Pexip Infinity Version 19
PexRTC JavaScript Client API

Contents

Introduction

This guide describes the PexRTC JavaScript client API. This API is designed for use by web-based custom voice/video applications that want to initiate or connect to conferences hosted on the Pexip Infinity platform.

It allows web developers to delegate to the Pexip platform the job of joining a meeting room etc. and just get back the key components that they need to assign to an HTML element on a web page.

Using the API

The Pexip web client JavaScript API is accessed by an object, "PexRTC", an instance of which provides methods and callback registers for driving the client interface, including initiating WebRTC and RTMP calls to Pexip Virtual Meeting Rooms (VMRs).

The path to the PexRTC object is **https://<node_address>/static/webrtc/js/pexrtc.js** where **<node_address>** is the address of a Conferencing Node.

Summary of API functions

This section summarizes the methods and callbacks that may be used. All functions and their parameters are then subsequently described in more detail.

Methods

Method	Description
Client control functions	
makeCall(node, conference, name, bandwidth, call_type, flash)	Join the specified conference. By default, bring up a WebRTC video call.
connect(pin, extension)	Proceed with connection. Must be called after onSetup callback has given initial information.
muteAudio(setting)	Mute or unmute the local outgoing audio stream.
muteVideo(setting)	Mute or unmute the local outgoing video stream.
sendDTMF(digits, uuid)	Send one or more DTMF digits to the conference.
sendChatMessage(message)	Send a chat message to all chat-enabled conference participants (WebRTC or Skype for Business / Lync).
disconnect()	Disconnect participant.
addCall(call_type, flash)	Escalate existing call to add video/presentation/screensharing. Typically used when currently in a call_type of "none" (roster-only view).
getPresentation()	Request the full framerate presentation stream to be activated.
stopPresentation()	Stops a full-framerate presentation stream if it is running.
present(call_type)	Activate or stop screen capture sharing.
getMediaStatistics()	Retrieve statistics of the media streams.
Conference control functions (all of these functions are only available to users with Host rights)	
dialOut(destination, protocol, role, cb, params)	Dial out from the conference.
setConferenceLock(setting)	Lock or unlock the conference (when locked, new participants cannot join).
setMuteAllGuests(setting)	Set or unset the "mute all Guests" setting on a conference (when set, no Guest participants can speak unless explicitly unmuted).
setParticipantMute(uuid, setting)	Administratively mute a participant.
setParticipantRxPresentation (uuid, setting)	Administratively disable the sending of a presentation to a participant.
setParticipantSpotlight(uuid, setting)	Enable or disable "spotlight" on a participant.
setRole(uuid, setting)	Change the role of another participant.
unlockParticipant(uuid)	Let the specified participant into a locked conference.

Method	Description
transferParticipant(uuid, destination, role, pin)	Transfers a participant to another conference.
startConference()	Starts a conference and allows Guests in the "waiting room" to join the meeting.
disconnectParticipant(uuid)	Disconnect a given participant.
disconnectAll()	Disconnect all participants from the conference.

Callbacks

Callback	Description
onSetup(stream, pin_status, conference_extension)	Initial setup is complete.
onConnect(stream)	The call has connected successfully (after the connect method has been called on the object).
onError(err)	An error has occurred during the call. This is fatal and the call must be considered closed.
onDisconnect(reason)	The call has been disconnected by the server (e.g. if the participant has been administratively disconnected).
onLayoutUpdate(view, participants)	The stage layout has changed.
onPresentation(setting, presenter)	A presentation has started or stopped.
onPresentationReload(url)	A new presentation frame is available in JPEG format.
onRosterList(roster)	This is deprecated in favor of onParticipantCreate/Update/Delete.
onParticipantCreate(participant)	A new participant has been added.
onParticipantUpdate (participant)	A participant has been updated.
onParticipantDelete(participant)	A participant has been deleted.
onChatMessage(message)	A chat message has been broadcast to the conference.
onStageUpdate(stage)	An update to the "stage layout" is available. This declares the order of active speakers, and their voice activity.
onPresentationConnected (stream)	The WebRTC incoming full-framerate presentation stream has been set up successfully.
onPresentationDisconnected (reason)	The WebRTC incoming presentation stream has been stopped.
onScreenshareConnected (stream)	The outgoing screenshare has been set up correctly.
onScreenshareStopped(reason)	The WebRTC screensharing presentation stream has been stopped. The floor may have been taken by another presenter, or the user stopped the screenshare, or some other error occurred.
onScreenshareMissing()	Called when attempting to present, but the Pexip screensharing extension cannot be found, which is required for Chrome screensharing.

