Zoho Corporation

Zoho Voice SDK | Details and Configuration

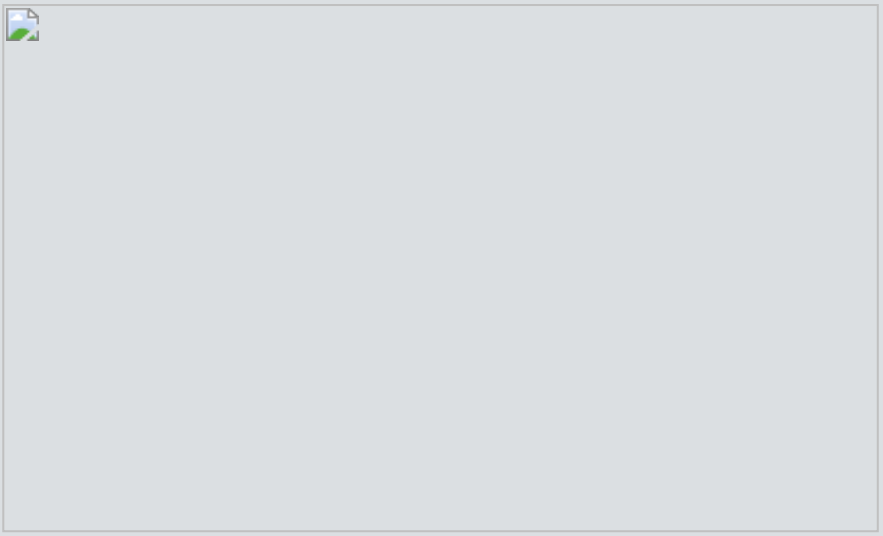
Table of contents

Java SDK

Introduction

Zoho Voice is a cloud-based telephony service using which you can make and receive calls. Zoho Voice also provides SDKs for developers to integrate their app with Zoho Voice.

SDK Details and Configuration	
Download Refresh Token	Step 1 : Login https://voice.zoho.com Step 2 : Please click the "java sdk refresh token" button under the user profile. Step 3 : Store refresh token in DB.

ZohoVoiceSDK Structure	
Source jar for SDK (jar needs to be copied to your lib directory)	ZohoVoiceSDK.jar available in lib/ZohoVoiceSDK.jar
Needed lib for SDK (Dependencies folder needs to be copied to your product zip)	httpcore-4.4.5.jar, json, httpclient-4.5.2.jar, commons-logging-1.1.1.jar available in lib/thirdParty_Jars
java docs for SDK	

Java SDK

SDK Object	
SDK Object Creation (mandatory)	ZohoVoiceSDK zohoVoiceSDK = new ZohoVoiceSDK(DC dc , String refreshToken); (ZohoVoiceSDK object created only in server start) Default DataCenter : DC.US

	Allowed DataCenter : DC.US
Set Proxy	zohoVoiceSDK.setViaProxy(boolean) Default value : false

Zoho Voice API Object

Zoho Voice API Object	
API object creation (mandatory)	ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK); zohoVoiceSDK was created in the above step.

Features of SDK

Agents

Import Agents

Import agents from your app to Zoho Voice.

```
// Create import object
ImportAgents importAgents = new ImportAgents();

// Create agentsinfo object
AgentsInfo agentsInfo = new AgentsInfo();
```

```
  agentsInfo.setName("ZohoVoice");
  agentsInfo.setEmailId("abc@sample.com");
  agentsInfo.setDepartmentName("ZVoice");
  // Using the associated number api allows to associate phone numbers.
JSONArray associatedNumber = new JSONArray(); // Collection of phone numbers
agentsInfo.setAssociatedNumbers(associatedNumber);

// Multiple agentsInfo object
ArrayList<AgentsInfo> agentsInfos = new ArrayList<AgentsInfo>();
agentsInfos.add(agentsInfo);

importAgents.setAgentsInfo(agentsInfos);

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
ImportAgentsResponse response = zohoVoiceAPI.importAgents(importAgents);

// Response Object
String status = response.getStatus();
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
ArrayList<Agents> agents = response.getAgents();
Agents agent = agents.get(0);
agent.getAgentId();
agent.getAgentNumber();
agent.getDepartmentName();
agent.getEmailId();
agent.getExtension();
agent.getName();
agent.getOnlineStatus();
```

