameter is provided. - Retrieves a list of Voicemail Folders within a mailbox if mailbox parameter is provided. - Retrieves a specific Folder if a folder name is provided.
getVoicemailMessageFile	- Retrieves a specific Voicemail Message File in Base64 format.
getVoicemailMessages	- Retrieves a list of Voicemail Messages if mailbox parameter is provided. - Retrieves a list of Voicemail Messages in a Folder if a folder is provided. - Retrieves a list of Voicemail Messages in a date range if a from and to are provided.
getVPRIIs	- Retrieves a list of vpri.

markListenedVoicemailMessage	- Mark a Voicemail Message as Listened or Unlistened. - If value is 'yes', the voicemail message will be marked as listened and will be moved to the Old Folder. - If value is 'no', the voicemail message will be marked as not-listened and will be moved to the INBOX Folder.
markUrgentVoicemailMessage	- Mark Voicemail Message as Urgent or not Urgent. - If value is 'yes', the voicemail message will be marked as urgent and will be moved to the Urgent Folder. - If value is 'no', the voicemail message will be unmarked as urgent and will be moved to the INBOX Folder.
moveFolderVoicemailMessage	- Move Voicemail Message to a Destination Folder.
sendVoicemailEmail	- Send a Voicemail Message File to an Email Address.
setVoicemail	- Updates the information from a specific Voicemail.

FUNCTION STATUS AND ERROR CODE

All the API functions return a Status value after execution.

- If the function executes without any problems, then status value will be "success".
- If the function finds problems, the status will return an error code giving you an idea of what went wrong.

You can find all possible error codes and short explanations in the following table

Error Codes	
account_with_dids	The Account has DIDs assigned to it.
api_limit_exceeded	API requests limit per minute has been reached
api_not_enabled	API has not been enabled or has been disabled
cancel_failed	The cancellation wasn't completed.
can_have_only_one_profile_without_pin	The conference can just have one profile member without pin
conference_member_relation_not_found	There is no relation between the profile member and the conference.
did_in_use	DID Number is already in use
did_limit_reached	You have reached the maximum number of DID numbers allowed for your account type. Please contact our team if you have a specific use case or if you would like to upgrade to a Business account.
uplicated_name	There is already another entry with this name
uplicated_pin	The given pin has been duplicated
e911_disabled	DID e911 service it's not enabled.
e911_pending	DID e911 service has been requested and is in validation process.
error_deleting_msg	Error when deleting message
error_moving_msg	Error when move the voicemail message to folder
exceeds_file_size	The file exceeds the limite size allowed.
existing_did	You can't set a callback to an existing VoIP.ms DID number
forwards_exceeded	Your account is limited to 4 forward entries
invalid_account	This is not a valid account
invalid_address	Address is missing or the format is invalid.
invalid_admin	This is not a valid admin
invalid_agent_ring_timeout	This is not a valid Agent ring time out value
invalid_allowedcodecs	One of the codecs provided is invalid Format and Values: ulaw;g729;gsm;all
invalid_announce_join_leave	This is not a valid "Announce join leave"
invalid_announce_only_user	This is not a valid "Announce only user"
invalid_announce_position_frequency	This is not a valid Announce position frequency
invalid_announce_round_seconds	This is not a valid "Announce round seconds"
invalid_announce_user_count	This is not a valid "Announce user count"
invalid_area_code	this is not a valid Area Code.
invalid_attachid	The given ID is invalid or doesn't exist.
invalid_attachmessage	this is not a valid AttachMessage Should be: yes/no
invalid_attach_file	Valid formats: PDF, MS Word, BMP, JPG
invalid_authtype	This is not a valid Auth Type
invalid_authtype_h323	You must select IP Auth to use H.323
invalid_authtype_iax2	You must use User/Password Authentication for IAX2
invalid_balancemanagement	This is not a valid BalanceManagement
invalid_base_recording	This is not a valid recording path
invalid_billingtype	This is not a valid Billing Type Allowed values: 1 = PerMinute, 2 = Flat
invalid_callback	This is not a valid Callback
invalid_callback_enable	This is not a valid Callback enable value

invalid_callback_retry	This is not a valid Callback retry
invalid_callerid	This is not a valid CallerID
invalid_calleridprefix	This is not a valid CID Prefix, lenght should be less than 20 chars
invalid_callerid_override	This is not a valid CallerID Override
invalid_callhunting	This is not a valid Call Hunting
invalid_callparking	This is not a valid Call Parking
invalid_callrecording	This is not a valid Call recording
invalid_call_type	Call Type is not valid.
invalid_canada_routing	This is not a valid Canada Route
invalid_carrier	This is not a valid Carrier
invalid_charge	This is not a valid Charge
invalid_city	City is missing or the format is invalid.
invalid_client	This is not a valid Client
invalid_cnam	This is not a valid CNAM Should be: 1/0
invalid_codec	This is not a valid Codec
invalid_conference	This is not a valid Conference ID
invalid_contact	This is not a valid Contact Number
invalid_country	Country is missing or the format is invalid, must be in format ISO 3166-1 alpha-2, example: US, CA, etc. (You can use the values returned by the method getCountries)
invalid_countryid	This is not a valid Country ID
invalid_credentials	Username or Password is incorrect
invalid_date	This is not a valid date Format is: yyyy-mm-dd
invalid_daterange	Date Range should be 92 days or less
invalid_datetime	This is not a valid datetime Format is: yyyy-mm-dd hh:mm:ss
invalid_date_from	The "From" date should be prior to the "To" date.
invalid_dayrange	This is not a valid Day Range
invalid_delay_before	This is not a valid DelayBefore
invalid_deletemessage	This is not a valid DeleteMessage Should be: yes/no
invalid_description	This is not a valid Description
invalid_destination	This is not a valid Destination
invalid_destination_folder	This is not a valid Destination Folder
invalid_devicetype	This is not a valid Device Type
invalid_dialtime	This is not a valid Dialtime
invalid_did	This is not a valid DID
invalid_digits	These are not valid DigitsOrderDIDVirtual: Digits must be 3 numbers
invalid_digit_timeout	This is not a valid DigitTimeOut
invalid_disa	This is not a valid DISA
invalid_diversion_header	This is not a valid Diversion Header. It must be a numeric value, accepting only 0 or 1.
invalid_drop_silence	This is not a valid "drop silence" value
invalid_dst	This is not a valid Destination Number
invalid_dtmfmode	This is no a valid DTMF Mode
invalid_dtmf_digits	This is no a valid DTMF digit
invalid_email	This is not a valid email or email is already in database
invalid_email_attachment_format	This is not a valid format value
invalid_email_enable	This is not a valid email enable value
invalid_enable_ip_restriction	This is not a valid Enable IP Restriction value
invalid_enable_pop_restriction	This is not a valid Enable POP Restriction value
invalid_endhour	This is not a valid End Hour
invalid_endminute	This is not a valid End Minute
invalid_extension	This is not a valid extension Extension can only contain digits
invalid_failover_header	This is not a valid failover header Should be: account/vm/fwd/none
invalid_fax_id	This is not a valid Fax Message ID
invalid_file	This is not a valid File
invalid_filter	This is not a valid Filter
invalid_firstname	First name is missing or the format is invalid.
invalid_foc_enddate	Invalid date format, must be: YYYY-mm-dd. Example: 2018-02-22
invalid_foc_startdate	Invalid date format, must be: YYYY-mm-dd. Example: 2018-02-22

invalid_folder	This is not a valid Folder
invalid_folder_id	This is not a valid Fax Folder ID
invalid_forwarding	This is not a valid forwarding
invalid_forwarding_did	Forwarding to the same did is not allowed
invalid_forward_enable	This is not a valid forward enable value
invalid_frequency_announcement	This is not a valid Frequency announce
invalid_from_number	This is not a valid sender number.
invalid_fullname	This is not a valid Full Name
invalid_id	This is not a valid ID
invalid_if_announce_position_enabled_report_e	This is not a Report estimated hold time type
invalid_internaldialtime	This is not a valid Internal Dialtime Should be: 1 to 60
invalid_internalvoicemail	This is not a valid Internal Voicemail
invalid_internationalroute	This is not a valid International Route
invalid_invoice_type	Invalid invoice type, possible values: 0 = US, 1 = CAN.
invalid_ip	This is an invalid IP
invalid_ip_auth	Do not provide an IP address for User/Pass Authentication
invalid_ip_iax2	Do not provide an IP address for IAX2
invalid_ivr	This is not a valid IVR
invalid_jitter_buffer	This is not a valid "jitter buffer" value
invalid_join_announcement	This is not a valid 'Join Announcement' Type for a Queue
invalid_join_empty_type	This is not a valid 'JoinWhenEmpty' Type for a Queue
invalid_language	This is not a valid Language Should be: es/en/fr
invalid_lastname	Lastname is missing or the format is invalid.
invalid_listened	This is not a valid Listened value
invalid_location	This is not a valid Location
invalid_lockinternational	This is not a valid Lock International
invalid_mailbox	This is not a valid mailbox
invalid_maximum_callers	This is not a valid maximum callers value
invalid_maximum_wait_time	This is not a valid maximum wait time value
invalid_max_expiry	This is not a valid Max Expiry (value must be between 60 and 3600 seconds)
invalid_member	This is not a valid Member
invalid_member_delay	This is not a valid Member Delay
invalid_message_num	This is not a valid Voicemail Message Number
invalid_method	This is not a valid Method
invalid_minute	This is not a valid Minute Rate
invalid_mixed_numbers	Toll-free numbers and local numbers can not be mixed in the same order.
invalid_monthly	This is not a valid Montly Fee
invalid_musiconhold	This is not a valid Music on Hold
invalid_name	This is not a valid name, Alphanumeric Only
invalid_nat	This is not a valid NAT
invalid_note	This is not a valid Note, lenght should be less than 50 chars
invalid_number	This is not a valid Number
invalid_numbermembers	The element format of multiple data is not correct or it size does not match with other elements
invalid_number_canadian	You have entered a Canadian number (not valid in this portability process).
invalid_number_exist	The number is already in our network
invalid_number_fax	The Fax number can not be ported into our network
invalid_number_porttype	You have entered a local number (not valid in this portability process)
invalid_number_us	You have entered a USA number (not valid in this portability process).
invalid_order	This is not a valid "order" value
invalid_package	This is not a valid Package
invalid_password	This is not a valid passwordVoicemail: Must be 4 Digits SubAccounts: More than 6 chars, Must Contain Alphanumeric and !#\$%&/()=?*[]_ : . , { } + -
invalid_password_auth	Do not provide a Password for IP Authentication
invalid_password_illegal_characters	This is not a valid password (Allowed characters: Alphanumeric and ! # \$ % & / () = ? * [] _ : . , { } + -)
invalid_password_lessthan_8characters_long	This is not a valid password (Less than 8 characters long)
invalid_password_missing_lowercase	This is not a valid password (Missing lower case character)
invalid_password_missing_number	This is not a valid password (Missing a number)

invalid_password_missing_uppercase	This is not a valid password (Missing upper case character)
invalid_pause	This is not a valid Pause
invalid_payment	This is not a valid Payment
invalid_phonebook	This is not a valid Phonebook
invalid_phonenumber	This is not a valid Phone Number
invalid_pin	This is not a valid PIN
invalid_pin_number	Must provide the account PIN number.
invalid_playinstructions	This is not a valid PlayInstructions Should be: u/su
invalid_pop_restriction	This is not a valid POP Restriction
invalid_portingid	The given ID is invalid or doesn't exist.
invalid_porttype	Must provide a valid port type.
invalid_port_status	The status code is invalid. (You can use the values returned by the method getListStatus)
invalid_priority	This is not a valid Priority
invalid_priority_weight	This is not valid weight/priority value
invalid_protocol	This is not a valid Protocol
invalid_provider_account	You must provide your account # with the current provider
invalid_provider_name	You must provide the service provider name
invalid_province	This is not a valid Province
invalid_quantity	This is not a valid quantity
invalid_query	This is not a valid Query
invalid_queue	This is not a valid Queue
invalid_quiet	This is not a valid "quiet" value
invalid_recording	This is not a valid recording
invalid_recording_sound_error_menu	"error menu" is not a valid recording
invalid_recording_sound_get_pin	"get pin" is not a valid recording
invalid_recording_sound_has_joined	"has_joined" is not a valid recording
invalid_recording_sound_has_left	"has_left" is not a valid recording
invalid_recording_sound_invalid_pin	"invalid pin" is not a valid recording
invalid_recording_sound_join	"join" is not a valid recording
invalid_recording_sound_kicked	"kicked" is not a valid recording
invalid_recording_sound_leave	"leave" is not a valid recording
invalid_recording_sound_locked	"locked" is not a valid recording
invalid_recording_sound_locked_now	"locked now" is not a valid recording
invalid_recording_sound_muted	"muted" is not a valid recording
invalid_recording_sound_only_one	"only one" is not a valid recording
invalid_recording_sound_only_person	"only person" is not a valid recording
invalid_recording_sound_other_in_party	"other in party" is not a valid recording
invalid_recording_sound_participants_muted	"participants muted" is not a valid recording
invalid_recording_sound_participants_unmuted	"participants unmuted" is not a valid recording
invalid_recording_sound_place_into_conference	"place into conference" is not a valid recording
invalid_recording_sound_there_are	"there are" is not a valid recording
invalid_recording_sound_unlocked_now	"unlocked now" is not a valid recording
invalid_recording_sound_unmuted	"unmuted" is not a valid recording
invalid_record_calls	Record calls is not valid.
invalid_report_hold_time_agent	This is not a valid Report hold time agent
invalid_resellerclient	This is not a valid Reseller Client
invalid_resellernextbilling	This is not a valid Reseller Next Billing date, date should not be set in the past.
invalid_resellerpackage	This is not a valid Reseller Package
invalid_response_timeout	This is not a valid ResponseTimeOut
invalid_retry_timer	This is not a valid Retry timer
invalid_ringgroup	This is not a valid Ring group
invalid_ring_inuse	This is not a valid Ring in use value
invalid_route	This is not a valid Route
invalid_routing_header	This is not a valid Routing header Should be: account/vm/fwd
invalid_rtp_hold_timeout	This is not a valid RTP Hold Time Out (value must be between 1 and 3600 seconds)
invalid_rtp_timeout	This is not a valid RTP Time Out (value must be between 1 and 3600 seconds)

invalid_saycallerid	This is not a valid SayCallerID Should be: yes/no
invalid_saytime	This is not a valid SayTime Should be: yes/no
invalid_security_code	This is not a valid Security Code. Should be alphanumeric.
invalid_serverpop	This is not a valid Server POP
invalid_setup	This is not a valid Setup Fee
invalid_silence_threshold	This is not a valid "silence threshold" value
invalid_sipuri	This is not a valid SIPURI
invalid_sip_traffic	This is not a valid Encrypted SIP Traffic value
invalid_skippassword	This is not a valid skippassword Should be: 1/0 - or - yes/no
invalid_smpp_password	This is not a valid SMPP Password
invalid_smpp_url	This is not a valid SMPP URL
invalid_smpp_username	This is not a valid SMPP Username
invalid_sms	This is not a valid SMS
invalid_sms_forward	This is not a valid SMS forward
invalid_snn	Must provide the 4 last digits of the SSN.
invalid_speed_dial	This is not a valid Speed Dial
invalid_starthour	This is not a valid Start Hour
invalid_startminute	This is not a valid Start Minute
invalid_start_muted	This is not a valid Start Muted
invalid_state	This is not a valid State
invalid_statement_name	Statement Name is missing or the format is invalid.
invalid_strategy	This is not a valid Ring Strategy
invalid_street_name	This is not a valid Street Name
invalid_street_number	This is not a valid Street Number
invalid_talking_threshold	This is not a valid "talking threshold" value
invalid_talk_detection	This is not a valid talk detection value
invalid_tfnnumber_porttype	You have entered a toll-free number (not valid in this portability process).
invalid_thankyou_for_your_patience	This is not a valid Thankyou for your patience value
Invalid_threshold	This is not a valid Threshold Amount. The Threshold Amount should be between 1 and 250
invalid_timecondition	This is not a valid Time Condition
invalid_timeout	This is not a valid timeout
invalid_timerange	This is not a valid Timer Range
invalid_timezone	This is not a valid TimezoneCDR and resellerCDR: Must be numeric Voicemail: Values from getTimezone
invalid_to_number	This is not a valid destination number
invalid_transcription_email	Transcription email is not valid
invalid_transcription_locale	Transcription locale is not valid.
invalid_type	This is not a valid Type
invalid_urgent	This is not valid urgent value
invalid_username	This is not a valid Username
invalid_voicemailsetup	This is not a valid voicemail
invalid_voice_announcement	This is not a valid Voice announce
invalid_weekdayend	This is not a valid Week End
invalid_weekdaystart	This is not a valid Week Start
invalid_wrapup_time	This is not a valid Wrapup time
invalid_zip	Zip Code is missing or the format is invalid.
ip_not_enabled	This IP is not enabled for API use
limit_reached	You have reached the maximum number of messages allowed per day. - SMS limit using the API. - Fax limit applies using any method.
max_phonebook	Your account is limited to 8 SIP, IAX or SIP URI members
members_exceeded	You have reached the maximum allowed entries for the Phonebook
member_already_included	The member has been included already
message_empty	The SMS Message is empty
message_not_found	The voicemail message was not found
method_maintenance	This API method is under maintenance
mismatch_email_confirm	e-mail confirm does not match with e-mail
mismatch_password_confirm	Pasword confirm does not match with Password
missing_account	Account was not provided

missing_address	Address was not provided
missing_agent_ring_timeout	Agent ring time out was not provided
missing_allowedcodecs	Allowed Codecs were not provided
missing_attachmessage	AttachMessage was not provided
missing_authtype	Auth Type was not provided
missing_balancemanagement	BalanceManagemente was not provided
missing_billingtype	Billing Type was not provided
missing_callback	Callback was not provided
missing_callerid	CallerID was not provided
missing_callhunting	Call hunting was not provided
missing_callparking	Call Parking was not provided
missing_callrecording	Call recording was not provided
missing_carrier	Carrier was not provided
missing_charge	Charge was not provided.
missing_choices	Choices was not provided
missing_city	City was not provided
missing_client	Client was not provided
missing_cnam	CNAM was not provided
missing_codec	Codec was not provided
missing_conference	Conference was not provided
missing_country	Country was not provided
missing_countryid	Country ID was not provided
missing_credentials	Username or Password was not provided
missing_datetime	DateTime value was not provided
missing_delay_before	DelayBefore was not provided
missing_deletemessage	DeleteMessage was not provided
missing_description	Description was not provided
missing_devicetype	Device Type was not provided
missing_dialtime	Dialtime was not provided
missing_did	DID was not provided
missing_digits	Digits were not provided
missing_digit_timeout	DigitTimeOut was not provided
missing_disa	DISA was not provided
missing_dtmfmode	DTMF Mode was not provided
missing_email	e-mail was not provided
missing_email_confirm	e-mail confirm was not provided
missing_enable	Enable was not provided
missing_endhour	End Hour was not provided
missing_endminute	End Minute was not provided
missing_failover_busy	Failover Busy was not provided
missing_failover_noanswer	Failover NoAnswer was not provided
missing_failover_unreachable	Failover Unreachable was not provided
missing_file	File was not provided
missing_filter	Filter was not provided
missing_firstname	Firstname was not provided
missing_folder	folder was not provided
missing_forwarding	Forwarding was not provided
missing_from_date	From date was not provided
missing_fullname	Full Name was not provided
missing_id	ID was not provided
missing_if_announce_position_enabled_report_e	If announce position enabled report estimated hold time' type was not provided
missing_internationalroute	International Route was not provided
missing_ip	You need to provide an IP if you select IP Authentication Method
missing_ip_h323	You must enter an IP Address for H.323
missing_ip_restriction	IP Restriction was not provided
missing_ivr	IVR was not provided