Variables and fields

- A few additional configuration changes can be undertaken via instance variables on the PexRTC object, before calling makeCall.
- A set of fields on the PexRTC object can be probed after **onSetup**, and provide useful information about the connection.

Client control functions

This section describes in detail the methods that may be used to initiate and manage a connection to a Conferencing Node.

makeCall(node, conference, name, bandwidth, call_type, flash)

Join the specified conference. By default, bring up a WebRTC video call.

Parameters:

node	Conferencing Node.
conference	Name of conference to join.
name	Display name of the participant.
bandwidth	Maximum bandwidth (in kbps) for up and down stream (can be null).
call_type	Optional (default is to bring up a WebRTC video call): • "presentation": receive-presentation-only WebRTC call • "screen": share-screen-only WebRTC call (Chrome only, using Pexip screensharing extension) • "audioonly": audio-only WebRTC call • "recvonly": receive-only WebRTC call • "rtmp": an RTMP video call • "stream": an RTMP stream • "none": do not initiate a video call, just join for conference control and events ("roster-only")
flash	Optional "PexVideo" Flash object to which the RTMP URL can be directly attached.

Return value: none.

(Successful callback will be onSetup.)

connect(pin, extension)

Proceed with connection. Must be called after onSetup callback has given initial information.

This function applies the PIN if a PIN is required (if the PIN is incorrect, **onSetup** will be called again), and initiates a remote video connection if requested by the **call_type**. If no PIN is required, PIN should be set to undefined.

If calling into a Pexip Virtual Reception, this may also be used as part of the two-stage dialing process to specify the extension to connect to. If the **onSetup** callback has specified that an extension is required, then onConnect must be called with the desired extension. This will trigger another **onSetup** call (if the extension is valid).

Parameters:

pin	User-supplied PIN (if required, otherwise null).
extension	Conference to connect to, when being used with a Virtual Reception (otherwise leave undefined).

Return value: none.

(Successful callback will be onConnect, unless PIN is incorrect, or extension is specified, in which cases onSetup will be called again.)

muteAudio(setting)

Mute or unmute the local outgoing audio stream.

Parameters:

setting	true = mute; false = unmute.

Return value: new setting.

muteVideo(setting)

Mute or unmute the local outgoing video stream.

Parameters:

setting	true = mute; false = unmute.

Return value: new setting.

sendDTMF(digits, uuid)

Send one or more DTMF digits to the conference.

Note that this does not send DTMF to all participants; it can either be used in a gateway call or be sent to a specific participant identified by the UUID (as seen in the participant list). This is typically used to enter codes in a remote audio or video IVR.

Parameters:

digits	String of DTMF digits.
uuid	UUID of the target participant (from the participant list). Leave undefined for a gateway call.

Return value: none.

sendChatMessage(message)

Send a chat message to all chat-enabled conference participants (WebRTC or Skype for Business / Lync).

Parameters:

message	Text message to send.

Return value: none.

disconnect()

Disconnect participant.

Parameters: none.

Return value: none; will return when signaling is complete.

addCall(call_type, flash)

Escalate existing call to add video/presentation/screensharing. Typically used when currently in a call_type of "none" (roster-only view).

Parameters:

call_type	As for makeCall, typically this will be null to add audio/video capability.
flash	As for makeCall, an optional Flash object to which to attach an RTMP stream.

Return value: none.

(Successful callback will be onConnect.)

getPresentation()

Request the full framerate presentation stream to be activated.

Although this method can be used at any time, it only makes sense to do this after onPresentation callback has said that a presentation is available.

Parameters: none.

Return value: none.

See onPresentationConnected callback, or onPresentationDisconnected if an error.

Note that there are two ways of getting presentation: full-framerate video (this method) and JPEG images via the onPresentationReload callback.

stopPresentation()

Stops a full-framerate presentation stream if it is running.

Parameters: none.

Return value: none.

present(call_type)

Activate or stop screen capture sharing. This is only supported by Chrome, and if the Pexip screensharing extension is installed.

Parameters:

call_type	Media source; currently only "screen" is supported, or null to stop screen sharing.

Return value: none.

getMediaStatistics()

Retrieve statistics of the media streams. This is only supported by Chrome, and the statistics are acquired directly from Chrome.

Parameters: none.