Get Login Details

Fetch the login details by inputting agent ID and email address.

```
GetAgentLoginDetail getAgentLoginDetail = new GetAgentLoginDetail();
// Set agentId or emailId .
getAgentLoginDetail.setAgentId("agentId");
getAgentLoginDetail.setEmailId("emailId");

ZohoVoiceAPI zVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
GetAgentLoginDetailResponse response = zVoiceAPI.getAgentLoginDetail(getAgentLoginDetail);

// Response Object
String status = response.getStatus();
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
JSONObject loginDetails = response.getLoginDetail();
//LoginDetails has the credentials to log in to the dialer
```

Get Agents

Input agent ID or search key and get the agent list.

```
GetAgents getAgents = new GetAgents();
getAgents.setAgentId("agentId");
getAgents.setSearchKey("searchKey");
getAgents.setFromIndex(2);
getAgents.setToIndex(10);

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK,);
GetAgentsResponse response =  zohoVoiceAPI.getAgents(getAgents);
```

```
// Response Object
String status = response.getStatus();
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
ArrayList<Agents> agents = response.getAgents();
Agents agent = agents.get(0);
agent.getAgentId();
agent.getAgentNumber();
agent.getDepartmentName();
agent.getEmailId();
agent.getExtension();
agent.getName();
agent.getOnlineStatus();
```

Associated Number

To get details about associated numbers and respective agents.

```
GetAssociatedNumbers getAssociatedNumbers = new GetAssociatedNumbers();
getAssociatedNumbers.setAgentId("agentId");

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
GetAssociatedNumbersResponse response = zohoVoiceAPI.getAssociatedNumbers(getAssociatedNumbe

// Response Object
String status = response.getStatus();
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
```

```
ArrayList<NumberInfo> outgoingNumbers = response.getOutgoingNumbers();
ArrayList<NumberInfo> incomingNumbers = response.getIncomingNumbers();
NumberInfo incomingNumber = incomingNumbers.get(0);
Long numberId = incomingNumber.getNumberId();
String displayName = incomingNumber.getDisplayName();
String telNumber = incomingNumber.getTelNumber();
String countryCode = incomingNumber.getCountryCode();
String didNumber = incomingNumber.getDidNumber();
```

Regenerate Key

To regenerate the agent-specific key. This key and password are used to log in to the Dialer.

```
RegenerateKey regenerateKey = new RegenerateKey();
regenerateKey.setAgentId("AgentId");
regenerateKey.setOldAgentKey("oldAgentPassword");

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
RegenerateKeyResponse response = zohoVoiceAPI.regenerateKey(regenerateKey);

// Response Object
String status = response.getStatus();
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
```

Regenerate Password

To regenerate the agent-specific password.

```
RegeneratePassword regeneratePassword = new RegeneratePassword();
regeneratePassword.setAgentId("AgentId");
```

```
regeneratePassword.setEmailId("abc@sample.com");
regeneratePassword.setOldPassword("oldPassword");

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
RegeneratePasswordResponse response = zohoVoiceAPI.regeneratePassword(regeneratePassword);

// Response Object
String status = response.getStatus();
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
```

Resend Mail

To resend the email that contains the regenerated password.

```
ResendMail resendMail = new ResendMail();
resendMail.setAgentId("AgentId");
resendMail.setEmailId("abc@sample.com");

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
ResendMailResponse  response = zohoVoiceAPI.resendMail(resendMail);

// Response Object
String status = response.getStatus();
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
```

Call Transfer - Agent or Queue List

Shows a list of agents or queue to whom a call needs to be transferred.

```
TransferAgentOrQueue transferAgentOrQueue = new TransferAgentOrQueue();
transferAgentOrQueue.setSearchText("searchText");
transferAgentOrQueue.setFromIndex(2);
transferAgentOrQueue.setToIndex(10);
transferAgentOrQueue.setIncludeQueues(true);

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
TransferAgentOrQueueResponse response = zohoVoiceAPI.transferAgentOrQueue(transferAgentOrQueu

// Response Object
String statusCode = response.getStatusCode();
String errorMessage = response.getErrorMessage();
JSONObject responseInJSON = response.getJSONObject();
ArrayList<Agents> agents = response.getAgents();
Agents agent = agents.get(0);
agent.getAgentId();
agent.getAgentNumber();
agent.getDepartmentName();
agent.getEmailId();
agent.getExtension();
agent.getName();
agent.getOnlineStatus();
```