missing_join_when_empty	JoinWhenEmpty' type was not provided
missing_language	Language was not provided
missing_lastname	LastName was not provided
missing_leave_when_empty	LeaveWhenEmpty' type was not provided
missing_length	Length was not provided
missing_listened	Listened code was not provided
missing_location	Location was not provided
missing_lockinternational	Lock International was not provided
missing_mailbox	Mailbox was not provided
missing_member	Member was not provided
missing_members	You need at least 1 member to create a ring group
missing_message_num	Voicemail message number was not provided
missing_method	Method must be provided when using the REST/JSON API
missing_minute	Minute Rate was not provided
missing_monthly	Monthly Fee was not provided
missing_musiconhold	Music on Hold was not provided
missing_name	Name was not provided
missing_nat	NAT was not provided
missing_number	Number was not provided
missing_numbers	You must enter at least one valid phone number.
missing_package	Package was not provided
missing_params	Required parameters were not provided
missing_password	Password was not provided
missing_password_confirm	Password Confirm was not provided
missing_payment	Payment was not provided.
missing_phonebook	Phonebook was not provided
missing_phonenumber	Phone Number was not provided
missing_pin	PIN was not provided
missing_playinstructions	PlayInstructions was not provided
missing_pop_restriction	POP Restriction was not provided
missing_priority	Priority was not provided
missing_priority_weight	Priority/Weight was not provided
missing_protocol	Protocol was not provided
missing_province	Province was not provided
missing_query	Query was not provided
missing_recording	Recording was not provided
missing_report_hold_time_agent	Report hold time agent was not provided
missing_resellerclient	Provide a Reseller Client or don't provide a Reseller Package
missing_resellerpackage	Provide a Reseller Package or don't provide a Reseller Client
missing_response_timeout	ResponseTimeOut was not provided
missing_ringgroup	Ring group was not provided
missing_ring_inuse	Ring in use was not provided
missing_ring_strategy	Ring strategy was not provided
missing_route	Route was not provided
missing_routing	Routing was not provided
missing_saycallerid	SayCallerID was not provided
missing_saytime	SayTime was not provided
missing_serverpop	Server POP was not provided
missing_setup	Setup Fee was not provided
missing_sipuri	SIPURI was not provided
missing_skippassword	SkipPassword was not provided
missing_sms	SMS was not provided
missing_speed_dial	Speed Dial was not provided
missing_start	Start date was not provided
missing_starthour	Start Hour was not provided
missing_startminute	Start Minute was not provided

missing_state	State was not provided
missing_street_name	Street Name was not provided
missing_street_number	Street Number was not provided
missing_thankyou_for_your_patience	Thankyou for your patience was not provided
missing_timecondition	Time Condition was not provided
missing_timeout	Timeout was not provided
missing_timezone	Timezone was not provided
missing_to_date	To date was not provided
missing_transcription_email	Transcription email is required.
missing_transcription_locale	Transcription locale is required.
missing_type	Type was not provided
missing_urgent	Urgent code was not provided
missing_uri	URI was not provided
missing_username	Username was not provided
missing_voicemailsetup	Voice mail setup was not provided
missing_weekdayend	Week End was not provide
missing_weekdaystart	Week Start was not provided
missing_zip	Zip Code was not provided
moving_fail	The Fax Message was not moved
name_toolong	The name exceeds character size limit
non_sufficient_funds	Your account does not have sufficient funds to proceed
note_toolong	The note exceeds character size limit
no_account	There are no accounts
no_attachments	Theres no attachments records to show.
no_base64file	File not encoded in base64
no_callback	There are not Callbacks
no_callhunting	There are no Call Huntings
no_callparking	There are no Call Parking
no_callstatus	No Call Status was provided. One of the following parameters needs to be set to "1": answered, noanswer, busy, failed
no_cdr	There are no CDR entries for the filter
no_change_billingtype	Imposible change DID billing plan
no_client	There are no Clients
no_conference	There are no Conferences
no_did	There are no DIDs
no_disa	There are no DISAs
no_filter	There are no Filters
no_forwarding	There was no Forwarding
no_ivr	There are no ivr
no_mailbox	There are no Mailboxes
no_member	There are no Static Members
no_message	There are no Fax Message(s)
no_messages	There are no Voicemail Message(s)
no_numbers	There are no Fax Numbers
no_package	there are no Packages
no_phonebook	There are no Phonebook entries
no_provision	E911 service wasn't activated, this response comes with a description of the error.
no_provision_update	E911 service wasn't updated, this response comes with a description of the error.
no_queue	There are no Queue entries
no_rate	There are no Rates
no_recording	There are no recordings
no_ringgroup	There are no Ring groups
no_sequences	No sequence has been found
no_sipuri	There are no SIP URIs
no_sms	There are no SMS messages
no_timecondition	There are no Time Conditions
order_failed	The order wasn't completed.

problem_sending_mail	There was a problem sending an email.
provider_outofservice	One of our providers is out of service
recording_in_use_caller_id_filtering	You have a Caller ID Filtering using this Recording
recording_in_use_caller_timecondition	You have a Time Condition using this Recording
recording_in_use_did	You have a DID using this Recording
recording_in_use_ivr	You have an IVR using this Recording
recording_in_use_queue	You have a Calling Queue using this Recording
repeated_ip	You already have a Subaccount using this IP and Protocol
reserved_ip	This is a reserved IP used by VoIP.ms or other Companies
rtp_timeout_greater_than_rtp_hold_timeout	RTP Time Out can't be greater than RTP Hold Time Out
same_did_billingtype	The Billing Type provided and DID billing type are the same
sent_fail	The Fax Message it wasn't send.
sipuri_in_phonebook	This SIPURI can't be deleted, it is mapped in the phonebook
sms_apply_regulations	The number was not updated due to SMS regulations, please contact customer service for more information
sms_failed	The SMS message was not sent
sms_toolong	The SMS message exceeds 160 characters
sms_wait_message	SMS was not (Enabled/Disabled) for this DID, please wait a minute before you try again.
tls_error	Theres was a TLS error, please try later.
Unable_to_purchase	Unable to purchase DIDs
unavailable_info	The information you requested is unavailable at this moment
unsifficient_stock	Theres no sufficient stock to complete the order.
used_description	You already have a record with this Description
used_email	You already have an entry with this Email
used_extension	You already have a subaccount using this extension
used_filter	You already have a record with this Filter
used_ip	There is already another customer using this IP Address
used_name	You already have an entry using this name
used_number	You already have a record with this Number
used_password	This password has been used previously by this account.
used_speed_dial	You have an entry with this Speed Dial
used_username	You already have a subaccount using this Username.
weak_password	This Password is too weak or too common

GENERAL FUNCTIONS

The following table explains the additional parameters needed by the General Functions and provides an example of their output when status is 'success'.

addMemberToConference			
Parameters	member conference => [Required] => [Required] Specific Member ID (Example: 6547) Specific Conference ID (Example: 234)		
Output	Array ([status] => success [member] => 4234)		
getBalance			
Parameters	advanced => True for Calls Statistics		
Output	Array ([status] => success [balance] => Array ([current_balance] => 112.192790 [spent_total] => 21.05321 [calls_total] => 2788 [time_total] => 10:17:48 [spent_today] => 0.1205 [calls_today] => 5 [time_today] => 10:50))		
getConference			

Parameters	conference => Code for a specific Conference (Example: 1599)
Output	<pre>Array ([status] => success [conference] => Array ([0] => Array ([conference] => 1599 [name] => testing [description] => testing conference [max_members] => 1 [sound_join] => 0 [sound_leave] => 0 [sound_has_joined] => 0 [sound_has_left] => 0 [sound_kicked] => 0 [sound_muted] => 0 [sound_unmuted] => 0 [sound_only_person] => 0 [sound_only_one] => 0 [sound_there_are] => 0 [sound_other_in_party] => 0 [sound_place_into_conference] => 0 [sound_get_pin] => 0 [sound_invalid_pin] => 0 [sound_locked] => 0 [sound_locked_now] => 0 [sound_unlocked_now] => 0 [sound_error_menu] => 0 [sound_participants_muted] => 0 [sound_participants_unmuted] => 0 [language] => en [members] => 30;31;32))))</pre>
getConferenceMembers	
Parameters	member => Code for a specific Member profile (Example: 1599)
Output	<pre>Array ([status] => success [members] => Array ([0] => Array ([member] => 1599 [name] => testing [description] => testing description [pin] => 1234 [announce_join_leave] => no [admin] => no [start_muted] => no [announce_user_count] => no [announce_only_user] => yes [moh_when_empty] => guitar_alchemy [quiet] => no [announcement] => 0 [drop_silence] => yes [talking_threshold] => 140 [silence_threshold] => 2500 [talk_detection] => yes [jitter_buffer] => yes))))</pre>
getConferenceRecordings	
Parameters	conference => [Required] ID for a specific Conference (Example: 5356) date_from => Start Date for Filtering Transactions (Example: '2016-06-03') date_to => End Date for Filtering Transactions (Example: '2016-06-04')
Output	<pre>Array ([status] => success [recordings] => Array ([0] => Array ([did] => 1234567890 [recording] => 1543338379 [conference] => 1234 [duration] => 6 [date] => 2018-11-27 12:06:19))))</pre>
getConferenceRecordingFile	
Parameters	conference => [Required] ID for a specific Conference (Example: 5356) recording => [Required] ID for a specific Conference Recording (Example: 1543338379)

Output	<pre>Array ([status] => success [recording] => Array ([0] => Array ([recording] => 1543338379 [data] => Uk1GRqTEAQBXQVZFZm10IBAAAAABAEEAQB8AAIA+AAACABAAZGF0YUYDEA....))))</pre>
getSequences	
Parameters	<pre>sequence => ID for a specific Sequence (Example: 5356) client => [Optional] ID for a specific Reseller Client (Example: 561115)</pre>
Output	<pre>Array ([status] => success [sequences] => Array ([0] => Array ([id] => 1599 [name] => testing [steps] => "[{ 'type': 'rec', 'id': '1234' }, { 'type': 'dtmf', 'codes': '1234' }, { 'type': 'sms', 'from': '1234567890', 'to': '0987654321', 'msg': 'This is a test.' }]"</pre>
getCountries	
Parameters	<pre>country => Code for a specific Country (Example: 'CA')</pre>
Output	<pre>Array ([status] => success [countries] => Array ([0] => Array ([value] => CA [description] => Canada))))</pre>
getIP	
Parameters	No Parameter
Output	<pre>Array ([status] => success [ip] => 174.132.203.114)</pre>
getLanguages	
Parameters	<pre>language => Code for a specific Language (Example: 'en')</pre>
Output	<pre>Array ([status] => success [languages] => Array ([0] => Array ([value] => en [description] => English))))</pre>
getLocales	
Parameters	<pre>locale => Code for a specific Locale Code (Example: 'en-US')</pre>

Output	<pre> Array ([status] => success [locales] => Array ([0] => Array ([value] => ar-EG [description] => Arabic (Egypt), modern standard))) </pre>
getServersInfo	
Parameters	server_pop => POP for a specific Server (Example: 1)
Output	<pre> Array ([status] => success [servers] => Array ([0] => Array ([server_name] => Houston, TX [server_shortcode] => Houston [server_hostname] => houston.voip.ms [server_ip] => 209.62.1.2 [server_country] => USA [server_pop] => 1 [server_recommended] => Yes))) </pre>
getTransactionHistory	
Parameters	date_from => [Required] Start Date for Filtering Transactions (Example: '2016-06-03') date_to => [Required] End Date for Filtering Transactions (Example: '2016-06-04')
Output	<pre> Array ([status] => success [transactions] => Array ([0] => Array ([date] => 2016-06-03 00:03:46 [uniqueid] => 12964421x41098i8c [type] => DID8555555555 [description] => DID Monthly Fee: 8555555555 [ammount] => -0.99))) </pre>

ACCOUNTS FUNCTIONS

The following table explains the additional parameters needed by the Accounts Functions and provides an example of their output when status is 'success'.

createSubAccount	
Parameters	<pre> username => [Required] Username for the Sub Account (Example: 'VoIP') protocol => [Required] Protocol used for the Sub Account (Values from getProtocols) description => Sub Account Description (Example: 'VoIP Account') auth_type => [Required] Authorization Type Code (Values from getAuthTypes) password => Sub Account Password (For Password Authentication) ip => Sub Account IP (For IP Authentication) device_type => [Required] Device Type Code (Values from getDeviceTypes) callerid_number => Caller ID Override canada_routing => Route Code (Values from getRoutes) lock_international => [Required] Lock International Code (Values from getLockInternational) allow225 => When Enabled, calls placed to *225 will provide the Current Balance of the VoIP.ms Account. => When Disabled, calls placed to *225 will be rejected. international_route => [Required] Route Code (Values from getRoutes) music_on_hold => [Required] Music on Hold Code (Values from getMusicOnHold) language => Language for system messages, such as "Invalid Option" (Values from getLanguages) record_calls => Record Calls (Boolean: 1/0) allowed_codecs => [Required] List of Allowed Codecs (Values from getAllowedCodecs) Codecs separated by semicolon (Example: ulaw;g729;gsm) dtmf_mode => [Required] DTMF Mode Code (Values from getDTMFModes) nat => [Required] NAT Mode Code (Values from getNAT) sip_traffic => Encrypted SIP Traffic (Boolean: 1/0) max_expiry => Max Expiry between 60 and 3600 (Example: 3000) rtp_timeout => RTP Time Out between 1 and 3600 (Example: 60) rtp_hold_timeout => RTP Hold Time Out between 1 and 3600 (Example: 600) ip_restriction => List of IP Addresses, IP Addresses/Netmask, or Fully Qualified Domain Names to allow outgoing calls separated by commas (Example: 123.45.3.21,10.255.12.0/22,device.mydomain.com) enable_ip_restriction => Enable IP Restriction (Boolean: 1/0) pop_restriction => List of POP Servers to allow outgoing calls separated by commas (values from getServersInfo. Example: 10,23,45) enable_pop_restriction => Enable POP Restriction (Boolean: 1/0) send_bye => Send BYE on successful transfer (Boolean: 1/0) transcribe => Enable Call Transcription (Boolean: 1/0) transcription_locale => Call Transcription Locale (values from getLocales) transcription_email => Call Transcription Email internal_extension => Sub Account Internal Extension (Example: 1 -> Creates 101) internal_voicemail => Sub Account Internal Voicemail (Example: 101) internal_dialtime => Sub Account Internal Dialtime (Example: 60 -> seconds) reseller_client => Reseller Account ID (Example: 561115) </pre>

	reseller_package => Reseller Package (Example: 92364) reseller_nextbilling => Reseller Next Billing Date (Example: '2012-12-31') reseller_chargesetup => True if you want to charge Package Setup Fee after Save parking_lot => ID for a specific Call Parking (Example: 323) transcription_start_delay => Call Transcription Delay Seconds between 0 and 60, Increments of 5 (Example: 10 -> seconds) enable_internal_cnam => Enable Internal CallerID internal_cnam => Internal CallerID Name call_pickup_behavior => 1 Can Pick Up and Be Picked Up, 2 Can Pick Up (Can 't be Picked Up), 3 Can Be Picked Up (Can't Pick Up), 4 Disabled (Can't Pick Up nor Get Picked Up) dialing_mode => Allows you to dial outgoing calls using either the NANPA configuration or the E164 configuration. (Values: 0 = Use Main Account Setting, 1 = E164, 2 = NANPA) tfcarrier => This allows you to select the carrier to be used for outgoing calls to toll-free numbers. (Values: -1 = Use main account settings, 0 = Default server setting, 1 = US carrier, 2 = Canadian carrier)
Output	Array ([status] => success [id] => 99785 [account] => 100000_VoIP)
delSubAccount	
Parameters	id => ID for a specific Sub Account (Example: 99785)
Output	Array ([status] => success)
getAllowedCodecs	
Parameters	codec => Code for a specific Codec (Example: 'ulaw')
Output	Array ([status] => success [allowed_codecs] => Array ([0] => Array ([value] => ulaw [description] => G.711U))))
getAuthTypes	
Parameters	type => Code for a specific Authorization Type (Example: 2)
Output	Array ([status] => success [auth_types] => Array ([0] => Array ([value] => 2 [description] => Static IP Authentication))))
getDeviceTypes	
Parameters	device_type => Code for a specific Device Type (Example: 1)
Output	Array ([status] => success [device_types] => Array ([0] => Array ([value] => 1 [description] => Asterisk, IP PBX, Gateway or VoIP Switch))))
getDTFMModes	
Parameters	dtmf_mode => Code for a specific DTMF Mode (Example: 'inband')
Output	Array ([status] => success [dtmf_modes] => Array ([0] => Array ([value] => inband [description] => INBAND))))
getInvoice	
Parameters	from => Start Date to generate invoice. (Example: '2020-11-25') to => End Date to generate invoice. (Example: '2020-12-25') range => Predefined ranges of dates to generate invoice:

		1 = Last Month: 1st day of previous month to last day of previous month. 2 = Last 2 Months: 1st day of the previous 2 months to last day of previous month. 3 = Last 3 Months: 1st day of the previous 3 months to last day of the previous month. 4 = Current Month: 1st day of this month until today 5 = Last Week: Monday of last week to Sunday of last week. 6 = Current Week: Monday of this week until today.
	type =>	Type of invoice to be generated, possible values: 0 = US, 1 = CAN
Output	Array ([status] => success [pdf] => http://www.voip.ms/invoice.php?data=HASHCODE)	
getLockInternational		
Parameters	lock_international =>	Code for a specific Lock International Mode (Example: 1)
Output	Array ([status] => success [lock_international] => Array ([0] => Array ([value] => 1 [description] => International Calls Disabled)))	
getMusicOnHold		
Parameters	music_on_hold =>	Code for a specific Music on Hold (Example: 'jazz')
Output	Array ([status] => success [music_on_hold] => Array ([0] => Array ([value] => jazz [description] => Easy Listening [recordings] => 100010,100011 [total] => 2 [volume] => mp3 [sort] => random [custom] => 1)))	
setMusicOnHold		
Parameters	name => Music on Hold Name (Values from getMusicOnHold) description => [Required] Music on Hold Description volume => Music on Hold Quiet Volume (Boolean: 1/0) sort => Selected recordings sort mode (Example: "alpha", "random") recordings => [Required] Selected recordings separated by commas (Values from getRecordings, example: (1234,1235,1236))	
Output	Array ([status] => success)	
delMusicOnHold		
Parameters	music_on_hold =>	[Required] Music on Hold Name (Values from getMusicOnHold)
Output	Array ([status] => success)	
getNAT		
Parameters	nat =>	Code for a specific NAT Option (Example: 'route')
Output	Array ([status] => success [nat] => Array ([0] => Array ([value] => route [description] => Route)))	
getProtocols		
Parameters	protocol =>	Code for a specific Protocol (Example: 3)
Output	Array (	

	<pre>[status] => success [protocols] => Array ([0] => Array ([value] => 3 [description] => IAX2))</pre>
getRegistrationStatus	
Parameters	account => [Required] Specific Account (Example: '100001_VoIP')
Output	<pre>Array ([status] => success [registered] => yes [rerouted] => 1 [from_server_pop] => 65 [registrations] => Array ([0] => Array ([server_name] => Atlanta, GA [server_shortname] => Atlanta [server_hostname] => atlanta.voip.ms [server_ip] => 174.34.146.162 [server_country] => USA [server_pop] => 15 [register_ip] => 189.148.32.179 [register_port] => 59870 [register_next] => 2010-11-30 16:48:30 [register_protocol] => SIP [register_transport] => UDP)))</pre>
getReportEstimatedHoldTime	
Parameters	type => Code for a specific 'ReportEstimatedHoldTime' Type (Example: 'yes')
Output	<pre>Array ([status] => success [types] => Array ([0] => Array ([value] => yes [description] => Yes)))</pre>
getRoutes	
Parameters	route => Code for a specific Route (Example: 2)
Output	<pre>Array ([status] => success [routes] => Array ([0] => Array ([value] => 2 [description] => Premium)))</pre>
getSubAccounts	
Parameters	account => Parameter could have the following values: * Empty Value [Not Required] * Specific Sub Account (Example: '100000_VoIP') * Specific Reseller Client ID (Example: 561115)
Output	<pre>Array ([status] => success [accounts] => Array ([0] => Array ([id] => 99785 [account] => 100000_VoIP [username] => VoIP [description] => My VoIP Account [protocol] => 1 [auth_type] => 1 [password] => voip@house [ip] => [device_type] => 2 [callerid_number] => 5551234567 [canada_routing] => 2 [lock_international] => 0 [international_route] => 1 [music_on_hold] => jazz [language] => en [allowed_codecs] => ulaw;gsm [dtmf_mode] => auto [nat] => yes [sip_traffic] => 0 [max_expiry] => 3600)))</pre>

	<pre>[rtp_timeout] => 60 [rtp_hold_timeout] => 600 [ip_restriction] => 187.26.33.87/25,104.255.12.0/22 [enable_ip_restriction] => 1 [pop_restriction] => 82,83,84,71,73,74,97 [enable_pop_restriction] => 1 [send_bye] => 1 [record_calls] => 1 [internal_extension] => 0 [internal_voicemail] => 100 [internal_dialtime] => 30 [reseller_client] => 561115 [reseller_package] => 92364 [reseller_nextbilling] => 2012-12-31 [transcribe] => 1 [transcription_locale] => en-US [transcription_email] => test@example.com [parking_lot]=> 323 [enable_internal_cnam] => 1 [internal_cnam] => office [tfcARRIER] => 0),)</pre>
setSubAccount	
Parameters	<pre>id => [Required] Sub Account ID (Example: 10236) description => Sub Account Description (Example: 'VoIP Account') auth_type => [Required] Authorization Type Code (Values from getAuthTypes) password => Sub Account Password (For Password Authentication) ip => Sub Account IP (For IP Authentication) device_type => [Required] Device Type Code (Values from getDeviceTypes) callerid_number => Caller ID Override canada_routing => Route Code (Values from getRoutes) lock_international => [Required] Lock International Code (Values from getLockInternational) allow225 => When Enabled, calls placed to *225 will provide the Current Balance of the VoIP.ms Account. When Disabled, calls placed to *225 will be rejected. international_route => [Required] Route Code (Values from getRoutes) music_on_hold => [Required] Music on Hold Code (Values from getMusicOnHold) language => Language for system messages, such as "Invalid Option" (Values from getLanguages) record_calls => Record Calls (Boolean: 1/0) allowed_codecs => [Required] List of Allowed Codecs (Values from getAllowedCodecs) Codecs separated by semicolon (Example: ulaw;g729;gsm) dtmf_mode => [Required] DTMF Mode Code (Values from getDTMFModes) nat => [Required] NAT Mode Code (Values from getNAT) sip_traffic => Encrypted SIP Traffic (Boolean: 1/0) max_expiry => Max Expiry between 60 and 3600 (Example: 3000) rtp_timeout => RTP Time Out between 1 and 3600 (Example: 60) rtp_hold_timeout => RTP Hold Time Out between 1 and 3600 (Example: 600) ip_restriction => List of IP/Netmask to allow outgoing calls separated by commas (Example: 123.45.3.21,10.255.12.0/22,device.mydomain.com) enable_ip_restriction => Enable IP Restriction (Boolean: 1/0) pop_restriction => List of POP Servers to allow outgoing calls separated by commas (values from getServersInfo. Example: 10,23,45) enable_pop_restriction => Enable POP Restriction (Boolean: 1/0) send_bye => Send BYE on successful transfer (Boolean: 1/0) internal_extension => Sub Account Internal Extension (Example: 1 -> Creates 101) internal_voicemail => Sub Account Internal Voicemail (Example: 101) internal_dialtime => Sub Account Internal Dialtime (Example: 60 -> seconds) reseller_client => Reseller Account ID (Example: 561115) reseller_package => Reseller Package (Example: 92364) reseller_nextbilling => Reseller Next Billing Date (Example: '2012-12-31') reseller_chargesetup => True if you want to charge Package Setup Fee after Save parking_lot => ID for a specific Call Parking (Example: 323) transcription_start_delay => Call Transcription Delay Seconds between 0 and 60, Increments of 5 (Example: 10 -> seconds) enable_internal_cnam => Enable Internal CallerID internal_cnam => Internal CallerID Name call_pickup_behavior => 1 Can Pick Up and Be Picked Up, 2 Can Pick Up (Can 't be Picked Up), 3 Can Be Picked Up (Can't Pick Up), 4 Disabled (Can't Pick Up nor Get Picked Up) dialing_mode => Allows you to dial outgoing calls using either the NANPA configuration or the E164 configuration. (Values: 0 = Use Main Account Setting, 1 = E164, 2 = NANPA) tfcARRIER => This allows you to select the carrier to be used for outgoing calls to toll-free numbers. (Values: -1 = Use main account settings, 0 = Default server setting, 1 = US carrier, 2 = Canadian carrier)</pre>
Output	<pre>Array ([status] => success)</pre>

	<pre> [destination] => parker [delay] => 5 [blf_lamps] => 10))) </pre>
setCallParking	
Parameters	<pre> callparking => ID for a specific Call Parking (Example: 235 / Leave empty to create a new one) name => [Required] Name for the Call Parking timeout => [Required] The number of seconds a Call will stay parked before it is forwarded to the Failover Destination music => [Required] Music on Hold Code (Values from getMusicOnHold) failover => [Required] Final destination where the call will be forwarded if it isn't answered. (Values: callback, system:hangup, vm:mailbox) language => [Required] Language for the Call Parking (values from getLanguages) destination => [Required] The system will make an automatic call to this destination to announce the extension of the parked call. (Values: parker, main account or sub-accounts) delay => [Required] The number of seconds before the Announce Destination receives an automatic call from the system to announce the extension of the parked call blf_lamps => [Required] You can enable BLF for your Parking Lot and select the amount of BLF Lamps that you require. (Values: 0 to 10) </pre>
Output	<pre> Array ([status] => success [callparking] => 235) </pre>
delCallParking	
Parameters	<pre> callparking => [Required] ID for a specific Call Parking (Example: 323) </pre>
Output	<pre> Array ([status] => success) </pre>

Call Recording

The following table explains the additional parameters needed by the Call Recordings Functions and provides an example of their output when status is 'success'.

getCallRecordings	
Parameters	<pre> account => [Required] Filter Call Recordings by Account (Values from getCallAccounts) start => Number of records shown previously, used for pages length => Number of records to show date_from => [Required] Start Date for Filtering Call Recording (Example:'2018-11-01') date_to => [Required] End Date for Filtering Call Recording (Example:'2018-12-01') call_type => Call Type for Filtering Call Recording (Accepted values: 'all', 'incoming' and 'outgoing') </pre>
Output	<pre> Array ([status] => success [recordings] => Array ([0] => Array ([callrecording] => MTA5Nzk5LTUyOTk5MjQyNjE5NC01Mjk5OTM5OTA1MjAtSU4tMTU0NDEzMTMwNg [datetime] => 2018-12-06 16:21:46 [destination] => 999999999 [type] => Incoming [subaccount] => 109799 [duration] => 00:26) [1] => Array ([callrecording] => MTA5Nzk5LTUyOTk5MjQyNjE5NC01Mjk5OTM5OTA1MjAtSU4tMTU0NDEzMTMwNg [datetime] => 2018-12-06 16:21:46 [destination] => 555555555 [type] => Outgoing [subaccount] => 109799 [duration] => 00:26))) </pre>
getCallRecording	
Parameters	<pre> account => [Required] Main Account or Sub Account related to the call recording (Values from getCallRecordings) callrecording => [Required] Call Recording (Values from getCallRecordings) </pre>
Output	<pre> Array ([status] => success [callrecording] => MTA5Nzk5LTUyOTk5MjQyNjE5NC01Mjk5OTM5OTA1MjAtSU4tMTU0NDEzMTMwNg [datetime] => 2018-12-06 16:21:46 [destination] => 999999999 [type] => Incoming [subaccount] => 109799 [duration] => 00:26 [base64file] => "\/M4xAAAAAAAAAAAAAEluZm8AAAAAPAAABZwA BL8AAAgUHCgwPEhQXGRwfISQmKSwuMTM2ODs + QENFSEpNT1JUUV1lcXmFkZm1rbnBzdnh7fYCDhYiKjZCS1ZeanJ+hpKapg66ws7W4u73AwsXIys3P0tTX2tzf4eTn6 e </pre>

)

Parameters	account => [Required] Filter Call Recordings by Account (Values from getCallAccounts) email => [Required] Email to send call recording callrecording => [Required] Call Recording (Values from getCallRecordings)
Output	Array ([status] => success [msg] => Email sent to mail@sonpany.com)

Parameters	account => [Required] Filter Call Recordings by Account (Values from getCallAccounts) callrecording => [Required] Call Recording (Values from getCallRecordings)
Output	Array ([status] => success)

The following table explains the additional parameters needed by the CDR Functions and provides an example of their output when status is 'success'

Parameters	client => ID for a specific Reseller Client (Example: 561115)
Output	<pre> Array ([status] => success [accounts] => Array ([0] => Array ([value] => all [description] => All Accounts) [1] => Array ([value] => 100000_VoIP [description] => 100000_VoIP))) </pre>

Parameters	No Parameter
Output	<pre> Array ([status] => success [call_billing] => Array ([0] => Array ([value] => all [description] => All Calls) [1] => Array ([value] => free [description] => Free Calls) [2] => Array ([value] => billed [description] => Billed Calls))) </pre>

Parameters	client => ID for a specific Reseller Client (Example: 561115)
Output	Array

	<pre> ([status] => success [call_types] => Array ([0] => Array ([value] => all [description] => All Calls) [1] => Array ([value] => outgoing [description] => Outgoing Calls) [2] => Array ([value] => incoming [description] => Incoming Calls) [3] => Array ([value] => 5551234567 [description] => Calls to 5551234567))) </pre>
--	--

getCDR

Parameters	<pre> date_from => [Required] Start Date for Filtering CDR (Example: '2010-11-30') date_to => [Required] End Date for Filtering CDR (Example: '2010-11-30') answered => Include Answered Calls to CDR (Boolean: 1/0) noanswer => Include NoAnswered calls to CDR (Boolean: 1/0) busy => Include Busy Calls to CDR (Boolean: 1/0) failed => Include Failed Calls to CDR (Boolean: 1/0) timezone => [Required] Adjust time of calls according to Timezome (Numeric: -12 to 13) calltype => Filters CDR by Call Type (Values from getCallTypes) callbilling => Filter CDR by Call Billing (Values from getCallBilling) account => Filter CDR by Account (Values from getCallAccounts) </pre>
Output	<pre> Array ([status] => success [cdr] => Array ([0] => Array ([date] => 2010-11-30 10:40:16 [callerid] => "John Doe" <5551112222> [destination] => 5551234567 [description] => Inbound - Local Caller [account] => 100000_VoIP [disposition] => ANSWERED [duration] => 00:00:11 [seconds] => 11 [rate] => 0.00000000 [total] => 0.00000000 [uniqueid] => 128238059 [destination_type] => IN:USA [call_logs] => Call logs for UniqueID: 999999 Routing to sub-account: 100000_VoIP Status is 'Channel not available' Failover due to 'Unreachable' status Routing to Ring group: test_ringgroup Hangup: Channel is unavailable) [1] => Array ([date] => 2010-09-30 10:36:32 [callerid] => "Jane Doe" <5551234567> [destination] => 5551112222 [description] => Outbound - Customer [account] => 100000_VoIP [disposition] => ANSWERED [duration] => 00:00:55 [seconds] => 55 [rate] => 0.00000000 [total] => 0.00000000 [uniqueid] => 128236206 [destination_type] => IN:USA [call_logs] => Call logs for UniqueID: 999999 Routing to Call hunting: test_ch Status is 'Channel not available' Failover due to 'Unreachable' status Routing to Disa: test_disa Hangup: Channel is unavailable))) </pre>

getRates

Parameters	<pre> package => [Required] ID for a specific Package (Example: 92364) query => [Required] Query for searching rates (Example: 'Canada') </pre>
Output	<pre> Array ([status] => success [rates] => Array ([0] => Array ([destination] => Canada - 418 Quebec City [prefix] => 1418 [client_increment] => 60 [client_rate] => 0.015 [real_increment] => 6 [real_rate] => 0.0052) [1] => Array (</pre>

		<pre> [destination] => Canada - 581 Quebec City [prefix] => 1581 [client_increment] => 60 [client_rate] => 0.015 [real_increment] => 6 [real_rate] => 0.0052))) </pre>
getTerminationRates		
Parameters	<pre> route => [Required] Route Code (Values from getRoutes)(Example: '2') query => [Required] Query for searching rates (Example: 'Canada') </pre>	
Output		<pre> Array ([status] => success [route] => Array ([value] => 2 [description] => Premium) [rates] => Array ([0] => Array ([destination] => Canada - 204 Manitoba [prefix] => 1204 [increment] => 6 [rate] => 0.009) [1] => Array ([destination] => Canada - 418 Quebec City [prefix] => 1418 [increment] => 6 [rate] => 0.009))) </pre>
getResellerCDR		
Parameters	<pre> date_from => [Required] Start Date for Filtering CDR (Example: '2010-11-30') date_to => [Required] End Date for Filtering CDR (Example: '2010-11-30') client => [Required] ID for a specific Reseller Client (Example: 561115) answered => Include Answered Calls to CDR (Boolean: 1/0) noanswer => Include NoAnswered calls to CDR (Boolean: 1/0) busy => Include Busy Calls to CDR (Boolean: 1/0) failed => Include Failed Calls to CDR (Boolean: 1/0) timezone => [Required] Adjust time of calls according to Timezone (Numeric: -12 to 13) calltype => Filters CDR by Call Type (Values from getCallTypes) callbilling => Filter CDR by Call Billing (Values from getCallBilling) account => Filter CDR by Account (Values from getCallAccounts) </pre>	
Output		<pre> Array ([status] => success [cdr] => Array ([0] => Array ([date] => 2010-11-30 10:40:16 [callerid] => "John Doe" <5551112222> [destination] => 5551234567 [description] => Inbound - Local Caller [account] => 100000 VoIP [disposition] => ANSWERED [duration] => 00:00:11 [seconds] => 11 [total] => 0.015 [uniqueid] => 128238059 [destination_type] => IN:USA [call_logs] => Call logs for UniqueID: 999999 Routing to sub-account: 100000 VoIP Status is 'Channel not available' Failover due to 'Unreachable' status Routing to Ring group: test_ringgroup Hangup: Channel is unavailable) [1] => Array ([date] => 2010-09-30 10:36:32 [callerid] => "Jane Doe" <5551234567> [destination] => 5551112222 [description] => Outbound - Customer [account] => 100000 VoIP [disposition] => ANSWERED [duration] => 00:00:55 [seconds] => 55 [total] => 0.015 [uniqueid] => 128236206 [destination_type] => IN:USA [call_logs] => Call logs for UniqueID: 999999 Routing to Call hunting: test_ch Status is 'Channel not available' Failover due to 'Unreachable' status Routing to Disa: test_disa Hangup: Channel is unavailable))) </pre>
getResellerSMS		
Parameters	<pre> sms => ID for a specific SMS (Example: 5853) from => Start Date for Filtering SMSs (Example: '2014-03-30') - Default value: Today </pre>	

	<pre> to => End Date for Filtering SMSs (Example: '2014-03-30') - Default value: Todayclient => [Required] ID for a specific Reseller Client (Example: 561115) type => Filter SMSs by Type (Boolean: 1 = received / 0 = sent) did => DID number for Filtering SMSs (Example: 5551234567) contact => Contact number for Filtering SMSs (Example: 5551234567) limit => Number of records to be displayed (Example: 20) - Default value: 50 timezone=> Adjust time of SMSs according to Timezone (Numeric: -12 to 13) all_messages=> Filter to recive all SMSs and MMSs, 1 recive all SMS and MMS, 0 if only need SMS, important: the sms ID must be 0 </pre>
Output	<pre> Array ([status] => success [sms] => Array ([0] => Array ([id] => 111120 [date] => 2014-03-30 10:24:16 [type] => 0 [did] => 8574884828 [contact] => 8577884821 [message] => hello+john)))) </pre>

getResellerMMS

Parameters	<pre> sms => ID for a specific MMS (Example: 5853) from => Start Date for Filtering SMSs (Example: '2014-03-30') - Default value: Today to => End Date for Filtering MMSs (Example: '2014-03-30') - Default value: Todayclient => [Required] ID for a specific Reseller Client (Example: 561115) type => Filter SMSs by Type (Boolean: 1 = received / 0 = sent) did => DID number for Filtering MMSs (Example: 5551234567) contact => Contact number for Filtering MMSs (Example: 5551234567) limit => Number of records to be displayed (Example: 20) - Default value: 50 timezone=> Adjust time of MMSs according to Timezone (Numeric: -12 to 13) all_messages=> Filter to recive all SMSs and MMSs, 1 recive all SMS and MMS, 0 if only need SMS, important: the sms ID must be 0 </pre>
Output	<pre> Array ([status] => success [sms] => Array ([0] => Array ([id] => 111120 [date] => 2014-03-30 10:24:16 [type] => 0 [did] => 8574884828 [contact] => 8577884821 [message] => hello+john)))) </pre>