Return value:

```
Object {
  outgoing: Object {
    audio: Object { bitrate/packets-lost/packets-received/percentage-lost }
    video: Object { decode-delay/bitrate/packets-lost/packets-received/percentage-
lost/resolution }
  }, incoming: Object {
    audio: Object { bitrate/packets-lost/packets-received/percentage-lost }
    video: Object { decode-delay/bitrate/packets-lost/packets-received/percentage-
lost/resolution }
  }
}
```

Conference control functions

This section describes in detail the methods that may be used to manage an existing conference.

dialOut(destination, protocol, role, cb, params)

Dial out from the conference. Only available to users with "chair" (Host) rights.

Parameters:

destination	Address to dial.
protocol	The protocol to use to place the outgoing call: • "sip" • "h323" • "rtmp" • "mssip" (for calls to Microsoft Skype for Business / Lync) • "auto" (to use Call Routing Rules)
role	The level of privileges the participant has in the conference: • "HOST": the participant has Host privileges • "GUEST": the participant has Guest privileges Default is "HOST" if unspecified.
cb	A callback to call when the dial-out request has been processed. This will return an object containing "result", which is an array of uuids of the new participant if dial-out was successfully initiated.

params	This is an object containing any of the following optional parameters:	
	presentation_uri	This additional parameter can be specified for RTMP calls to send the presentation stream to a separate RTMP destination.
	streaming	Identifies the dialed participant as a streaming or recording device: • true: streaming/recording participant • false: not a streaming/recording participant Default: false
	dtmf_sequence	An optional DTMF sequence to be transmitted after the call to the dialed participant starts.
	call_type	Limits the media content of the call: • "video": main video plus presentation • "video-only": main video only • "audio": audio-only Default: "video"
	keep_conference_alive	Determines whether the conference continues when all other non-ADP participants have disconnected: • "keep_conference_alive": the conference continues to run until this participant disconnects (applies to Hosts only). • "keep_conference_alive_if_multiple": the conference continues to run as long as there are two or more "keep_conference_alive_if_multiple" participants and at least one of them is a Host. • "keep_conference_alive_never": the conference terminates automatically if this is the only remaining participant. Default: "keep_conference_alive" for non-streaming participants, and "keep_conference_alive_never" for streaming participants.
	remote_display_name	An optional friendly name for this participant. This may be used instead of the participant's alias in participant lists and as a text overlay in some layout configurations.
	overlay_text	Optional text to use instead of **remote_display_name** as the participant name overlay text.
	For example: `{presentation_uri: "rtmp://foo/bar", streaming: "yes", dtmf_sequence: "1234"}`	

Return value: if the **cb** parameter is not specified, the call will block and return an object containing "result". This is an array of UUIDs of new participants, if dial-out was successfully initiated. In most cases the dial-out will only generate a single call and thus a single UUID in this array, however if Pexip Infinity forks the call there may end up being multiple UUIDs. Only one of these will be answered, however, and the rest will be disconnected.

The call UUIDs will appear in the participant list immediately, with a **service_type** of "connecting". The participant list can then be monitored for this participant's join — if the call is answered the participant will typically update with a **service_type** of "conference". The call will time out in 30 seconds if not answered and the participants will be disconnected.

setConferenceLock(setting)

Lock or unlock the conference (when locked, new participants cannot join). Only available to users with "chair" (Host) rights.

Parameters:

setting	true = locked; false = unlocked.

Return value: none.

setMuteAllGuests(setting)

Set or unset the "mute all Guests" setting on a conference (when set, no Guest participants can speak unless explicitly unmuted). Only available to users with "chair" (Host) rights.

Parameters:

setting	true = all Guests muted; false = all Guests unmuted.

Return value: none.

setParticipantMute(uuid, setting)

Administratively mute a participant. Only available to users with "chair" (Host) rights.

Parameters:

uuid	UUID of the participant (from the participant list).
setting	true = muted; false = unmuted.

Return value: none.

Changes in state are reflected in participant list updates (onParticipantUpdate callback).

setParticipantRxPresentation(uuid, setting)

Administratively disable the sending of a presentation to a participant. Only available to users with "chair" (Host) rights.

Parameters:

uuid	UUID of the participant (from the participant list).
setting	true = will receive presentation; false = will not receive presentation.

Return value: none.

Changes in state are reflected in participant list updates (onParticipantUpdate callback).

setParticipantSpotlight(uuid, setting)

Enable or disable "spotlight" on a participant. Only available to users with "chair" (Host) rights.