Logs

Get Call Logs

Fetch the call logs from Zoho Voice.

```
GetCallLogs getCallLogs = new GetCallLogs();
```

```
  getCallLogs.setFromIndex(1);
  getCallLogs.setToIndex(20);
  getCallLogs.setFromDate("2021-09-27");
  getCallLogs.setToDate("2021-09-30");
  getCallLogs.setAllCountry(true);
  getCallLogs.setMinCredits(1.0);
  getCallLogs.setMaxCredits(10.0);
  getCallLogs.setMinDuration(20); // min duration in secs
getCallLogs.setMaxDuration(30);// max duration in secs
// agent id get from Get Agents
ArrayList<String> agnetIds = new ArrayList<String>();
agnetIds.add("615000001010123");
getCallLogs.setAgnetIds(agnetIds);
// call types
ArrayList<String> callTypes = new ArrayList<String>();
callTypes.add("incoming");
callTypes.add("outgoing");
callTypes.add("missed");
callTypes.add("forward");
callTypes.add("bridged");
getCallLogs.setCallType(callTypes);

ZohoVoiceAPI zohoVoiceAPI = new ZohoVoiceAPI(zohoVoiceSDK);
GetLogsResponse getLogsResponse = zVoiceAPI.getCallLogs(getCallLogs);

// Response
String status = getLogsResponse.getStatus();
String statusCode = getLogsResponse.getStatusCode();
String errorMessage = getLogsResponse.getErrorMessage();
JSONObject responseInJSON = getLogsResponse.getJSONObject();
```

```
ArrayList<CallLog> logs = getLogsResponse.getCallLogs();
CallLog callLog = logs.get(0);
String agentName = callLog.getAgentName();
String callDuration = callLog.getCallDuration();
String callerIdName = callLog.getCallerIdName();
String callerIdNumber = callLog.getCallerIdNumber();
String callType = callLog.getCallType();
String destinationNumber = callLog.getDestinationNumber();
String endTime = callLog.getEndTime();
String startTime = callLog.getStartTime();
```

Client SDK

Overview

Zoho Voice WebSDK allows you to make and receive calls directly from any web browser that supports webRTC. Using our SDK you can create applications like Click to Call, Conferencing Apps and even Webphones.

INCLUDING THE JAVASCRIPT

Include the Zoho Voice javascript file as shown below

```
Without Dialpad UI
<script src="https://js.zohostatic.com/zvoice_plugin/latest/js/zohovoice.min.js"></script>

With Dialpad UI
<script src="https://js.zohostatic.com/zvoice_plugin/latest/js/zohovoice-ui.min.js"></script>
```

INITIALIZING THE ZOHO VOICE OBJECT:

The Zoho Voice WebSDK object needs to be initialized.

```
var options = {
        development:false,
        debug:false,
        draggable:true,
```

```
        }] //Zoho Outgoing List
}
var zohovoice = new ZohoVoice(options);

//Authenticate via accounts session
zohovoice.ajaxOpts.domain = 'http://app.zoho.com';  // replace your domain name
zohovoice.ajaxOpts.csrf = {key:'CSRF_KEY', value:'zvtpname'}; //replace key name to your CSRF cookie name



//Authenticate via OAuth Token
zohovoice.ajaxOpts.isOAuth = true;
zohovoice.ajaxOpts.oAuthCallBack = (callBack) => {
    callBack(token)
}

zohovoice.initialize();
```

EVENT REGISTRATION:

Pass function references to the events produced from the SDK. This is where your UI manipulation should handle all the different call flows.