CLIENT FUNCTIONS

The following table explains the additional parameters needed by the Client Functions and provides an example of their output when status is 'success'.

addCharge	
Parameters	<pre> client => [Required] ID for a specific Reseller Client (Example: 561115) charge => [Required] Amount of money that will be Debited from the customer (Example: 4.99) description => Charge Description test => Set to true if testing how adding charges works </pre>
Output	<pre> Array ([status] => success) </pre>
addPayment	
Parameters	<pre> client => [Required] ID for a specific Reseller Client (Example: 561115) payment => [Required] Amount of money that will be Credited to the customer (Example: 4.99) description => Payment Description test => Set to true if testing how adding payments works </pre>
Output	<pre> Array ([status] => success) </pre>

assignDIDvPRI		
Parameters	did => [Required] DID Number to be assign into our Vpri (Example: 561115) vpri => [Required] Id for specific Vpri	
Output	Array ([status] => success [vpri] => 5 [DIDAdded] => 5551234567 [monthly] => 2.00)	
getBalanceManagement		
Parameters	balance_management => Code for a specific Balance Management Setting (Example: 1)	
Output	Array ([status] => success [balance_management] => Array ([0] => Array ([value] => 1 [description] => Hard - suspend the client)))	
getCharges		
Parameters	client => [Required] ID for a specific Reseller Client (Example: 561115)	
Output	Array ([status] => success [charges] => Array ([0] => Array ([id] => 475568 [date] => 2010-10-29 [amount] => 2.00 [description] => DID5551234567) [1] => Array ([id] => 475567 [date] => 2010-10-29 [amount] => 1.00 [description] => SETUP5551234567) [3] => Array ([id] => 467988 [date] => 2010-09-15 [amount] => 3.00 [description] => SETUPPACKAGE:8378)))	
getClientPackages		
Parameters	client => [Required] ID for a specific Reseller Client (Example: 561115)	
Output	Array ([status] => success [packages] => Array ([0] => Array ([value] => 8378 [description] => VoIP)))	
getClients		
Parameters	client => Parameter could have the following values: * Empty Value [Not Required] * Specific Reseller Client ID (Example: 561115) * Specific Reseller Client e-mail (Example: 'john.doe@mydomain.com')	
Output	Array ([status] => success [clients] => Array ([0] => Array ([client] => 561115 [email] => john.doe@my-domain.com [password] => johnspassword [company] => Consulting [firstname] => John [lastname] => Doe [address] => 5491 190TH [city] => OPA LOCKA [state] => FLORIDA))	

	<pre> [country] => UNITED STATES [zip] => 33055 [phone_number] => 5551112222 [balance_management] => 1))) </pre>
getClientThreshold	
Parameters	client => [Required] ID for a specific Reseller Client (Example: 561115)
Output	<pre> Array ([status] => success [threshold_information] => Array ([threshold] => 130 [email] => john.doe@my-domain.com)) </pre>
getDeposits	
Parameters	client => [Required] ID for a specific Reseller Client (Example: 561115)
Output	<pre> Array ([status] => success [deposits] => Array ([0] => Array ([id] => 808639 [date] => 2010-10-15 [amount] => 4.00 [description] => Credit : 2 Setup Charges) [1] => Array ([id] => 808384 [date] => 2010-11-30 [amount] => 10.00 [description] => Want To Keep Positive Balance))) </pre>
getPackages	
Parameters	package => Code for a specific Package (Example: 8378)
Output	<pre> Array ([status] => success [packages] => Array ([0] => Array ([package] => 8378 [name] => VoIP [markup_fixed] => 0.01 [markup_percentage] => 50 [pulse] => 60 [international_route] => 1 [canada_route] => 1 [monthly_fee] => 2.00 [setup_fee] => 1.00 [free_minutes] => 0))) </pre>
getResellerBalance	
Parameters	client => [Required] ID for a specific Reseller Client (Example: 561115)
Output	<pre> Array ([status] => success [balance] => Array ([current_balance] => 8.0588 [spent_total] => 1.9012 [calls_total] => 56 [time_total] => 51:35 [spent_today] => 0.045 [calls_today] => 2 [time_today] => 2:50)) </pre>
setClient	
Parameters	<pre> client => [Required] ID for a specific Reseller Client (Example: 561115) email => [Required] Client's e-mail password => [Required] Client's Password company => Client's Company firstname => [Required] Client's Firstname lastname => [Required] Client's Lastname address => Client's Address city => Client's City state => Client's State country => Client's Country (Values from getCountries) zip => Client's Zip Code phone_number => [Required] Client's Phone Number </pre>

	balance_management => Balance Management for Client (Values from getBalanceManagement)
Output	Array ([status] => success)
setClientThreshold	
Parameters	client => [Required] ID for a specific Reseller Client (Example: 561115) threshold => [Required] Threshold amount between 1 and 250 (Example: 10) email => Client's e-mail for balance threshold notification
Output	Array ([status] => success)
setConference	
Parameters	conference => [Required] ID for a specific Conference (Example: 5356) name => [Required] Conference name description => [Required] Conference description members => Conference Members max_members => [Required] Members Max Value sound_join => The recording played when a user joins, typically some kind of beep sound (Values from getRecordings) sound_leave => The recording played when a user leaves, typically some kind of beep sound (Values from getRecordings) sound_has_joined => The recording played as a user intro (Values from getRecordings) sound_has_left => The recording played as a user leaves the conference (Values from getRecordings) sound_kicked => The recording played to a user who has been kicked from the conference (Values from getRecordings) sound_muted => The recording played to a user when the mute option is toggled on (Values from getRecordings) sound_unmuted => The recording played to a user when the mute option is toggled off (Values from getRecordings) sound_only_person => The recording played when a user is the only person in the conference (Values from getRecordings) sound_only_one => The recording played to a user when there is only one other person in the conference. (Values from getRecordings) sound_there_are => The recording played when announcing how many users there are in a conference. (Values from getRecordings) sound_other_in_party => The recording used in conjunction with the There are option, used like There are (number of participants) Other in party (Values from getRecordings) sound_place_into_conference => The recording played when a user is placed into a conference that cannot start until a marked user enters (Values from getRecordings) sound_get_pin => The recording played when prompting for a conference PIN (Values from getRecordings) sound_invalid_pin => The recording played when an invalid PIN is entered too many (3) times (Values from getRecordings) sound_locked => The recording played to a user trying to join a locked conference (Values from getRecordings) sound_locked_now => The recording played to an Admin-level user after toggling the conference to locked mode (Values from getRecordings) sound_unlocked_now => The recording played to an Admin-level user after toggling the conference to unlocked mode (Values from getRecordings) sound_error_menu => The recording played when there is an error on the menu. (Values from getRecordings) sound_participants_muted => The recording played when all non-admin participants are muted. (Values from getRecordings) sound_participants_unmuted => The recording played when all non-admin participants are unmuted. (Values from getRecordings) language => Conference Language (Values from getLanguages)
Output	Array ([status] => success [conference] => 1234)
setConferenceMember	
Parameters	conference => [Required] ID for a specific Conference (Example: 5356) member => [Required] ID for a specific Member profile (Example: 5356) name => [Required] Member name. description => Member description. pin => Assigned PIN. announce_join_leave => Sets if the conference recording when a member joins or leaves will be played (yes/no). admin => Sets if the member is an admin or not (yes/no). start_muted => Sets if the member should start out muted after entering the conference (yes/no). announce_user_count => Sets if the number of members in the conference should be announced to the caller as he joins (yes/no). announce_only_user => Sets if the "only user" announcement should be played when a caller enters an empty conference (yes/no). moh_when_empty => Sets whether music on hold (MOH) should be played when only one person is in the conference (Values from getMusicOnHold). quiet => When set to "yes", enter/leave prompts and user introductions are not played (yes/no). announcement => If set, this recording will be heard only by the user as he joins the conference (Values from getRecordings). drop_silence => The system will drop what is detected as silence from entering into the conference (yes/no). talking_threshold => The time, in milliseconds, that a users needs to be sending sound or voice before the system can consider them to be talking (allowed values are 100, 120, 140, 160, 180, 200, 220, 240 or 250). silence_threshold => The time, in milliseconds, that silence needs to be present in the user's sound stream before the system can consider it to be in fact silent and close the audio (allowed values are 2000, 2100, 2200, 2300, 2400, 2500, 2600, 2700, 2800, 2900 or 3000). talk_detection => If set to YES, the conference dashboard will display a notification when a participant starts and stops talking (yes/no). jitter_buffer => When set to YES, the system will place a jitter buffer on the caller's audio stream before any audio mixing is performed (yes/no).
Output	Array ([status] => success [member] => 1234)
setSequences	
Parameters	sequence => [Required] ID for a specific Sequence (Example: 5356) name => [Required] Sequence name

	<pre>steps => [Required] Sequence steps (Example: "[{ "type": "rec", "id": "1234" }, { "type": "dtmf", "codes": "1234" }, { "type": "sms", "from": "1234567890", "to": "0987654321", "msg": "This is a test." }]")</pre>
	<pre>client => [Optional] ID for a specific Reseller Client (Example: 561115)</pre>
Output	<pre>Array ([status] => success [sequence] => 1234)</pre>
signupClient	
Parameters	<pre>firstname => [Required] Client's Firstname lastname => [Required] Client's Lastname company => [Required] Client's Company address => [Required] Client's Address city => [Required] Client's City state => [Required] Client's State country => [Required] Client's Country (Values from getCountries) zip => [Required] Client's Zip Code phone_number => [Required] Client's Phone Number email => [Required] Client's e-mail confirm_email => [Required] Client's Confirmation e-mail password => [Required] Client's Password confirm_password => [Required] Client's Confirmation Password activate => [Required] Activates Client (Boolean: 1/0) balance_management => [Required] Balance Management for Client (Values from getBalanceManagement)</pre>
Output	<pre>Array ([status] => success [client] => 1234)</pre>

DID FUNCTIONS

The following table explains the additional parameters needed by the DID Functions and provides an example of their output when status is 'success'.

backOrderDIDUSA	
Parameters	<pre>quantity => [Required] Number of DIDs to be Ordered (Example: 3) state => [Required] USA State (values from getStates) ratecenter => [Required] USA Ratecenter (Values from getRateCentersUSA) routing => [Required] Main Routing for the DID failover_busy => [Required] Busy Routing for the DID failover_unreachable => [Required] Unreachable Routing for the DID failover_noanswer => [Required] NoAnswer Routing for the DID voicemail => [Required] Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => [Required] Caller ID Prefix for the DID note => [Required] Note for the DID billing_type => [Required] Billing type for the DID (1 = Per Minute, 2 = Flat) test => [Required] Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made</pre> routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: <pre>account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action</pre> Examples: <pre>'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo'</pre>

Output	Array ([status] => success)
backOrderDIDCAN	
Parameters	quantity => [Required] Number of DIDs to be Ordered (Example: 3) province => [Required] Canadian Province (values from getProvinces) ratecenter => [Required] USA Ratecenter (Values from getRateCentersUSA) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID note => Note for the DID billing_type => [Required] Billing type for the DID (1 = Per Minute, 2 = Flat) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: <pre> 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo' </pre>
Output	Array ([status] => success)
cancelDID	
Parameters	did => [Required] DID to be canceled and deleted (Example: 5551234567) cancelcomment => Comment for DID cancellation portout => Set to true if the DID is being ported out test => Set to true if testing how cancellation works - Cancellation can not be undone - When testing, no changes are made
Output	Array ([status] => success)
connectDID	
Parameters	did => [Required] DID to be Connected to Reseler Sub Account (Example: 5551234567) account => [Required] Reseller Sub Account (Example: '100001_VoIP') monthly => [Required] Montly Fee for Reseller Client (Example: 3.50) setup => [Required] Setup Fee for Reseller Client (Example: 1.99) minute => [Required] Minute Rate for Reseller Client (Example: 0.03) next_billing => Next billing date (Example: '2014-03-30') dont_charge_setup => If set to true, the setup value will not be charged after Connect dont_charge_monthly => If set to true, the monthly value will not be charged after Connect
Output	Array ([status] => success)
delCallback	
Parameters	callback => [Required] ID for a specific Callback (Example: 19183)
Output	Array ([status] => success)
delCallerIDFiltering	
Parameters	filtering => [Required] ID for a specific CallerID Filtering (Example: 19183)
Output	Array ([status] => success)

	) [status] => success
delCallHunting	
Parameters	callhunting => [Required] ID for a specific Call Hunting (Example: 323)
Output	Array ([status] => success)
delClient	
Parameters	client => [Required] ID for a specific Reseller Client (Example: 1998)
Output	Array ([status] => success)
delConference	
Parameters	conference => [Required] ID for a specific Conference (Example: 737)
Output	Array ([status] => success)
delConferenceMember	
Parameters	member => [Required] ID for a specific Member Profile (Example: 737)
Output	Array ([status] => success)
delSequences	
Parameters	sequence => [Required] ID for a specific Sequence (Example: 5356) client => [Optional] ID for a specific Reseller Client (Example: 561115)
Output	Array ([status] => success)
delDISA	
Parameters	disa => [Required] ID for a specific DISA (Example: 19183)
Output	Array ([status] => success)
deleteSMS	
Parameters	id => [Required] ID for a specific SMS (Example: 1918)
Output	Array ([status] => success)
deleteMMS	
Parameters	id => [Required] ID for a specific MMS (Example: 1918)
Output	Array ([status] => success)
delForwarding	
Parameters	forwarding => ID for a specific Forwarding (Example: 19183)
Output	Array ([status] => success)
delIVR	
Parameters	ivr => [Required] ID for a specific IVR (Example: 19183)

Output	Array ([status] => success)
delPhonebook	
Parameters	phonebook => [Required] ID for a specific Phonebook (Example: 19183)
Output	Array ([status] => success)
delPhonebookGroup	
Parameters	group => [Required] ID for a specific Phonebook group
Output	Array ([status] => success)
delQueue	
Parameters	queue => [Required] ID for a specific Queue (Example: 13183)
Output	Array ([status] => success)
delRecording	
Parameters	recording => [Required] ID for a specific Recording (Example: 19183)
Output	Array ([status] => success)
delRingGroup	
Parameters	ringgroup => [Required] ID for a specific Ring Group (Example: 19183)
Output	Array ([status] => success)
delSIPURI	
Parameters	sipuri => [Required] ID for a specific SIP URI (Example: 19183)
Output	Array ([status] => success)
delStaticMember	
Parameters	member => [Required] ID for a specific Member Queue (Example: 1918) queue => [Required] ID for a specific Queue (Example: 27183)
Output	Array ([status] => success)
delTimeCondition	
Parameters	timecondition => [Required] ID for a specific Time Condition (Example: 19183)
Output	Array ([status] => success)
getCallbacks	
Parameters	callback => ID for a specific Callback (Example: 2359)
Output	Array ([status] => success [callbacks] => Array ([0] => Array ([callback] => 2359 [description] => John Smith [number] => 5551234567 [delay_before] => 5 [response_timeout] => 10 [digit_timeout] => 5))))

	<pre> [perminute_monthly] => 0.99 [perminute_minute] => 0.0100 [perminute_setup] => 0.50 [flat_monthly] => 4.95 [flat_minute] => 0.00 [flat_setup] => 1.00 [sms] => 1)) [2] => Array ([did] => 2264836086 [ratecenter] => Stoney Point [province] => ON [province_description] => ONTARIO [perminute_monthly] => 0.99 [perminute_minute] => 0.0100 [perminute_setup] => 0.50 [flat_monthly] => 4.95 [flat_minute] => 0.00 [flat_setup] => 1.00 [sms] => 1))) </pre>
--	--

getDIDsInfo

Parameters	client => Parameter could have the following values: * Empty Value [Not Required] * Specific Reseller Client ID (Example: 561115) * Specific Sub Account (Example: '100001 VoIP') did => DID from Client or Sub Account (Example: '5551234567')
Output	<pre> Array ([status] => success [dids] => Array ([0] => Array ([did] => 5551234567 [description] => MIAMI, FL [routing] => account:100001_VoIP [failover_busy] => none: [failover_unreachable] => fwd:1521 [failover_noanswer] => none: [voicemail] => 1001 [pop] => 16 [dialtime] => 60 [cnam] => 1 [e911] => 0 [callerid_prefix] => MIA [555] [note] => MIA [555] [port_out_pin] => 1234 [billing_type] => 2 [next_billing] => 2014-08-30 [order_date] => 2014-07-30 15:09:04 [reseller_account] => 100001 VoIPReseller [reseller_next_billing] => 2014-08-30 [reseller_monthly] => 0.00000000 [reseller_minute] => 0.01000000 [reseller_setup] => 0.00000000 [sms_available] => 1 [sms_enabled] => 1 [sms_email] => john.doe@example.com [sms_email_enabled] => 1 [sms_forward] => 2264836086 [sms_forward_enabled] => 1 [sms_url_callback] => http://myurl.com [sms_url_callback_enabled] => 1 [sms_url_callback_retry] => 1 [smpp_enabled] => 1 [smpp_url] => http://myurl.com [smpp_user] => [smpp_pass] => [transcribe] => 1 [transcription_locale] => en-US [transcription_email] => test@example.com)))) </pre>

getDIDsInternationalGeographic

Parameters	country_id => [Required] ID for a specific Country (Values from getDIDCountries)
Output	<pre> Array ([status] => success [locations] => Array ([0] => Array ([location_id] => a65 [location_name] => HELSINKI [country] => FINLAND [area_code] => 9 [stock] => 2 [monthly] => 5.00 [setup] => 0.00 [minute] => 0.00 [monthly_per_minute] => 7.00, [setup_per_minute] => 5.50, [channels] => 2) [1] => Array ([location_id] => b9468 [location_name] => KESKI-SUOMI [country] => FINLAND [area_code] => 14 [stock] => 93 [monthly] => 5.00 [setup] => 0.00)))) </pre>