The spotlight feature locks any spotlighted participants in the primary positions in the stage layout. When any participants have been spotlighted, the first one to be spotlighted has the main speaker position, the second one has the second position (leftmost small video, for example), and so on. All remaining participants are arranged by most recent voice activity, as is default.

Parameters:

uuid	UUID of the participant (from the participant list).
setting	true = set spotlight on participant; false = remove spotlight from participant.

Return value: none.

Changes in state are reflected in participant list updates (onParticipantUpdate callback).

setRole(uuid, setting)

Change the role of another participant. Only available to users with "chair" (Host) rights.

Parameters:

uuid	UUID of the participant (from the participant list).
setting	"chair" = Host participant; "guest" = Guest participant

Return value: none.

Changes in state are reflected in participant list updates (onParticipantUpdate callback).

unlockParticipant(uuid)

Let the specified participant into a locked conference. Only available to users with "chair" (Host) rights.

Parameters:

uuid	UUID of the participant (from the participant list).

Return value: none.

Changes in state are reflected in participant list updates (onParticipantUpdate callback).

transferParticipant(uuid, destination, role, pin)

Transfers a participant to another conference.

The target conference is identified by the alias in "destination", and they will have the specified "role". If the target is PIN-protected, the PIN for the target role must be specified in the "pin" field. Only available to users with "chair" (Host) rights.

Parameters:

uuid	UUID of the participant (from the participant list).
destination	Target conference alias.
role	Either "guest" or "chair" (Host). Default is "chair" if unspecified.
pin	PIN code for the specified role at the specified conference, if required.

Return value: none.

startConference()

Starts a conference and allows Guests in the "waiting room" to join the meeting.

If the only user with Host rights is connected to the conference without media (as a control-only participant), Guests will remain in the "Waiting for Host" screen. This command starts the conference and any Guests in the "waiting room" will join the meeting. Only available to users with "chair" (Host) rights.

Parameters: none.

Return value: none.

disconnectParticipant(uuid)

Disconnect a given participant. Only available to users with "chair" (Host) rights.

Parameters:

uuid	UUID of the participant (from the participant list).

Return value: none.

Removal of participant is reflected in participant list updates (onParticipantUpdate callback).

disconnectAll()

Disconnect all participants from the conference. The calling participant will also be disconnected. Only available to users with "chair" (Host) rights.

Parameters: none.

Return value: none.

Callbacks

All callbacks are functions written by the API user, and are set as instance variables of the object, for example:

```
function rtc_onconnect(url) { … }
rtc = new PexRTC();
rtc.onConnect = rtc_onconnect;
```

onSetup(stream, pin_status, conference_extension)

Initial setup is complete.

Parameters:

stream	A MediaStream or URL (depending on the browser version) of the local media stream that can be applied to a <video> element. May be null for receive-only or roster-only call types.
pin_status	One of the following: • "none": no PIN required • "required": PIN is required • "optional": PIN is optional (conference Hosts require PIN, Guests can enter with a PIN of "")
conference_extension	Present only if this is a call to a Pexip Virtual Reception, where a target extension needs to be specified in the call to **connect**. This value, if present, is either "standard" for a standard Virtual Reception, or "mssip" if the Virtual Reception is a gateway to a Skype for Business / Lync conference.

Users of this API should call connect to continue connecting, with a PIN and/or **conference_extension** if required.

Note that this can be called more than once; e.g. first for initial API setup (with no **stream**, but with a **pin_status**) and later with a **stream** after the local media stream has been acquired, if requested.

onConnect(stream)

The call has connected successfully (after the connect method has been called on the object).

Parameters:

stream	A MediaStream or URL (depending on the browser version) of the remote media stream that can be applied to a <video> element. May be null if roster-only or screensharing-only.

Note that this will be called more than once: first for initial roster-only API setup (with no **stream**) and later with a **stream** after video has been acquired, if requested.

onError(err)

An error has occurred during the call. This is fatal and the call must be considered closed. Possible causes include:

- If before onConnect, there has been an error in getting access to the user's camera/microphone.
- If during a call, the connection to server is broken or liveness check fails.

Parameters:

err	A description of the error.

onDisconnect(reason)

The call has been disconnected by the server (e.g. if the participant has been administratively disconnected).

Parameters:

reason	An explanation for the disconnection.

onLayoutUpdate(view, participants)

The stage layout has changed.