Callback Events:

```
regState
// Triggered when registration state change
on('regState', function(regObject) {})
//regObject response
{
    status:'connecting',    //No I18N
    username:self.opts.username
}
//regState Status(connecting,registered,failed,error,disconnected,websocket closed,unregistered)

callState
// Triggered when call state change
on('callState', function(callerObject) {})
//callerObject response
/*
```

```
{
   callId: "6d2bcd2a-e87a-4518-91dc-52ed3caee986"
   callStatus: "connected"
   didName: "Direct Agent"
   didNumber: "14695016096"
   duration: "00:00"
   id: "14695016095-1625564868881"
   isAccepted: true
   isCallStarted: true
   isConf: false
   isConnecting: true
   isExtension: false
   isInCall: true
   isMissed: false
   isOutgoing: false
   isSecondCall: false
   name: "Unknown"
   number: "14695016095"
   startTime: 1625564868881
}
*/
//callState callStatus(incoming,connecting,ringing,connected,callend)
```

error

```
// Triggered when error occured
on('error', function(errorObject) {})
//errorObject response
/*
{
   code: 1,
   desc:'Error message'
}
*/
```

login

```
// Triggered when login
on('login', function() {})
```

logout

```
// Triggered when logout
on('logout', function() {})
```

```
ready
//Triggered on element created
on('ready', function() {})

ready
//Triggered on element created
on('ready', function() {})

voiceProfile
//Triggered on voiceProfile update
on('voiceProfile', function() {})

//Triggered to update call duration while call is running
on('timer', function(tObj) {})

//Triggered on mediaDeviceChange
on('mediaDeviceChange', function(tObj) {})

//Triggered on microphoneChange
on('microphoneChange', function(tObj) {})

//Triggered on voice sdk destroy
on('destroy', function(tObj) {})

//Triggered to updateCallerInfo
on('updateCallerInfo', function(cInfo) {})

transferStaus
on('transferStaus', function(transferObj){});
//transferObj response
{
   callerNumber: "xxxxxxxx",
   callerName: "xxxxxxxx",
   callId: "6d2bcd2a-e87a-4518-91dc-52ed3caee986",
   callTransfered: true,
   departmentName: "AlarmsOne",
   duration: "00:00",
   emailid: "brindha.r+101@zohotest.com",
   extension: 98750,
   name: "brindha 101 Tech",
   number: "58421713_4501000000516029",
   isTransferCall: true, //on caller C
```

```
    onlineStatus: "Available",
    serviceName: "ZohoVoice",
    transferId: 1625564926802,
    status: {type: "caller", status: "unhold"}
}
//status response
{type:"transfer", status:"dialing"} //on transfer call dialing a agent/queue
{type:"transfer", status:"ringing"} //on transfer call ringing a agent/queue
{type:"transfer", status:"bridged"} //on transfer call agent has disconnected
{type:"transfer", status:"unavailable"} //on transfer call agent is not available
{type:"threeway", status:"bridged"} //call has merged on transfer call
{type:"caller", status:"hold"}  //on transfer caller A as on hold
{type:"caller", status:"unhold"} //on transfer caller A as on unhold
{type:"caller", status:"hangup"} //on transfer caller A has disconnected the call

//Example for listening events
var zohovoice= new ZohoVoice(options);
zohovoice.on('regState', function({
  //Create what you want
});
```

SDK Reference

The following are the configuration parameters:

Attribute	Type	Description	Values
debug	Boolean	If debug is enabled, logs will be shown in the browser's developer console.	true|false
sipDebug	Boolean	If sipDebug is enabled, sip logs will be shown in the browser's developer console.	true|false
development	Boolean	If development is enabled, it will connect to development server else it will connect to production server.	true|false
agentKey	String	The agentkey used as a authentication for make/receive calls.	
name	String	The registered name will be used as the Caller ID name.	

username	String	Zoho Voice login username.	
password	String	Zoho Voice login password.	
element	String	This attribute will be used as a placeholder for Zoho Voice widget.	
defaultCountry	String	This attribute will be used as a default country in dialpad.	
show	Boolean	To show/hide the Zoho Voice widget in the UI.	
autoRegister	Boolean	To register with Zoho Voice server automatically. Default value is "true".	true|false
isDarkMode	Boolean	enable it for dark mode.	true|false
callHeaders	Array	Custom header for SIP Invite.	
turnServers	Array	Custom turn servers. [{ urls: ["stun:us-relay2.zohovoice.com:9090", "turn:us-relay2.zohovoice.com:9090?transport=tcp", "turn:us-relay2.zohovoice.com:9090?transport=udp"], username: 'xxxxxx', credential: 'xxxxxxx' }];	
clientRegion	String	To use MediaServer based on the user's region, to improve the call quality. ["Mumbai", "WDC", "Singapore", "Sydney", "Amsterdam", "Dallas", "Sanjose"];	
outgoingNumb	JSONA	List of numbers purchased by the user to make the outbound calls.	
outgoingNumb	JSONO	Default outgoing number.	
enableAgentSt	Boolean	To show or hide Agent Status by default is false	true|false
numberSugges	Boolean	enable or disable Number Suggestions	true|false