	<pre>[minute] => 0.00 [monthly_per_minute] => 7.00, [setup_per_minute] => 5.50, [channels] => 2)))</pre>
getDIDsInternationalNational	
Parameters	country_id => [Required] ID for a specific Country (Values from getDIDCountries)
Output	<pre>Array ([status] => success [locations] => Array ([0] => Array ([location_id] => a5644 [location_name] => NATIONAL [country] => SPAIN [area_code] => 901 [stock] => 4 [monthly] => 11.50 [setup] => 11 [minute] => 0 [monthly_per_minute] => 6.00, [setup_per_minute] => 5.50,) [1] => Array ([location_id] => b6847 [location_name] => NATIONAL [country] => SPAIN [area_code] => 902 [stock] => 135 [monthly] => 11.50 [setup] => 11 [minute] => 0 [monthly_per_minute] => 7.00, [setup_per_minute] => 5.50,))))</pre>
getDIDsInternationalTollFree	
Parameters	country_id => [Required] ID for a specific Country (Values from getDIDCountries)
Output	<pre>Array ([status] => success [locations] => Array ([0] => Array ([location_id] => a6422 [location_name] => TOLLFREE [country] => ARGENTINA [area_code] => 800 [stock] => 19 [monthly] => 37.50 [setup] => 72.50 [minute] => 0.20) [1] => Array ([location_id] => b6424 [location_name] => TOLLFREE [country] => ARGENTINA [area_code] => 822 [stock] => 15 [monthly] => 37.50 [setup] => 72.50 [minute] => 0.20))))</pre>
getDIDsUSA	
Parameters	state => [Required] United States State (Values from getStates) ratecenter => United States Ratecenter (Values from getRateCentersUSA)
Output	<pre>Array ([status] => success [dids] => Array ([0] => Array ([did] => 7862285619 [ratecenter] => MIAMI [state] => FL [state_description] => FLORIDA [perminute_monthly] => 0.99 [perminute_minute] => 0.01 [perminute_setup] => 0.50 [flat_monthly] => 4.95 [flat_minute] => 0.00 [flat_setup] => 1.00 [has_port_out_pin] => 1 [sms] => 1) [1] => Array ([did] => 7862285871 [ratecenter] => MIAMI [state] => FL [state_description] => FLORIDA</pre>

	<pre> [perminute_monthly] => 0.99 [perminute_minute] => 0.01 [perminute_setup] => 0.50 [flat_monthly] => 4.95 [flat_minute] => 0.00 [flat_setup] => 1.00 [has_port_out_pin] => 0 [sms] => 1))) </pre>
getDIDvPRI	
Parameters	vpri => [Required] Id for specific Vpri
Output	<pre> Array ([status] => success [status] => vpri [dids] => Array ([0] => 7862285619 [1] => 7862285871)) </pre>
getDISAs	
Parameters	disa => ID for a specific DISA (Example: 2114)
Output	<pre> Array ([status] => success [disa] => Array ([0] => Array ([disa] => 2114 [name] => Jonhs DISA [pin] => 1234 [digit_timeout] => 10 [callerid_override] => 5552223333))) </pre>
getForwardings	
Parameters	forwarding => ID for a specific Forwarding (Example: 18635)
Output	<pre> Array ([status] => success [forwardings] => Array ([0] => Array ([forwarding] => 18635 [phone_number] => 5553334444 [callerid_override] => 5551234567 [description] => RESELLER 5553334444 [dtmf_digits] => 123 [pause] => 1.5 [diversion_header] => 1))) </pre>
getInternationalTypes	
Parameters	type => Code for a specific International Type (Example: 'NATIONAL')
Output	<pre> Array ([status] => success [types] => Array ([0] => Array ([value] => NATIONAL [description] => National Numbers (Non-Geographical)))) </pre>
getIVRs	
Parameters	ivr => ID for a specific IVR (Example: 4636)
Output	<pre> Array ([status] => success [ivrs] => Array ([0] => Array ([ivr] => 4636 [name] => Johns IVR [recording] => 7696 [timeout] => 5 [language] => fr [voicemailsetup] => 1 [choices] => 1=sip:5096;2=fwd:20222))) </pre>

getJoinWhenEmptyTypes	
Parameters	type => Code for a specific 'JoinWhenEmpty' Type (Example: 'yes')
Output	<pre>Array ([status] => success [types] => Array ([0] => Array ([value] => yes [description] => Yes)))</pre>
getPhonebook	
Parameters	phonebook => ID for a specific Phonebook entry (Example: 32207) name => Name to be searched in database (Example: 'jane') group => ID for a specific Phonebook group group_name => Group Name
Output	<pre>Array ([status] => success [phonebooks] => Array ([0] => Array ([phonebook] => 32207 [speed_dial] => *7501 [name] => Jane Smith [number] => 5553334444 [callerid] => 5552223333 [note] => Janes Mobile [group] => 3220 [group_name] => Work)))</pre>
getPhonebookGroups	
Parameters	group => ID for a specific Phonebook group name => Group Name
Output	<pre>Array ([status] => success [phonebooks] => Array ([0] => Array ([phonebook_group] => 32207 [name] => Work [members] => 123456;654321)))</pre>
getPortability	
Parameters	did => [Required] DID Number to be ported into our network (Example: 5552341234)
Output	<pre>Array ([status] => success [portable] => yes [sms] => yes [plans] => Array ([0] => Array ([title] => Per Minute Plan [pricePerMonth] => 0.99 [pricePerMin] => 0.01) [1] => Array ([title] => Monthly Flat Rate Plan [pricePerMonth] => 4.95 [pricePerMin] => 0.00)))</pre>
getRecordings	
Parameters	recording => ID for a specific Recording (Example: 7567)
Output	<pre>Array ([status] => success [recordings] => Array ([0] => Array ([value] => 7567 [description] => IVR Recording)))</pre>

	)
getRecordingFile	
Parameters	recording => [Required] ID for a specific Recording (Example: 7567)
Output	<pre> Array ([status] => success [recordings] => Array ([0] => Array ([value] => 7567 [data] => Uk1GRqTEAQBXQVZFZm10IBAAAAABAQ8BAIA+AAACABAAZGF0YyDEA...)))) </pre>
getRingGroups	
Parameters	ring_group => ID for a specific Ring Group (Example: 4768)
Output	<pre> Array ([status] => success [ring_groups] => Array ([0] => Array ([ring_group] => 4768 [name] => Account & Mobile [caller_announcement] => 7567 [music_on_hold] => jazz [language] => en [members] => account:100001;fwd:16006 [voicemail] => 101)))) </pre>
getRingStrategies	
Parameters	strategy => ID for a specific Ring Strategy (Example: 'rrmemory')
Output	<pre> Array ([status] => success [strategies] => Array ([0] => Array ([value] => rrmemory [description] => Round Robin Memory)))) </pre>
getSIPURIs	
Parameters	sipuri => ID for a specific SIP URI (Example: 6199)
Output	<pre> Array ([status] => success [sipuris] => Array ([0] => Array ([sipuri] => 6199 [uri] => 5552223333@sip.voip.ms [description] => Jonh Smith [callerid_override] => 529999999 [callerid_e164] => 1)))) </pre>
getSMS	
Parameters	sms => ID for a specific SMS (Example: 5853) from => Start Date for Filtering SMSs (Example: '2014-03-30') - Default value: Today to => End Date for Filtering SMSs (Example: '2014-03-30') - Default value: Today type => Filter SMSs by Type (Boolean: 1 = received / 0 = sent) did => DID number for Filtering SMSs (Example: 5551234567) contact => Contact number for Filtering SMSs (Example: 5551234567) limit => Number of records to be displayed (Example: 20) - Default value: 50 timezone=> Adjust time of SMSs according to Timezone (Numeric: -12 to 13) all_messages=> Filter to recive all SMSs and MMSs, 1 recive all SMS and MMS, 0 if only need SMS, important: the sms ID must be 0
Output	<pre> Array ([status] => success [sms] => Array ([0] => Array ([id] => 111120 [date] => 2014-03-30 10:24:16)))) </pre>

	<pre> [type] => 0 [did] => 8574884828 [contact] => 8577884821 [message] => hello+john))) </pre>
getMMS	
Parameters	<pre> mms => ID for a specific MMS (Example: 1918) from => Start Date for Filtering MMSs (Example: '2014-03-30') - Default value: Today to => End Date for Filtering MMSs (Example: '2014-03-30') - Default value: Today type => Filter MMSs by Type (Boolean: 1 = received / 0 = sent) did => DID number for Filtering MMSs (Example: 5551234567) contact => Contact number for Filtering MMSs (Example: 5551234567) limit => Number of records to be displayed (Example: 20) - Default value: 50 timezone=> Adjust time of MMSs according to Timezone (Numeric: -12 to 13) all_messages=> Filter to recive all MMSs and SMSs, 1 recive all SMS and MMS, 0 if only need MMS, important: the sms ID must be 0 </pre>
Output	<pre> Array ([status] => success [sms] => Array ([0] => Array ([id] => 111120 [date] => 2014-03-30 10:24:16 [type] => 0 [did] => 8574884828 [contact] => 8577884821 [message] => hello+john [col_media1] => "https://voip.ms/media.php?map=MTU5MzUzMdCtk2NNUw==" [col_media2] => "" [col_media3] => "" [media] => Array ([0] => 'https://voip.ms/media.php?map=MTU5MzUzMdCtk2NNUw==' [1] => '' [2] => '')))) </pre>
getMediaMMS	
Parameters	<pre> id => ID for a specific MMS (Example: 1918) media_as_array => Return the list of media attachments as an Array if the value is 1 or as a JSON Object if the value is 0 (Default: 0) </pre>
Output	<pre> Array ([status] => success [id] => 272377221 [date] => 2014-03-30 10:24:16 [media] => ([0] => 'https://voip.ms/media.php?map=MTU5MzUzMdCtk2NNUw==' [1] => '' [2] => '')) </pre>
getProvinces	
Parameters	No Parameter
Output	<pre> Array ([status] => success [provinces] => Array ([0] => Array ([province] => ON [description] => ONTARIO) [1] => Array ([province] => QC [description] => QUEBEC) [2] => Array ([province] => AB [description] => ALBERTA))) </pre>
getQueues	
Parameters	<pre> queue => ID for a specific Queue (Example: 4764) </pre>

Output	<pre>Array ([status] => success [queues] => Array ([0] => Array ([queue] => 3008 [queue_name] => call_center5 [queue_number] => 6 [queue_language] => en [queue_password] => [callerid_prefix] => [join_announcement] => none [priority_weight] => 1 [agent_announcement] => 0 [report_hold_time_agent] => yes [member_delay] => 10 [music_on_hold] => jazz [maximum_wait_time] => 60 [maximum_callers] => 10 [join_when_empty] => no [leave_when_empty] => yes [ring_strategy] => rrmemory [ring_inuse] => yes [agent_ring_timeout] => 5 [retry_timer] => 5 [wrapup_time] => 20 [voice_announcement] => 0 [frequency_announcement] => none [announce_position_frequency] => none [announce_round_seconds] => none [if announce_position_enabled_report_estimated_hold_time] => yes [thankyou_for_your_patience] => yes [fail_over_routing_timeout] => fwd:19999 [fail_over_routing_full] => vm:149601 [fail_over_routing_join_empty] => fwd:15555 [fail_over_routing_leave_empty] => none: [fail_over_routing_join_unavail] => none: [fail_over_routing_leave_unavail] => none:))))</pre>
--------	---

getRateCentersCAN	
--------------------------	--

Parameters	province => [Required] Canadian Province (Values from getProvinces)
Output	<pre>Array ([status] => success [ratecenters] => Array ([0] => Array ([ratecenter] => Calgary [available] => yes) [1] => Array ([ratecenter] => Canmore [available] => yes) [2] => Array ([ratecenter] => Edmonton [available] => yes)))</pre>

getRateCentersUSA	
--------------------------	--

Parameters	state => [Required] United States State (Values from getStates)
Output	<pre>Array ([status] => success [ratecenters] => Array ([0] => Array ([ratecenter] => ALFORD [available] => no) [1] => Array ([ratecenter] => APOPKA [available] => yes) [2] => Array ([ratecenter] => ARCADIA [available] => yes)))</pre>

getStates	
------------------	--

Parameters	No Parameter
Output	<pre>Array ([status] => success [states] => Array ([0] => Array (</pre>

	<pre> [state] => AL [description] => ALABAMA) [1] => Array ([state] => AR [description] => ARKANSAS) [2] => Array ([state] => AZ [description] => ARIZONA))) </pre>
getStaticMembers	
Parameters	<pre> queue => [Required] ID for a specific Queue (Example: 4136) member => ID for a specific Static Member (Example: 163) - The Member must belong to the queue provided </pre>
Output	<pre> Array ([status] => success [members] => Array ([0] => Array ([member] => 163 [queue_number] => 4136 [name] => as [account] => SIP/10000_johnsmith [priority] => 0))) </pre>
getTimeConditions	
Parameters	<pre> timecondition => ID for a specific Time Condition (Example: 1830) </pre>
Output	<pre> Array ([status] => success [timecondition] => Array ([0] => Array ([timecondition] => 1830 [name] => Johns Work [routing_match] => account:100001 [routing_nomatch] => vm:101 [starthour] => 8;8 [startminute] => 0;0 [endhour] => 16;12 [endminute] => 0;0 [weekdaystart] => mon;sat [weekdayend] => fri;sat))) </pre>
getVoicemailSetups	
Parameters	<pre> voicemailsetup => ID for a specific Voicemail Setup (Example: 2) </pre>
Output	<pre> Array ([status] => success [voicemailsetups] => Array ([0] => Array ([value] => 2 [description] => Account Voicemail))) </pre>
getVoicemailAttachmentFormats	
Parameters	<pre> email_attachment_format => ID for a specific attachment format (Example: wav49) </pre>
Output	<pre> Array ([status] => success [email_attachment_formats] => Array ([0] => Array ([value] => wav49 [description] => Recommended Smaller File Size) [1] => Array ([value] => wav [description] => Uncompressed Bigger File Size))) </pre>

orderDID	
	did => [Required] DID to be Ordered (Example: 5552223333) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID note => Note for the DID billing_type => [Required] Billing type for the DID (1 = Per Minute, 2 = Flat) account => Reseller Sub Account (Example: '100001_VoIP') monthly => Montly Fee for Reseller Client (Example: 3.50) setup => Setup Fee for Reseller Client (Example: 1.99) minute => Minute Rate for Reseller Client (Example: 0.03) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo'
Parameters	
Output	Array ([status] => success)
orderDIDInternationalGeographic	
	location_id => [Required] ID for a specific International Location (Values from getDIDsInternationalGeographic) quantity => [Required] Number of dids to be purchased (Example: 2) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID billing_type => [Required] Billing type for the DID (1 = Per Minute, 2 = Flat) note => Note for the DID account => Reseller Sub Account (Example: '100001_VoIP') monthly => Montly Fee for Reseller Client (Example: 3.50) setup => Setup Fee for Reseller Client (Example: 1.99) minute => Minute Rate for Reseller Client (Example: 0.03) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo'
Parameters	
Output	Array ([status] => success)

orderDIDInternationalNational	
Parameters	location_id => [Required] ID for a specific International Location (Value from getDIDsInternationalNational) quantity => [Required] Number of dids to be purchased (Example: 2) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID billing_type => [Required] Billing type for the DID (1 = Per Minute, 2 = Flat) note => Note for the DID account => Reseller Sub Account (Example: '100001_VoIP') monthly => Monthly Fee for Reseller Client (Example: 3.50) setup => Setup Fee for Reseller Client (Example: 1.99) minute => Minute Rate for Reseller Client (Example: 0.03) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo'
Output	<pre> Array ([status] => success) </pre>
orderDIDInternationalTollFree	
Parameters	location_id => [Required] ID for a specific International Location (Value from getDIDsInternationalTollFree) quantity => [Required] Number of dids to be purchased (Example: 2) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID note => Note for the DID account => Reseller Sub Account (Example: '100001_VoIP') monthly => Monthly Fee for Reseller Client (Example: 3.50) setup => Setup Fee for Reseller Client (Example: 1.99) minute => Minute Rate for Reseller Client (Example: 0.03) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo'
Output	<pre> Array </pre>

	<pre>([status] => success)</pre>
orderDIDVirtual	
Parameters	<pre> digits => [Required] Three Digits for the new Virtual DID (Example: 001) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID note => Note for the DID account => Reseller Sub Account (Example: '100001_VoIP') monthly => Monthly Fee for Reseller Client (Example: 3.50) setup => Setup Fee for Reseller Client (Example: 1.99) minute => Minute Rate for Reseller Client (Example: 0.03) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made </pre> routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: <pre> account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo' </pre>
Output	<pre> Array ([status] => success) </pre>
orderTollFree	
Parameters	<pre> did => [Required] DID to be Ordered (Example: 8772223333) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID note => Note for the DID account => Reseller Sub Account (Example: '100001_VoIP') monthly => Monthly Fee for Reseller Client (Example: 3.50) setup => Setup Fee for Reseller Client (Example: 1.99) minute => Minute Rate for Reseller Client (Example: 0.03) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made </pre> routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: <pre> account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo' </pre>