Parameters:

view	The layout currently seen by the participant, including: • "1:0": main speaker only • "1:7": main speaker and up to 7 previous speakers • "1:21": main speaker and up to 21 previous speakers • "2:21": two main speakers and up to 21 previous speakers
participants	An array of UUIDs for the participants, in order, starting from the main speaker position.

For example:

```
{view: "1:7", participants: ["a0196175-b462-48a1-b95c-f322c3af57c1", "65b4af2f-657a-4081-98a8-b17667628ce3”]}
```

onPresentation(setting, presenter)

A presentation has started or stopped.

Parameters:

setting	true = presentation has started; false = presentation has stopped.
presenter	The name of the presenter (only given when **setting** = true, else null).

You can receive **onPresentation(true)** after **onPresentation(true)** without first receiving **onPresentation(false)**. This will occur, for example, if the presenter has changed.

Users can use getPresentation to get a full-framerate video stream, or listen for onPresentatonReload to fetch JPEG frames.

onPresentationReload(url)

A new presentation frame is available in JPEG format.

Parameters:

url	The URL of the new presentation frame.

onRosterList(roster)

This is deprecated in favor of onParticipantCreate/Update/Delete.

An update to the participant list is available.

Parameters:

roster	The new participant list. This is an array of objects per participant. Each participant includes the following fields:

call_direction	Either "in" or "out" as to whether this is an inbound or outbound call.
display_name	The display name of the participant.
encryption	"On" or "Off" as to whether this participant is connected via encrypted media.
external_node_uuid	The UUID of an external node e.g. a Skype for Business / Lync meeting associated with an external participant. This allows grouping of external participants as the UUID will be the same for all participants associated with that external node.
has_media	Boolean indicating whether the user has media capabilities.
is_audio_only_call	Set to "YES" if the call is audio only.
is_external	Boolean indicating if it is an external participant, e.g. coming in from a Skype for Business / Lync meeting.
is_muted	Set to "YES" if the participant is administratively muted.
is_presenting	Set to "YES" if the participant is the current presenter.
is_streaming_conference	Boolean indicating whether this is a streaming/recording participant.
is_video_call	Set to "YES" if the call has video capability.
local_alias	The calling or "from" alias. This is the alias that the recipient would use to return the call.
overlay_text	Text that may be used as an alternative to **display_name** as the participant name overlay text.
protocol	The protocol with which the participant is connecting.
role	Either "chair" (Host) or "guest".
rx_presentation_policy	Set to "ALLOW" if the participant is administratively allowed to receive presentation, or "DENY" if disallowed.
service_type	The service type: • "connecting": for a dial-out participant that has not been answered • "waiting_room": if waiting to be allowed to join a locked conference • "ivr": if on the PIN entry screen • "conference": if in the VMR • "gateway": if it is a gateway call • "test_call": if it is a Test Call Service
spotlight	A Unix timestamp of when this participant was spotlighted, if spotlight is used.
start_time	A Unix timestamp of when this participant joined (UTC).
uuid	The UUID of this participant, to use with other operations.
uri	The URI of the participant.
vendor	The vendor identifier of the browser/endpoint with which the participant is connecting.

onParticipantCreate(participant)

A new participant has been added.

Parameters:

participant	The new participant object, as for onRosterList.

onParticipantUpdate(participant)

A participant has been updated.

Parameters:

participant	The updated participant object, as for onRosterList.

onParticipantDelete(participant)

A participant has been deleted.

Parameters:

participant	The participant being deleted. Object with only one field: • uuid: the UUID of this participant.

onChatMessage(message)

A chat message has been broadcast to the conference.

Parameters:

message	This is an object containing the following fields:	
	origin	Name of the sending participant.
	uuid	UUID of the sending participant.
	type	MIME content-type of the message, usually text/plain.
	payload	Message contents.

onStageUpdate(stage)

An update to the "stage layout" is available. This declares the order of active speakers, and their voice activity.

Parameters:

stage	This is an array of objects per active participant. Each participant has the following fields:	
	participant_uuid	The UUID of the participant.
	stage_index	The index of the participant on the "stage". 0 is most recent speaker, 1 is the next most recent etc.
	vad	Audio speaking indication. 0 = not speaking, 100 = speaking.

onPresentationConnected(stream)

The WebRTC incoming full-framerate presentation stream has been set up successfully.

Parameters:

stream	A MediaStream or URL (depending on the browser version) of the incoming presentation media stream that can be applied to a <video> element.

onPresentationDisconnected(reason)

The WebRTC incoming presentation stream has been stopped. Note that this does not occur when someone else starts presenting; rather, it occurs on errors and call disconnect.