enableCallSche	Boolean	enable or disable to show call schedule	true|false
callNotes	Boolean	enable or disable to show call callnotes	true|false
extensions	Boolean	enable or disable extensions calling	true|false
menuList	Array	custom menu list for logs, settings	['logs','sett
showLauncherI	Boolean	enable or disable Launcher Icon	true|false
launcherIcon	String	launcher icon url	
profileCache	Boolean	enable it to store profile details in localStorage to load dialpad faster.	true/false
widgetStyle	Object	{ pos: { top: 'auto', //No I18N bottom: '10px', //No I18N right: '10px', //No I18N left: 'auto' //No I18N }, width: '100%', //No I18N height: '100%', //No I18N 'max-width': '300px', //No I18N 'max-height': '520px' //No I18N }	

Zoho Voice Methods

These are the methods supported

Name	Parameters	Description
makeCall	var zohovoice = ZohoVoice({ apiKey:"xxxx-xxxx-xxx-xxx" })	

	zohovoice.makeCall({
	number:130000,
	name:'Zoho Voice',
	photo:'URL'
	})
	(or)
	zohovoice.makeCall('123456');	

endCall	//Normal call disconnect
	zohovoice.endCall({callId:'xxxxxx-xxx-xxx'});
	
	//Transfer call agent disconnect
	zohovoice.endCall({transferId:'xxxxxx-xxx-xxx'});	

answerCall	zohovoice.answerCall({callId:'xxxxxx-xxx-xxx'});	
dtmf	zohovoice.dtmf(digit, {callId:'xxxxxx-xxx-xxx'});	
setMute	zohovoice.setMute(true|false, {callId:'xxxxxx-xxx-xxx'});	
setHold	zohovoice.setHold(true|false, {callId:'xxxxxx-xxx-xxx'});	

show	true/false	
unRegisterServer		
registerServer		
showDialer		
destroy		
setAgentStatus	(Available/On Break/On Call/Offline)	

setTurnServerList	var zohovoice = ZohoVoice({
	apiKey:"xxxx-xxxx-xxx-xxx"
	})
	var turnservers = [{
	urls: ["stun:us-relay2.zohovoice.com:9090",
	"turn:us-relay2.zohovoice.com:9090?transport=tcp",
	"turn:us-relay2.zohovoice.com:9090?transport=udp"],
	username: 'xxxxxx',
	credential: 'xxxxxxx'
	}]
	zohovoice.setTurnServerList(turnservers);	

setOutgoingNumber	{
	number:"123456789",
	numberId:"123456789",
	isDefault:true
	}	

setDefaultCountry	zohovoice.setDefaultCountry('us')	
holdAndAccept	zohovoice.holdAndAccept({callId:'xxxxxx-xxx-xxx'})	
endAndAccept	zohovoice.endAndAccept({callId:'xxxxxx-xxx-xxx'})	
switchCall	zohovoice.switchCall({callId:'xxxxxx-xxx-xxx'})	
whisper	zohovoice.whisper({callId:'xxxxxx-xxx-xxx'})	
threeway	zohovoice.threeway({callId:'xxxxxx-xxx-xxx'})	
barge	zohovoice.threeway({callId:'xxxxxx-xxx-xxx'})	
transfer	(transferAgent/Queue), callId zohovoice.transfer({ "departmentName":"US Sales", "number":"xxxxx-xxxxxx", "extension":12345, "onlineStatus":"Available", "name":"Zyleker", "emailid":"zylker@zohotest.com", "serviceName":"ZohoVoice" }, {callId;'xxxxx-xxxxxx'})	Transfer a incoming call to an agent/queue
merge	{callId:'xxxxxx-xxx-xxx-xx'} zohovoice.merge({callId;'xxxxx-xxxxxx'})	(Threeway) Merge the

		transfered call
updateCallerInfo	zohovoice.updateCallerInfo({ "name":"Zyleker", "email":"zylker@zohotest.com", "company":"zylker", "infoUrl":"https://zoho.com", "appName":"zoho_voice" })	To update the caller info in dialpad