Output	<pre> Array ([status] => success) </pre>
orderVanity	
Parameters	<pre> did => [Required] DID to be Ordered (Example: 8772223333) routing => [Required] Main Routing for the DID failover_busy => Busy Routing for the DID failover_unreachable => Unreachable Routing for the DID failover_noanswer => NoAnswer Routing for the DID voicemail => Voicemail for the DID (Example: 101) pop => [Required] Point of Presence for the DID (Example: 5) dialtime => [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam => [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix => Caller ID Prefix for the DID note => Note for the DID carrier => [Required] Carrier for the DID (Values from getCarriers) account => Reseller Sub Account (Example: '100001_VoIP') monthly => Montly Fee for Reseller Client (Example: 3.50) setup => Setup Fee for Reseller Client (Example: 1.99) minute => Minute Rate for Reseller Client (Example: 0.03) test => Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made </pre> routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: <pre> account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup = Hangup the Call busy = Busy tone noservice = System Recording: Number not in service disconnected = System Recording: Number has been disconnected dtmf = DTMF Test echo = Sound Quality Test none Used to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo' </pre>
Output	<pre> Array ([status] => success) </pre>
removeDIDvPRI	
Parameters	<pre> vpri => Id for specific Vpri did => [Required] DID Number to be remove from our Vpri (Example: 561115) </pre>
Output	<pre> Array ([status] => success [vpri] => 358 [DIDRemoved] => 561115) </pre>
searchDIDsCAN	
Parameters	<pre> province => Canadian Province (Values from getProvinces) type => [Required] Type of search (Values: 'starts', 'contains', 'ends') query => [Required] Query for searching (Examples: 'JOHN', '555', '123ABC') </pre>
Output	<pre> Array ([status] => success [dids] => Array ([0] => Array ([did] => 4502381568 [ratecenter] => Chomedey [province] => QC [province_description] => QUEBEC [perminute_monthly] => 0.99 [perminute_minute] => 0.0100 [perminute_setup] => 0.50 [flat_monthly] => 4.95 [flat_minute] => 0.00 [flat_setup] => 1.00)))) </pre>
searchDIDsUSA	
Parameters	<pre> state => United States State (Values from getStates) </pre>

	type=> [Required] Type of search (Values: 'starts', 'contains', 'ends') query=> [Required] Query for searching (Examples: 'JOHN', '555', '123ABC')
Output	<pre>Array ([status] => success [dids] => Array ([0] => Array ([did] => 7863643023 [ratecenter] => MIAMI [state] => FL [state_description] => FLORIDA [perminute_monthly] => 0.99 [perminute_minute] => 0.01 [perminute_setup] => 0.50 [flat_monthly] => 4.95 [flat_minute] => 0.00 [flat_setup] => 1.00))))</pre>
searchTollFreeCanUS	
Parameters	type=> Type of search (Values: 'starts', 'contains', 'ends') query=> Query for searching (Examples: 'JOHN', '555', '123ABC')
Output	<pre>Array ([status] => success [dids] => Array ([0] => Array ([did] => 8774530644 [monthly] => 1.49 [minute] => 0.0320 [setup] => 0.00))))</pre>
searchTollFreeUSA	
Parameters	type=> Type of search (Values: 'starts', 'contains', 'ends') query=> Query for searching (Examples: 'JOHN', '555', '123ABC')
Output	<pre>Array ([status] => success [dids] => Array ([0] => Array ([did] => 8888355579 [monthly] => 0.99 [minute_usa] => 0.019 [minute_canada] => 0.08 [minute_puertorico] => 0.095 [minute_alaska] => 0.17 [setup] => 0))))</pre> Note: Canada, Puerto Rico and Alaska are Unlocked on Request
searchVanity	
Parameters	type=> [Required] Type of Vanity Number Values: '8**', '800', '833', '844', '855', '866', '877', '888' query=> [Required] Query for searching : 7 Chars Examples: '***JHON', '**555**', '**HELLO'
Output	<pre>Array ([status] => success [dids] => Array ([0] => Array ([did] => 8776605646 [monthly_american] => 0.99 [monthly_canadian] => 1.49 [minute_american_usa] => 0.019 [minute_american_canada] => 0.08 [minute_canadian] => 0.032 [setup_american] => 15 [setup_canadian] => 30) [1] => Array ([did] => 8776645646 [monthly_american] => 0.99 [monthly_canadian] => 1.49 [minute_american_usa] => 0.019 [minute_american_canada] => 0.08 [minute_canadian] => 0.032 [setup_american] => 15 [setup_canadian] => 30))))</pre>
sendSMS	

Parameters	did=> [Required] DID Numbers which is sending the message (Example: 5551234567) dst=> [Required] Destination Number (Example: 5551234568) message=> [Required] Message to be sent (Example: 'hello John Smith' max chars: 160)
Output	Array ([status] => success [sms] => 23434)
sendMMS	
Parameters	did=> [Required] DID Numbers which is sending the message (Example: 5551234567) dst=> [Required] Destination Number (Example: 5551234568) message=> [Required] Message to be sent (Example: 'hello John Smith' max chars: 2048) media1=> [Optional] Url to media file (Example: 'https://voip.ms/themes/voipms/assets/img/talent.jpg?v=2') media2=> [Optional] Base 64 image encode (Example: ...) media3=> [Optional] Empty value (Example: '') Requests can be made by the GET and POST methods. When sending multimedia via POST and base64, the file limit is based on the maximum allowed per message, 1.2 mb per file. When sending multimedia via GET and base64, the file limit is based on the maximum allowed by the GET request type, which supports a length of 512 characters, approximately 160kb total weight. In both GET and POST when using file URL submission, this limitation does not exist.
Output	Array ([status] => success [mms] => 23434)
setCallback	
Parameters	callback=> ID for a specific Callback (Example: 2359 / Leave empty to create a new one) description=> [Required] Description for the Callback number=> [Required] Number that will be called back delay_before=> [Required] Delay befor calling back response_timeout=> [Required] Time before hanging up for incomplete input digit_timeout=> [Required] Time between digits input callerid_number=> Caller ID Override for the callback
Output	Array ([status] => success [callback] => 2359)
setCallerIDFiltering	
Parameters	filter=> ID for a specific Caller ID Filtering (Example: 18915 / Leave empty to create a new one) callerid=> [Required] Caller ID that triggers the Filter (i = Not North American format, 0 = Anonymous, NPANXXXXXX, s or sb or sc = STIR/SHAKEN) Attestation Level, p = All Phone Book, p:XXXX = Specific Phone Book Group) did=> [Required] DIDs affected by the filter (all, NPANXXXXXX) routing=> [Required] Route the call follows when filter is triggered failover_unreachable=> Route the call follows when unreachable failover_busy=> Route the call follows when busy failover_noanswer=> Route the call follows when noanswer note=> Note for the Caller ID Filtering
Output	Array ([status] => success [filtering] => 18915)
setCallHunting	
Parameters	callhunting=> ID for a specific Call Hunting (Example: 235 / Leave empty to create a new one) description=> [Required] Description for the Call Hunting music=> [Required] Music on Hold Code (Values from getMusicOnHold) recording=> [Required] Recording for the Call Hunting (values from getRecordings) language=> [Required] Language for the Call Hunting (values from getLanguages) order=> [Required] The members will be called in follow or random order (values follow or random) members=> [Required] The list of members assigned to the call hunting ring_time=> [Required] The Maximum amount of time the call will ring the member press=> [Required] This option confirm if the member will take the call by pressing 1 * ring_time and press parameters need to have the same amount of items as the members parameter, one for each member.
Output	Array ([status] => success [call_hunting] => 235)
setDIDBillingType	
Parameters	did=> [Required] DID affected by the new billing plan billing_type=> [Required] Billing type for the DID (1 = Per Minute, 2 = Flat)
Output	Array ([status] => success)

setDIDInfo	
Parameters	did=> [Required] DID to be Updated (Example: 5551234567) routing=> [Required] Main Routing for the DID failover_busy=> Busy Routing for the DID failover_unreachable=> Unreachable Routing for the DID failover_noanswer=> NoAnswer Routing for the DID voicemail=> Voicemail for the DID pop=> [Required] Point of Presence for the DID ("server_pop" values from getServersInfo. Example: 3) dialtime=> [Required] Dial Time Out for the DID (Example: 60 -> in seconds) cnam=> [Required] CNAM for the DID (Boolean: 1/0) callerid_prefix=> Caller ID Prefix for the DID record_calls=> Record Calls (Boolean: 1/0) note=> Note for the DID port_out_pin=> Port Out PIN protection is used as a means of authorizing outgoing portability (only for selected US numbers with the lock icon) billing_type=> [Required] Billing type for the DID (1 = Per Minute, 2 = Flat) transcribe=> Enable Call Transcription (Boolean: 1/0) transcription_locale=> Call Transcription Locale (values from getLocales) transcription_email=> Call Transcription Email sms_sipaccount_enabled=> If Enable, SMS Messages received by your DID will be sent to the specified SIP account or sub-account (Values:1=Enable / 0=Disable) sms_sipaccount=> SIP account or sub-account that will receive the SMS Messages transcription_start_delay=> Call Transcription Delay Seconds between 0 and 60, Increments of 5 (Example: 10 -> seconds) voicemail_threshold=> Voicemail Threshold Seconds between 0 and 60, Increments of 5 (Example: 10 -> seconds) routing, failover_busy, failover_unreachable and failover_noanswer can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: accountUsed for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwdUsed for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vmUsed for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function sys System Options: hangup= Hangup the Call busy= Busy tone noservice= System Recording: Number not in service disconnected= System Recording: Number has been disconnected dtmf= DTMF Test echo= Sound Quality Test noneUsed to route calls to no action Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101' 'none:' 'sys:echo'
Output	Array ([status] => success)
setDIDPOP	
Parameters	did=> [Required] DID to be Updated (Example: 5551234567) pop=> [Required] Point of Presence for the DID ("server_pop" values from getServersInfo. Example: 3)
Output	Array ([status] => success)
setDIDRouting	
Parameters	did=> [Required] DID to be Updated (Example: 5551234567) routing=> [Required] Main Routing for the DID routing can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none. Examples: accountUsed for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwdUsed for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vmUsed for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101'
Output	Array ([status] => success)
setDIDVoicemail	
Parameters	did=> [Required] DID to be Updated (Example: 5551234567) voicemail=> Mailbox for the DID

Output	<pre>Array ([status] => success)</pre>		
setDISA			
Parameters	disa =>	ID for a specific DISA (Example: 2114 / Leave empty to create a new one)	
	name =>	[Required]	Name for the DISA
	pin =>	[Required]	Password for the DISA
	digit_timeout =>	[Required]	Time between digits
	callerid_override =>		Caller ID Override for the DISA
Output	<pre>Array ([status] => success [disa] => 2114)</pre>		
setForwarding			
Parameters	forwarding =>	ID for a specific Forwarding (Example: 19183 / Leave empty to create a new one)	
	phone_number =>	[Required]	Phone Number for the Forwarding
	callerid_override =>		Caller ID Override for the Forwarding
	description =>		Description for the Forwarding
	dtmf_digits =>		Send DTMF digits when call is answered
	pause =>		Pause (seconds) when call is answered before sending digits (Example: 1.5 / Values: 0 to 10 in increments of 0.5)
	diversion_header =>		If enabled, we will add a Diversion Header to your forwarded call
Output	<pre>Array ([status] => success [forwarding] => 19183)</pre>		
setIVR			
Parameters	ivr =>	ID for a specific IVR (Example: 4636 / Leave empty to create a new one)	
	name =>	[Required]	Name for the IVR
	recording =>	[Required]	Recording for the IVR (values from getRecordings)
	timeout =>	[Required]	Maximum time for type in a choice after recording
	language =>	[Required]	Language for the IVR (values from getLanguages)
	voicemailsetup =>	[Required]	Voicemail Setup for the IVR (values from getVoicemailSetups)
	choices =>	[Required]	Choices for the IVR (Example: '1=sip:5096 ; 2=fwd:20222')
Output	<pre>Array ([status] => success [ivr] => 4636)</pre>		
setPhonebook			
Parameters	phonebook =>	ID for a specific Phonebook entry (Example: 32207 / Leave empty to create a new one)	
	speed_dial =>		Speed Dial for the Phonebook entry
	name =>	[Required]	Name for the Phonebook Entry
	number =>	[Required]	Number or SIP for the Phonebook entry (Example: 'sip:2563' or '5552223333')
	callerid =>		Caller ID Override when dialing via Speed Dial
	note =>		Note for the phonebook entry
	group =>		ID for a specific Phonebook group
Output	<pre>Array ([status] => success [phonebook] => 32207)</pre>		
setPhonebookGroup			
Parameters	group =>	ID for a specific Phonebook group (Example: 32207 / Leave empty to create a new one)	
	name =>	[Required]	Name for the Phonebook group
	members =>		Phonebook entry codes associated to this group separated by a semicolon
Output	<pre>Array ([status] => success [group] => 32207)</pre>		
setRecording			
Parameters	recording =>	ID for a specific Phonebook entry (Example: 33221 / Leave empty to create a new one)	
	file =>	[Required]	Base64 encoded file (Provide Recording ID and file if you want update the file only)
	name =>	[Required]	Name for the Recording Entry (Example: 'recording1') (Provide Recording ID and name if you want update the name only) (Provide Recording ID, file and name if you want update both parameters at the same time)
Output	<pre>Array ([status] => success)</pre>		

	)	[recording] => 33221
setQueue		
Parameters	queue => ID for a specific Queue entry (Example: 32208 / Leave empty to create a new one) queue_name => [Required] Queue entry name queue_number => [Required] Queue entry number queue_language => [Required] Language Code (Values from getLanguages) queue_password => Queue Password callerid_prefix => Caller ID Prefix for queue join_announcement => Recording Code (Values from getRecordings or 'none') priority_weight => [Required] weight/priority of queue (Values 1 to 60) agent_announcement => Recording Code (Values from getRecordings or 'none') report_hold_time_agent => [Required] Report hold time to agent (Values from getReportEstimatedHoldTime) member_delay => Member delay when the agent is connected to the caller (Values 1 to 15 in seconds or 'none') maximum_wait_time => Ammount of time a caller can wait in queue	(Values in seconds: multiples of 30, max value: 1200 or 'unlimited') maximum_callers => Maximum callers (Values: 1 to 60 or 'unlimited') join_when_empty => [Required] How caller join to the queue (Values from getJoinWhenEmptyTypes)
	leave_when_empty => [Required] How caller leave the queue (Values 'yes'/'no'/'strict') Examples: yes Callers can join a queue with no members or only unavailable members no Callers cannot join a queue with no members strict Callers cannot join a queue with no members or only unavailable members ring_strategy => [Required] Ring strategy (Values from getRingStrategies) ring_inuse => [Required] If you want the queue to avoid sending calls to members (Values 'yes'/'no') agent_ring_timeout => Number of seconds to ring an agent (Values 5 to 60) retry_timer => How long do we wait before trying all the members again (Values 5 to 60 seconds or 'none'= No Delay) wrapup_time => After a successful call, the number of seconds to wait before sending a free agent another call	(Values 1 to 60 seconds or 'none'= No Delay) voice_announcement => Code for Recording (Values from getRecordings or 'none') frequency_announcement => Periodic interval to play voice announce recording
Parameters	announce_position_frequency => (Values in seconds: multiples of 15, max value: 1200 or 'none' = No announcement) How often to make any periodic announcement	
	announce_round_seconds => (Values in seconds: multiples of 15, max value: 1200 or 'none' = No announcement) Announce seconds (Values in seconds: 1 to 60 or 'none' = Do not announce)	
Parameters	if_announce_position_enabled_report_estimated_hold_time => Include estimated hold time in position announcements (Values 'yes'/'no'/'once')	
	thankyou_for_your_patience => Yes to say "Thank you for your patience" immediatly after announcing Queue Position and Estimated hold time left	
Parameters	music_on_hold => (Values 'yes'/'no') Music on Hold Code (Values from getMusicOnHold) fail_over_routing_timeout => Failover routing to Maximum wait time reached fail_over_routing_full => Failover routing to Maximum callers reached fail_over_routing_join_empty=> A call was sent to the queue but the queue had no members (Only works when Join when Empty is set to no) fail_over_routing_leave_empty=> The last agent was removed form the queue before alls calls were handled	
	fail_over_routing_join_unavail=> (Only works when Leave when Empty is set to yes) Same as routingjoinempty, except that there were still queue members, but all were status unavailable fail_over_routing_leave_unavail=> Same as routingleaveempty, except that there were still queue members, but all were status unavailable	
Parameters	routings can receive values in the following format => header:record_id Where header could be: account, fwd, vm, sip, grp, ivr, sys, recording, queue, cb, tc, disa, none.	
	Examples: account Used for routing calls to Sub Accounts You can get all sub accounts using the getSubAccounts function fwd Used for routing calls to Forwarding entries. You can get the ID right after creating a Forwarding with setForwarding or by requesting all forwardings entries with getForwardings. vm Used for routing calls to a Voicemail. You can get all voicemails and their IDs using the getVoicemails function Examples: 'account:100001_VoIP' 'fwd:1026' 'vm:101'	
Output	Array ([status] => success [queue] => 32208)	
setRingGroup		
Parameters	ring_group => ID for a specific Ring Group (Example: 4768 / Leave empty to create a new one) name => [Required] Name for the Ring Group members => [Required] Members for the Ring Group (Example: 'account:100001;fwd:16006') voicemail => [Required] Voicemail for the Ring Group (Values from getVoicemails) caller_announcement => Recording Code (Values from getRecordings) music_on_hold => Music on Hold Code (Values from getMusicOnHold) language => Code for Language (Values from getLanguages)	
	"members" can receive the following routing headers: account Used to route the call to an Account or a Sub Account fwd Used to route the call to a Forwarding entry sip Used to route the call to a SIP URI Each member can a specific ring time and press1 values, you can add those to the routing header as follow: Example: 'account:100001,25,0;fwd:16006,10,1' 25 = Default ring time value (Values from 1 to 60 sec) 0 = Default press1 value (Values allowed 0 and 1)	
Output	Array ([status] => success	

	) [ring_group] => 4768		
setSIPURI			
Parameters	sipuri uri description callerid_override callerid_e164	=> => [Required] => => => =>	ID for a specific SIP URI (Example: 6199 / Leave empty to create a new one) SIP URI (Example: '555223333@sip.voip.ms') Description for the SIP URI This setting is optional. You can configure a 'CallerID Number Override'. Your default CallerID number will be changed to the override you have configured here when using this SIP URI. If this option is Enabled, then your CallerID will be E164 compliant.
setSMS			
Parameters	did enable email_enabled email_address sms_forward_enable sms_forward url_callback_enable url_callback url_callback_retry webhook_enable webhook dialmode smpp_enabled smpp_url smpp_user smpp_pass sms_sipaccount_enabled sms_sipaccount	=> [Required] => [Required] => => => => => => => => => => => => => => => =>	DID to be Updated (Example: 5551234567) Enable/Disable the DID to receive SMS Messages (Values:1=Enable / 0=Disable) If Enable, SMS Messages received by your DID will be sent to the email address provided (Values:1=Enable / 0=Disable) SMS Messages received by your DID will be sent to the email address provided If Enable, SMS Messages received by your DID will be forwarded to the phone number provided (Values:1=Enable / 0=Disable) SMS Messages received by your DID will be forwarded to the phone number provided (Example: 5551234567) If Enable, SMS Messages received by your DID will be send a GET request to the URL callback provided (Values:1=Enable / 0=Disable) SMS Messages received by your DID will be send a GET request to the URL callback provided Available Variables for your URL {FROM} The phone number that sent you the message. {TO} The DID number that received the message. {MESSAGE} The content of the message. Example: http://mysite.com/sms.php?to={TO}&from={FROM}&message={MESSAGE} Enable URL callback Retry (Values:1=Enable / 0=Disable) we will be expecting an 'ok' output (without quotes) from your URL callback page as an indicator that you have received the message correctly. If we don't receive the 'ok' letters (wihntout quotes) from your callback page, we will keep sending you the same message every 30 minutes. Enable/Disable URL Webhook (Values:1=Enable / 0=Disable) SMS/MMS Webhook URL SMS/MMS Messages received by your DID will send a POST request to your 3CX or PortSIP PBX using the Webhook URL provided. Dialing Mode for SMS (API) (Values:1=NANPA / 2=E164) NANPA: Your messages will be sent using only 10 digits (without the +1 prefix). E164: Your messages will be sent using Country Code first (with the +1 prefix) If Enable, SMS Messages received by your DID will be sent to the specified SMPP URL (Values:1=Enable / 0=Disable) SMS Messages received by your DID will be sent to the specified SMPP URL using the submit_sm command. The URL needs to specify if we should send the message encrypted or not using smpp (not encrypted) or ssmpp (encrypted). You can also specify the username, password, and the port in the URL. The SMPP Username that will be used to authenticate you. Outgoing messages should be sent to smpp.voip.ms through the regular 2775 port or the encrypted 3550 port. These messages can be sent using either deliver_sm or submit_sm commands. The SMPP Password that will be used to authenticate you. Outgoing messages should be sent to smpp.voip.ms through the regular 2775 port or the encrypted 3550 port. These messages can be sent using either deliver_sm or submit_sm commands. If Enable, SMS Messages received by your DID will be sent to the specified SIP account or sub-account (Values:1=Enable / 0=Disable) SIP account or sub-account that will receive the SMS Messages
Output	Array ([status] => success [sipuri] => 6199)		
setStaticMember			
Parameters	member queue member_name account priority	=> => [Required] => [Required] => => [Required]	ID for a specific Member (Example: 619 / Leave empty to create a new one) ID for a specific Queue Member Description Static Member Routing to receive calls - You can get all sub accounts using the getSubAccounts function Values for get calls first (Example: 0)
Output	Array ([status] => success [member] => 619)		
setTimeCondition			
Parameters	timecondition name routing_match routing_nomatch starthour startminute endhour endminute weekdaystart weekdayend	=> => [Required] => [Required] => [Required] => [Required] => [Required] => [Required] => [Required] => [Required] => [Required]	ID for a specific Time Condition (Example: 1830 / Leave empty to create a new one) Name for the Time Condition Routing for the Call when condition matches Routing for the Call when condition does not matche All the Start Hour Conditions (Example: '8;8') All the Start Minute Conditions (Example: '0;0') All the End Hour Conditions (Example: '16;12') All the End Minute Conditions (Example: '0;0') All the Week Day Start Conditions (Example: 'mon;sat') All the Week Day End Conditions (Example: 'fri;sat')
Output	Array ([status] => success [timecondition] => 1830)		
disconnectDID			