Parameters:

reason	An explanation for the disconnection.

onScreenshareConnected(stream)

The outgoing screenshare has been set up correctly.

Parameters:

stream	A MediaStream or URL (depending on browser version) representing the outgoing presentation stream.

onScreenshareStopped(reason)

The WebRTC screensharing presentation stream has been stopped. The floor may have been taken by another presenter, or the user stopped the screenshare, or some other error occurred.

Parameters:

reason	An explanation for the disconnection.

onScreenshareMissing()

Called when attempting to present, but the Pexip screensharing extension cannot be found, which is required for Chrome screensharing.

Parameters: none.

Instance variables

A few additional configuration changes can be undertaken via instance variables on the PexRTC object, before calling makeCall:

Instance variable	Description
audio_source / video_source	Can be set to: • null: default sources • false: do not request • a uuid of a media source gathered through device enumeration (Chrome only)
recv_audio / recv_video	Booleans to specify whether to request receiving audio or video. Defaults: true
bandwidth_in / bandwidth_out	Bandwidth settings, to allow different bandwidths for incoming and outgoing directions (kbps). Defaults: 1280; overridden by makeCall if **makeCall** specifies a bandwidth.
png_presentation	Boolean to specify whether to provide presentation URLs as PNGs rather than JPG images (PNG is higher quality than JPG but uses more bandwidth). Default: false
screenshare_fps	Sets the framerate in fps for the presentation stream. Default: 5
default_stun	The default STUN server to use, to be added to those presented by the Pexip Conferencing Node. Default: none
turn_server	A TURN server to use; specified in the form of a JavaScript option as used in a **PeerConnection**, e.g. {'url': 'turn:10.0.0.1', 'username': 'user', 'credential': 'password' } Default: none
screenshare_api	If writing a custom Chrome screenshare extension, this is the prefix to use. Contact your Pexip authorized support representative for more information. Default: pexGetScreen

Fields

The following fields on the PexRTC object are immutable but can be probed after onSetup, and provide useful information about the connection:

Field	Description
role	"HOST" or "GUEST", depending on which role the participant holds in the conference.
version	The version of the Pexip server being communicated with, e.g. "10 (25010.0.0)".
chat_enabled	Whether the chat functionality is administratively enabled or disabled on the Pexip system.
service_type	Either "conference", "gateway" or “test_call” depending on whether this is a VMR, gateway or Test Call Service respectively.
current_service_type	Your current service_type, i.e. "conference" / "waiting_room" / "gateway" / etc, as given in roster list updates.

Changelog

Changes in version 19:

None.

Changes in version 18:

The **participant** parameter in **onParticipantCreate** and **onParticipantUpdate** callbacks has a new **external_node_uuid** field.

Changes in version 17:

None.

Changes in version 16:

Support for Safari with WebRTC capabilities.

Changes in version 15:

None.

Changes in version 14:

- New service_type of “test_call”.
- The **dialOut** command has a new **overlay_text** parameter.

Changes in version 13:

- New instance variables:
  - **png_presentation**: boolean to specify whether to provide presentation URLs as PNGs rather than JPG images (default: false)
  - **screenshare_fps**: sets the framerate for the presentation stream (defaults to 5 fps)

Changes in version 12:

- **startConference** is a new method that allows you to start a conference and allow Guests in the "waiting room" into the meeting.
- **transferParticipant** is a new method that allows you to transfer a participant to another conference.
- **dialOut** has two new parameters: **keep_conference_alive** and **remote_display_name**; the **protocol** parameter can be "auto".
- Supports Microsoft Edge and ORTC when used with adapter.js (http://webrtc.github.io/adapter/adapter-latest.js).
- Supports 1080p resolutions if enabled on Pexip Infinity.

Changes in version 11:

- **setRole** is a new method that allows you to change the role of a participant.
- **dialOut** can now indicate if it is a streaming participant, send DTMF to the dialed participant, and limit the call capability.
- **onLayoutUpdate** callback has been added.

Changes in version 10:

- **dialOut** now returns an array of call UUIDs.
- **dialOut** now permits dialing out a separate presentation stream for RTMP calls.

- **onSetup/connect** exchange now adds support for Virtual Receptions.
- **sendDTMF** can now send DTMF to a specific participant.

Changes in version 9.1:

- **dialOut** is now non-blocking if the "cb" parameter is used.

More information

Questions about this API and requests for demonstration files should be directed to your Pexip authorized support representative.