Parameters	did => [Required] DID to be Unconnected from Reseller Sub Account(Example: 5551234567)
Output	<pre> Array ([status] => success) </pre>

FAX FUNCTIONS

The following table explains the additional parameters needed by the Voicemail Functions and provides an example of their output when status is 'success'.

connectFAX	
Parameters	did => [Required] FAX DID to be Connected to Reseller Sub Account (Example: 5551234567) account => [Required] Reseller Sub Account (Example: '100001_VoIP') monthly => [Required] Montly Fee for Reseller Client (Example: 3.50) setup => [Required] Setup Fee for Reseller Client (Example: 1.99) minute => [Required] Minute Rate for Reseller Client (Example: 0.03) next_billing => Next billing date (Example: '2014-03-30') dont_charge_setup => If set to true, the setup value will not be charged after Connect dont_charge_monthly => If set to true, the monthly value will not be charged after Connect
Output	<pre> Array ([status] => success) </pre>
unconnectFAX	
Parameters	did => [Required] FAX DID to be Unconnected from Reseller Sub Account (Example: 5551234567)
Output	<pre> Array ([status] => success) </pre>
cancelFaxNumber	
Parameters	id => [Required] ID for a specific Fax Number (Example: 923) test => Set to true if testing how cancel a Fax Number
Output	<pre> Array ([status] => success [deleted_did] => 5551234567) </pre>
deleteFaxMessage	
Parameters	id => [Required] ID for a specific Fax Message (Example: 923) test => Set to true if testing how cancel a Fax Message
Output	<pre> Array ([status] => success) </pre>
delEmailToFax	
Parameters	id => [Required] ID for a specific "Email To Fax Configuration" (Example: 923) test => Set to true if testing how cancel a "Email To Fax Configuration"
Output	<pre> Array ([status] => success) </pre>
delFaxFolder	
Parameters	id => [Required] ID for a specific Fax Folder (Example: 923) test => Set to true if testing how to delete a Fax Folder
Output	<pre> Array ([status] => success) </pre>
getBackOrders	
Parameters	id => ID for a specific backorder DID
Output	<pre> Array ([status] => success [quantity] => 1 [back_orders] => Array (</pre>

	<pre>[0] => Array ([id] => 1234 [description] => MIAMI, FL [routing] => account:100001_VoIP [failover_busy] => none: [failover_unreachable] => fwd:1521 [failover_noanswer] => none: [voicemail] => 1001 [pop] => 16 [dialtime] => 60 [cnam] => 1 [billing_type] => 2 [order_date] => 2014-07-30 15:09:04)</pre>
--	---

getFaxProvinces

Parameters	province => CODE for a specific Province (Example: AB)
Output	<pre>Array ([status] => success [provinces] => Array ([0] => Array ([province] => AB [province_id] => 60 [province_name] => Alberta [country_code] => CAN)))</pre>

getFaxStates

Parameters	state => CODE for a specific State (Example: AL)
Output	<pre>Array ([status] => success [states] => Array ([0] => Array ([state] => AL [state_id] => 1 [state_name] => Alabama [country_code] => USA)))</pre>

getFaxRateCentersCAN

Parameters	province => [Required] Province two letters code (Example: AB)
Output	<pre>Array ([status] => success [ratecenters] => Array ([0] => Array ([location] => 20294 [area_code] => 587 [ratecenter] => EDMONTON [available] => yes)))</pre>

getFaxRateCentersUSA

Parameters	state => [Required] Province two letters code (Example: AL)
Output	<pre>Array ([status] => success [ratecenters] => Array ([0] => Array ([location] => 1196 [area_code] => 205 [ratecenter] => ALABASTER [available] => yes)))</pre>

getFaxNumbersInfo

Parameters	did => Fax Number to retrieves the information of a single number, or not send if you want retrieves the information of all your Fax Numbers.
Output	<pre>Array ([status] => success [numbers] => Array ([0] => Array (</pre>

		<pre> [id] => 0000 [did] => 5551234567 [description] => United States, CA, HALF MOON BAY [state] => CA [city] => HALF MOON BAY [country] => United States [email_enabled] => 1 [email] => address@email.com [url_enabled] => 1 [url] => www.example.com [retry] => 0 [attach_file] => 1 [note] => "testing" [reseller_account] => 100001_VoIPReseller [reseller_next_billing] => 2014-08-30 [reseller_monthly] => 0.00000000 [reseller_minute] => 0.01000000 [reseller_setup] => 0.00000000 [fax_to_sip_enabled] => 1 [fax_to_sip_enabled_account] => 100001_VoIPReseller2 </pre>
	)	)
	)	)
getFaxNumbersPortability		
Parameters	did	=> [Required] DID Number to be ported into our network (Example: 5552341234)
Output	Array	<pre> ([status] => success [getFaxNumbersPortability] => yes) </pre>
getFaxMessages		
Parameters	from to folder id	=> Start Date for Filtering Fax Messages (Example: '2014-03-30') - Default value: Today => End Date for Filtering Fax Messages (Example: '2014-03-30') - Default value: Today => Name of specific Fax Folder (Example: SENT) - Default value: ALL => ID for a Specific Fax Message (Example: 23434)
Output	Array	<pre> ([status] => success [faxes] => Array ([0] => Array ([id] => 0000 [folder] => INBOX [date] => 2016-01-01 00:00:00 [callerid] => 5552341234 [stationid] => 5552341234 [destination] => 7772341234 [description] => INBOUND FAX [pages] => 2 [duration] => 1:16 [status] => SUCCESS [rate] => 0.000 [total] => 0.000 [msg] => </pre>
	)	)
	)	)
getFaxMessagePDF		
Parameters	id	=> [Required] ID of the Fax Message requested (Values from getFaxMessages)
Output	Array	<pre> ([status] => success [message_base64] => PDF file encrypted in Base64) </pre>
getFaxFolders		
Parameters		
Output	Array	<pre> ([status] => success [folders] => Array ([0] => Array ([id] => 0000 [name] => INBOX </pre>
	)	)
	)	)
getEmailToFax		
Parameters		ID of the "Email to Fax" configuration
Output	Array	<pre> ([status] => success [emailToFax] => Array ([0] => Array ([id] => 0000 </pre>

)))

Parameters		
id	=>	[Required] ID of the Fax Message requested (Values from getFaxMessages)
email	=>	[Required] Destination email address (example: yourname@company.com)

```
Output      Array
           (
            [status] => success
            [message_status] => message_sent
           )
```

Parameters	fax_id	=>	[Required] ID of the Fax Message requested (Values from getFaxMessages)
	folder_id	=>	[Required] ID of the destination Fax Folder (Values from getFaxFolders)
	test	=>	Set to true if testing how to move a Fax Message

```
Output      Array
           (
             [status] => success
           )
```

Parameters		
	location	=> [Required] Location ID of the Fax Number (Values from getFaxRateCentersCAN/getFaxRateCentersUSA)
	quantity	=> [Required] Quantity of Fax Numbers to order (Example: 3)
	email	=> Email address where send notifications when receive Fax Messages - (Example: yourname@company.com)
	email_enable	=> Flag to enable the email notifications. - (Values: 1 = true, 0 = false) - Default: 0
	email_attach_file	=> Flag to enable attach the Fax Message as a PDF file in the notifications. - (Values: 1 = true, 0 = false) - Default: 0
	url_callback	=> URL where make a POST when you receive a Fax Message.
	url_callback_enable	=> Flag to enable the URL Callback functionality. - (Values: 1 = true, 0 = false) - Default: 0
	url_callback_retry	=> Flag to enable retry the POST action in case we don't receive "ok".
	test	=> Set to true if testing how Orders work - Orders can not be undone - When testing, no Orders are made

```

Output      Array
            (
                [status] => success
                [dids]   => array(
                    [0]   => 5552341234
                    [1]   => 5552341234
                    [2]   => 5552341234
                    ...
                )
            )

```

Parameters	name	=>	[Required] Name of the Fax Folder to create or update (Example: FAMILY)
	id	=>	[Only for updates] ID of the Fax Folder to edit (Values from getFaxFolders)
	test	=>	Set to true if testing how to create/update a Fax Folder

```
Output      Array
           (
             [status] => success
           )
```

Parameters		
	id	=> [Only for updates] ID of the "Email to Fax" to edit (Values from getEmailToFax)
	enabled	=> If Enable, we will send Fax Message when we receive an email from the provided address. (Values:1=Enable / 0=Disable)
	auth_email	=> [Required] Email address from you will sent Fax Messages
	from_number_id	=> [Required] Fax number that will appear as fax sender. (values from getFaxNumbersInfo)
	security_code_enabled	=> If Enable, we will check the mail subject if this include a Security Code before send the Fax. (Values:1=Enable / 0=Disable)
	security_code	=> [Required] An alphanumeric code to identify your emails before send as Fax.
	test	=> Set to true if testing.

```
Output      Array
           (
            [status] => success
           )
```

Parameters	area code => [Required] Area code number, as the initial of the Fax Number you looking for. (values from getFaxRateCentersCAN)
------------	--

```

Output
Array
(
    [status] => success
    [ratecenters] => array(
        [0] => Array
            (
                [ratecenter] => ACTON
                [area code] => 226
            )
    )
)

```

[available] => yes

```
)
)
```

searchFaxAreaCodeUSA

Parameters area_code => [Required] Area code number, as the initial of the Fax Number you looking for. (values from getFaxRateCentersUSA)

Output

```
Array
(
    [status] => success
    [ratecenters] => array(
        [0] => array(
            [ratecenter] => COLD SPRING
            [area_code] => 845
            [available] => yes
        )
    )
)
```

setFaxNumberInfo

Parameters

did => [Required] DID Number to be ported into our network (Example: 5552341234)

email => Email address where send notifications when receive Fax Messages
- (Example: yourname@company.com)

email_enable => Flag to enable the email notifications.
- (Values: 1 = true, 0 = false) - Default: 0

email_attach_file => Flag to enable attach the Fax Message as a PDF file in the notifications.
- (Values: 1 = true, 0 = false) - Default: 0

url_callback => URL where make a POST when you receive a Fax Message.

url_callback_enable => Flag to enable the URL Callback functionality.
- (Values: 1 = true, 0 = false) - Default: 0

url_callback_retry => Flag to enable retry the POST action in case we don't receive "ok".

fax_to_sip_enabled => Flag to enable Fax to SIP.
- (Values: 1 = true, 0 = false) - Default: 0

fax_to_sip_enabled_account => Fax to SIP subaccount to use (Values from getSubaccounts) Eg: #####_SubAccount.

note => Description you wish to indicate for this DID (for your own use, but not mandatory).

test => Set to true if testing how to update the information of a Fax Number

Output

```
Array
(
    [status] => success
)
```

setFaxNumberEmail

Parameters

did => [Required] DID Number to be ported into our network (Example: 5552341234)

email => Email address where send notifications when receive Fax Messages
- (Example: yourname@company.com)

email_enable => Flag to enable the email notifications.
- (Values: 1 = true, 0 = false) - Default: 0

email_attach_file => Flag to enable attach the Fax Message as a PDF file in the notifications.
- (Values: 1 = true, 0 = false) - Default: 0

test => Set to true if testing how to set the email of a Fax Number

Output

```
Array
(
    [status] => success
)
```

setFaxNumberURLCallback

Parameters

did => [Required] DID Number to be ported into our network (Example: 5552341234)

url_callback => URL where make a POST when you receive a Fax Message.

url_callback_enable => Flag to enable the URL Callback functionality.
- (Values: 1 = true, 0 = false) - Default: 0

url_callback_retry => Flag to enable retry the POST action in case we don't receive "ok".

test => Set to true if testing how to set the URL callback of a Fax Number

Output

```
Array
(
    [status] => success
)
```

sendFaxMessage

Parameters

to_number => [Required] Destination DID Number (Example: 5552341234)

from_name => [Required] Name of the sender (Example: 5552341234)

from_number => [Required] DID number of the Fax sender

send_email_enabled => Flag to enable the send of a copy of your Fax via email.
- (Values: 1 = true, 0 = false) - Default: 0

send_email => Email address where you want send a copy of your Fax.

station_id => An word to identify a equipment or department sending the Fax.

file => [Required] The file must be encoded in Base64, and in one of the following formats: pdf, txt, jpg, gif, png, tif

test => Set to true if testing how to send a Fax Message

Output

```
Array
(
    [status] => success
)
```

The following table explains the additional parameters needed by the E911 number management and provides an example of their output when status is 'success'.

e911AddressTypes	
Parameters	type => Code for a specific Address Type (Example: Apartment)
Output	<pre>Array ([status] => success [types] => Array(Array ([value] => "Apartment", [description] => "Apartment"), Array ([value] => "Basement", [description] => "Basement")))</pre>
e911Cancel	
Parameters	did => [Required] DID to be canceled.
Output	<pre>Array ([status] => success)</pre>
e911Info	
Parameters	did => [Required] DID with e911 enabled / in process.
Output	<pre>Array ([status] => success [info] => Array ([did] => "7472127447", [status] => "2", [full_name] => "test", [street_number] => "23", [street_name] => "W BROAD ST", [address_type] => "Hanger", [address_number] => "401", [city] => "RICHMOND", [state] => "VA", [zip_code] => "12345", [country] => "US", [language] => "EN", [email] => "john.doe@my-domain.com", [other_info] => ""))</pre>
e911Provision	
Parameters	<pre>did => [Required] DID that will be sent to the e911 service. full_name => [Required] Full Name that will be sent to the e911 service. street_number => [Required] Street Number that will be sent to the e911 service. street_name => [Required] Street Name that will be sent to the e911 service. address_type => Address Type that will be sent to the e911 service (Values from e911AddressTypes). address_number => Address Number that will be sent to the e911 service. city => [Required] City that will be sent to the e911 service. state => [Required] State / Province that will be sent to the e911 service. country => [Required] Country that will be sent to the e911 service. Value can be US (United states) or CA (Canada). zip_code => [Required] Zip / Postal code that will be sent to the e911 service. language => [Required] Language that will be sent to the e911 service. Only available for addresses from Canada. Value can be EN (English) or FR (French). email => [Required] Email that will be sent to the e911 service. other_info => Additional Address Information that will be sent to the e911 service. Only available for addresses from Canada.</pre>
Output	<pre>Array ([status] => success)</pre>
e911ProvisionManually	

Parameters	did => [Required] DID that will be sent to the e911 service. full_name => [Required] Full Name that will be sent to the e911 service. street_number => [Required] Street Number that will be sent to the e911 service. street_name => [Required] Street Name that will be sent to the e911 service. address_type => Address Type that will be sent to the e911 service (Values from e911AddressTypes). address_number => Address Number that will be sent to the e911 service. city => [Required] City that will be sent to the e911 service. state => [Required] State / Province that will be sent to the e911 service. country => [Required] Country that will be sent to the e911 service. Value can be US (United states) or CA (Canada). zip_code => [Required] Zip / Postal code that will be sent to the e911 service. language => [Required] Language that will be sent to the e911 service. Only available for addresses from Canada. Value can be EN (English) or FR (French). email => [Required] Email that will be sent to the e911 service. other_info => Additional Address Information that will be sent to the e911 service. Only available for addresses from Canada.
Output	Array ([status] => success)
e911Update	
Parameters	did => [Required] DID with e911 enabled / in process. full_name => [Required] Full Name that will be updated to the e911 service. street_number => [Required] Street Number that will be updated to the e911 service. street_name => [Required] Street Name that will be updated to the e911 service. address_type => Address Type that will be updated to the e911 service (Values from e911AddressTypes). address_number => Address Number that will be updated to the e911 service. city => [Required] City that will be updated to the e911 service. state => [Required] State / Province that will be updated to the e911 service. country => [Required] Country that will be updated to the e911 service. Value can be US (United states) or CA (Canada). zip_code => [Required] Zip / Postal code that will be updated to the e911 service. language => [Required] Language that will be updated to the e911 service. Only available for addresses from Canada. Value can be EN (English) or FR (French). email => [Required] Email that will be updated to the e911 service. other_info => Additional Address Information that will be updated to the e911 service. Only available for addresses from Canada.
Output	Array ([status] => success)
e911Validate	
Parameters	did => [Required] DID with e911 enabled / in process. full_name => [Required] Full Name that will be validated. street_number => [Required] Street Number that will be validated. street_name => [Required] Street Name that will be validated. address_type => Address Type that will be validated (Values from e911AddressTypes). address_number => Address Number that will be validated. city => [Required] City that will be validated. state => [Required] State / Province that will be validated. country => [Required] Country that will be validated. Value can be US (United states) or CA (Canada). zip_code => [Required] Zip / Postal code that will be validated. language => Language that will be validated. Only available for addresses from Canada. Value can be EN (English) or FR (French). email => [Required] Email that will be validated. other_info => Additional Address Information that will be validated. Only available for addresses from Canada.
Output	Array ([status] => success)

LOCAL NUMBER PORTABILITY (LNP) FUNCTIONS

The following table explains the additional parameters needed by the LOCAL NUMBER PORTABILITY (LNP) and provides an example of their output when status is 'success'.

addLNPPort	
Parameters	numbers => [Required] DID(s) to port into VoIP.ms network (Example: 5552341234,5552341233). If you are porting more than one number, please separate them with commas. portType => [Required] Digits from 1 to 4: 1: United States Local numbers 2: Canadian Local Numbers 3: US/CA Toll Free Numbers 4: United States Fax numbers 5: Canadian Fax Numbers isPartial => If you have more then 1 number with your current carrier and not porting them all, choose yes. If you are porting all the numbers, choose no. Please note that you still need to include all numbers you want to port. If you have 2 numbers and want to port both, you need to include both numbers in the list of numbers to port. - (Values: 1 = true, 0 = false) - Default: 0 locationType => - (Values: 1 = Business, 0 = Residential) - Default: 0 => - (Values: 1 = All the numbers are mobile numbers, 0 = false) - Default: 0 isMobile

	pin => PIN Number imei => [Required] If portType = 2 or portType = 5] If you are porting your mobile number(s), please provide the IMEI*. To get the IMEI* of your phone, you can dial *#06# btn => [Required] If isMobile = 1] BTN: It is the phone number to which all the other numbers of the customer are charged, in a consolidated telephone bill (instead of showing separate charges for each number you own). Please try to find the BTN on your invoice, and if you are unable to do so please contact the current provider to obtain it. services => [Required] If isMobile = 1] Please be specific and describe ALL remaining services with the current carrier. This includes DSL/Data services, Hunt Group services, etc. Any services NOT listed below may be disconnected upon completion of this port order. tfType => [Required] If portType = 3] Values: 1 - American Carrier, American Callers Only, 2 - American Carrier, American and Canadian Callers allowed, 3 - Canadian Carrier statementName => [Required] This is for Business numbers only. Please type your Company Name if applicable, otherwise leave it blank. firstName => [Required] This is the "Customer First Name" as it appears on the CSR (Customer Service Record) of the losing carrier. Please Enter the first name of the owner of the number or the authorized contact. No company name must be entered in the field. lastName => [Required] This is the "Customer Last Name" as it appears on the CSR (Customer Service Record) of the losing carrier. Please Enter the last name of the owner of the number or the authorized contact. No company name must be entered in the field. address1 => [Required] This is the "Customer Address" as it appears on the CSR (Customer Service Record) of the losing carrier. address2 => [Required] Optional Address information (e.g: Suite 343) city => [Required] This is the "City" as it appears on the CSR (Customer Service Record) of the losing carrier. zip => [Required] This is the "ZIP or Postal Code" as it appears on the CSR (Customer Service Record) of the losing carrier. state => [Required] This is the "State or Province" as it appears on the CSR (Customer Service Record) of the losing carrier. country => [Required] This is the "Country" as it appears on the CSR (Customer Service Record) of the losing carrier. providerName => [Required] The name of your current service provider. providerAccount => [Required] Your Account with your current service provider. notes => - If you would like to include additional information regarding this port, you can use this parameter.
--	---

Output	<pre> Array ([status] => success [port] => ##### (Number ID of the created record))</pre>
--------	---

addLNPFFile

Parameters	Note: Only accepted through POST request portid => [Required] ID of the port previously created. file => [Required] Base 64 code of the file to be attached
------------	---

Output	<pre> Array ([status] => success [attachment] => ##### (Number ID of the created record))</pre>
--------	---

getLNPListStatus

Parameters	No Parameter
Output	<pre> Array ([status] => success [list_status] => Array("status_code" => "Status Description",))</pre>

getLNPList

Parameters	portid => [Required] ID of the port previously created. portStatus => Status code for filtering Ports. Example: precessing. (You can use the values returned by the method getLNPListStatus) startDate => Start Date for filtering Ports. (Example: '2014-03-30') endDate => End Date for filtering Ports. (Example: '2014-03-30')
Output	<pre> Array ([status] => success [list] => Array(Array ([portid] => "0000", [numbers] => "####,###", [foc_date] => "2019-01-08", [status] => "processing"), Array ([portid] => "1111", [numbers] => "####", [foc_date] => false, [status] => "processing")))</pre>

getLNPPStatus

Parameters	portid => [Required] ID of the port previously created.
Output	<pre> Array ([status] => success [post_status] => "pending_approval" [post_status_description] => "Pending Approval")</pre>

getLNPDDetails

Parameters	portid => [Required] ID of the port previously created.
------------	---

Output	<pre> Array ([status] => success [id] => 0000, [numbers] => Array (Array ([did] => "#####", [rateCenter] => "COVINA", [state] => "CA"),), [isPartial] => "0", [locationType] => "0", [isMobile] => "0", [mobileInfo] => "", [tfType] => "0", [portType] => "1", [btn] => "", [services] => "", [statementName] => "", [firstName] => "", [lastName] => "", [address1] => "Covina 12 St.", [address2] => "", [city] => "CA", [zip] => "", [state] => "CA", [country] => "US", [providerName] => "", [providerAccount] => "", [customer_notes] => "", [notes] => Array (Array ([note] => "You have submitted a new LNP Order. Current status is Pending Approval. A member of our staff will review your submission and update status accordingly.", [date] => "2019-02-25", [time] => "15:05:11"),), [post_status] => "pending_approval" [post_status_description] => "Pending Approval" [date] => "2019-02-25 15:05:11", [focDate] => "", [attachments] => Array (Array ([id] => "0000", [description] => "Signed Invoice", [type] => "pdf", [bytes] => "151600"),),)) </pre>
getLNPNotes	
Parameters	portid => [Required] ID of the port previously created.
Output	<pre> Array ([status] => success [list] => Array(Array ([note] => "You have submitted a new LNP Order. Current status is Pending Approval. A member of our staff will review your submission and update status accordingly.", [date] => "2019-02-25", [time] => "15:05:11"))) </pre>
getLNPAttach	
Parameters	portid => [Required] ID of the port previously created. attachid => [Required] ID of the invoice (attachment) previously uploaded.
Output	<pre> Array ([status] => success [type] => "pdf", [size] => "151600", [base64] => "Base64 Code") </pre>
getLNPAttachList	
Parameters	portid => [Required] ID of the port previously created.
Output	<pre> Array ([status] => success [list] => Array(Array ([attachid] => "000", [type] => "pdf", [size] => "151600"))) </pre>

The following table explains the additional parameters needed by the Voicemail Functions and provides an example of their output when status is 'success'.

createVoicemail	
Parameters	digits => [Required] Digits used to create the voicemail (Example: 01) Minimum 1 digit, maximum 10 digits name => [Required] Name for the Mailbox password => [Required] Password for the Mailbox skip_password => [Required] True if Skipping Password (Boolean: 1/0) email => Email address for receiving messages, multiple email addresses are allowed if separated by a comma attach_message => [Required] Yes for Attaching WAV files to Message (Values: 'yes'/'no') delete_message => [Required] Yes for Deleting Messages (Values: 'yes'/'no') say_time => [Required] Yes for Saying Time Stamp (Values: 'yes'/'no') timezone => [Required] Time Zone for Mailbox (Values from getTimeZones) say_callerid => [Required] Yes for Saying the Caller ID (Values: 'yes'/'no') play_instructions => [Required] Code for Play Instructions Setting (Values from getPlayInstructions) language => [Required] Code for Language (Values from getLanguages) client => [Optional] ID for a specific Reseller Client (Example: 561115) transcription => Voicemail Transcription transcription_locale => Transcription locale code (values from getLocales) email_attachment_format => Code for Email Attachment format (Values from getVoicemailAttachmentFormats) unavailable_message_recording => Recording for the Unavailable Message (values from getRecordings)
Output	Array ([status] => success)
delMessages	
Parameters	mailbox => [Required] ID for a specific Mailbox (Example: 1001) folder => Name for specific Folder (Required if message id is passed, Example: 'INBOX', values from: getVoicemailFolders) message_num => ID for specific Voicemail Message (Required if folder is passed, Example: 1)
Output	Array ([status] => success)
delMemberFromConference	
Parameters	member => [Required] ID for a specific Member (Example: 101) conference => ID for a specific Conference (Example: 3829)
Output	Array ([status] => success)
delVoicemail	
Parameters	mailbox => [Required] ID for a specific Mailbox (Example: 1001)
Output	Array ([status] => success)
getPlayInstructions	
Parameters	play_instructions => Code for a specific Play Instructions setting (Example: 'u')
Output	Array ([status] => success [play_instructions] => Array ([0] => Array ([value] => u [description] => Yes)))
getTimezones	
Parameters	timezone => Code for a specific Time Zone (Example: 'America/Buenos_Aires')
Output	Array ([status] => success [timezones] => Array ([0] => Array ([value] => America/Buenos_Aires [description] => America/Buenos Aires)))
getVoicemails	

Parameters	mailbox => ID for specific Mailbox (Example: 1001) client => [Optional] ID for a specific Reseller Client (Example: 561115)
Output	<pre>Array ([status] => success [voicemails] => Array ([0] => Array ([mailbox] => 1001 [name] => VoIP [password] => 1234 [skip_password] => 1 [email] => john.doe@my-domain.com [attach_message] => yes [delete_message] => no [say_time] => yes [timezone] => America/Chicago [say_callerid] => yes [play_instructions] => su [language] => en [email_attachment_format] => wav49 [unavailable_message_recording]=> 19183 [new] => 3 [urgent] => 2 [transcribe] => 1 [transcription_locale] => en-US))))</pre>

getVoicemailFolders

Parameters	folder => Folder Name (Example: 'INBOX') mailbox => ID for specific Mailbox (Example: 1001)
Output	<pre>Array ([status] => success [folders] => Array ([0] => Array ([value] => INBOX [description] => INBOX))))</pre>

getVoicemailMessageFile

Parameters	mailbox => [Required] ID for specific Mailbox (Example: 1001) folder => [Required] Name for specific Folder (Example: 'INBOX', values from: getVoicemailFolders) message_num=> [Required] ID for specific Voicemail Message (Example: 1) format => File format. Default value is mp3. Values: mp3, wav, wav49, gsm.
Output	<pre>Array ([status] => success [message] => Array ([0] => Array ([mailbox] => 1001 [folder] => INBOX [message_num] => 1 [data] => Uk1GRqTEAQBXQVZFZm10IBAAAAABAAEAQB8AAIA+AAACABAAZGF0YUYDEA....))))</pre>

getVoicemailMessages

Parameters	mailbox => [Required] ID for specific Mailbox (Example: 1001) folder => Name for specific Folder (Example: 'INBOX', values from: getVoicemailFolders) date_from => Start Date for Filtering Voicemail Messages (Example: '2016-01-30') date_to => End Date for Filtering Voicemail Messages (Example: '2016-01-30')
Output	<pre>Array ([status] => success [messages] => Array ([0] => Array ([mailbox] => 1001 [folder] => INBOX [message_num] => 1 [date] => 2016-01-30 [callerid] => 5551234567 [duration] => 00:00:06 [urgent] => yes [listened] => no))))</pre>

getVPRIIs

Parameters	vpri => [Required] ID for specific vpri (Example: 1001)
Output	<pre>Array ([status] => success</pre>

	<pre> [vpri] => Array ([0] => Array ([vpri] => 100112 [name] => main vpri [note] => Important notes [channels] => 1 [monthly_fee] => 0.2 [next_billing] => 2022-08-15 [burst_enabled] => 1 [burst_max_channels] => 2 [burst_percentage_charge] => 1)) </pre>
markListenedVoicemailMessage	
Parameters	<pre> mailbox => [Required] ID for specific Mailbox (Example: 1001) folder => [Required] Name for specific Folder (Example: 'INBOX', values from: getVoicemailFolders) message_num => [Required] ID for specific Voicemail Message (Example: 1) listened => [Required] Code for mark voicemail as listened or not-listened (Values: 'yes'/'no') </pre>
Output	<pre> Array ([status] => success) </pre>
markUrgentVoicemailMessage	
Parameters	<pre> mailbox => [Required] ID for specific Mailbox (Example: 1001) folder => [Required] Name for specific Folder (Example: 'INBOX', values from: getVoicemailFolders) message_num => [Required] ID for specific Voicemail Message (Example: 1) urgent => [Required] Code for mark voicemail as urgent or not-urgent (Values: 'yes'/'no') </pre>
Output	<pre> Array ([status] => success) </pre>
moveFolderVoicemailMessage	
Parameters	<pre> mailbox => [Required] ID for specific Mailbox (Example: 1001) folder => [Required] Name for specific Folder (Example: 'INBOX', values from: getVoicemailFolders) message_num => [Required] ID for specific Voicemail Message (Example: 1) new_folder => [Required] Destination Folder (Example: 'Urgent', values from: getVoicemailFolders) </pre>
Output	<pre> Array ([status] => success) </pre>
sendVoicemailEmail	
Parameters	<pre> mailbox => [Required] ID for specific Mailbox (Example: 1001) folder => [Required] Name for specific Folder (Example: 'INBOX', values from: getVoicemailFolders) message_num => [Required] ID for specific Voicemail Message (Example: 1) email_address=> [Required] Destination Email address (Example: john.doe@my-domain.com) </pre>
Output	<pre> Array ([status] => success) </pre>
setVoicemail	
Parameters	<pre> mailbox => [Required] ID for a specific Mailbox (Example: 1001) name => [Required] Name for the Mailbox password => [Required] Password for the Mailbox skip_password=> [Required] True if Skipping Password (Boolean: 1/0) email => Email address for receiving messages, multiple email addresses are allowed if separated by a comma attach_message => [Required] Yes for Attaching WAV files to Message (Values: 'yes'/'no') delete_message => [Required] Yes for Deleting Messages (Values: 'yes'/'no') say_time => [Required] Yes for Saying Time Stamp (Values: 'yes'/'no') timezone => [Required] Time Zone for Mailbox (Values from getTimeZones) say_callerid => [Required] Yes for Saying the Caller ID (Values: 'yes'/'no') play_instructions => [Required] Code for Play Instructions Setting (Values from getPlayInstructions) language => [Required] Code for Language (Values from getLanguages) client => ID for a specific Reseller Client ID (Example: 54374) transcription => Voicemail Transcription transcription_locale => Transcription locale code (values from getLocales) email_attachment_format => Code for Email Attachment format (Values from getVoicemailAttachmentFormats) unavailable_message_recording => Recording for the Unavailable Message (values from getRecordings) </pre>
Output	<pre> Array ([status] => success) </pre>

The following samples show how to get all Servers Information from our database and how to select a specific Server for your display purposes.

We provide a complete VoIPms Class with all functions on our Example Files

PHP5 - Class VoIPms - Sample Code

```
<?
class VoIPms{
    /*****\
    * VoIPms - API Credentials
    \*****/
    var $api_username   = 'john.doe@mydomain.com';
    var $api_password   = 'johnspassword';

    /*****\
    * VoIPms - SoapClient / SoapCall
    \*****/
    var $soap_client;
    function soapClient(){
        $this->soap_client = new SoapClient(null, array(
            'location'   => "https://voip.ms/api/v1/server.php",
            'uri'         => "urn://voip.ms",
            'soap_version' => SOAP_1_2,
            'trace'       => 1
        ));
    }

    function soapCall($function, $params){
        if(!$this->soap_client){$this->soapClient();}
        try { return $this->soap_client->__soapCall($function, $params);}
        catch (SoapFault $e) { trigger_error("SOAP Fault: [{$e->faultcode}] {$e->faultstring}", E_USER_ERROR); }
    }

    /*****\
    * VoIPms - API Functions
    \*****/

    function getServersInfo($server_pop){
        $function = "getServersInfo";
        $params = array(
            "params" => array(
                "api_username" => $this->api_username,
                "api_password" => $this->api_password,
                "server_pop"   => $server_pop
            )
        );
        return $this->soapCall($function,$params);
    }
}
?>
```

PHP5 - Using Class VoIPms - Sample Code

```
<?
require_once("class.voipms.php");
$voipms = new VoIPms();

echo "<pre>";

echo "
Display Specific Server
=====
";
$response = $voipms->getServersInfo(1);
print_r($response);

echo "
Display All Servers
=====
";
$response = $voipms->getServersInfo();
print_r($response);

echo "</pre>";
?>
```

The following samples show how to get all Servers Information from our database and how to select a specific Server for your display purposes.

Please Note:

- When using our REST/JSON API you need to send the Method to be used and the Required Parameters as part of the URL.
- By default the output Content-Type is "text/html".
- If you want the output Content-Type to be "application/json", add the following to your URL: &content_type=json

PHP - Using cURL GET - Sample Code

```
<?
$ch = curl_init();

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true );

curl_setopt($ch, CURLOPT_URL,
"https://voip.ms/api/v1/rest.php?api_username=john@domain.com&api_password=password&method=getServersInfo&server_pop=1");
```

```

$result1 = curl_exec($ch);

curl_setopt($ch, CURLOPT_URL,
"https://voip.ms/api/v1/rest.php?api_username=john@domain.com&api_password=password&method=getServersInfo");
$result2 = curl_exec($ch);

curl_close($ch);

/* Convert JSON to Array */
$data1=json_decode($result1,true);
$data2=json_decode($result2,true);

echo "<pre>";

echo "
Display Specific Server
=====
";
print_r($data1);

echo "
Display All Servers
=====
";
print_r($data2);

echo "</pre>";
?>

```

PHP - Using cURL POST - Sample Code

```

<?
$postfields = array(
    'api_username'=>'john@domain.com',
    'api_password'=>'password',
    'method'=>'getServersInfo',
    'server_pop'=>'1'
);

$ch = curl_init();
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true );
curl_setopt($ch, CURLOPT_POST, true );
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);
curl_setopt($ch, CURLOPT_URL, "https://voip.ms/api/v1/rest.php");
$result = curl_exec($ch);
curl_close($ch);

$data=json_decode($result,true);

echo "<pre>";
print_r($data);
echo "</pre>";
?>

```



Information and Sales

Mon - Fri: 9:00 - 17:00

USA/CANADA: 1.877.7.VOIP.MS

USA/CANADA (Français):
1.866.914.5294

International: 1.214.615.8599



Live Chat

Mon - Fri: 9:00 - 22:00
Sat - Sun: 9:00 - 17:00

IM with Customer Service



Email

Sales: sales@voip.ms (en/fr/es)

Support: support@voip.ms (en/fr/es)

Jobs: jobs@voip.ms (en/fr/es)



Fax

USA Toll-Free: 1.888.311.7782

International: 1.214.723.7555

