Pexip Infinity

PexRTC JavaScript Client API

Software Version 31

Document Version 31.a

March 2023

]pexip[

Contents

Introduction 2
Using the API 3
Summary of API functions 4
Client control functions 7
Conference control functions 11
Callbacks 20
Instance variables 27
Fields 29
Changelog 30
More information 31

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.

© 2023 Pexip AS

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)	Join the specified conference. By default, bring up a WebRTC video call.
connect(pin, extension, idp)	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.
sendChatMessage(message, uuid)	Send a chat message to chat-enabled conference participants (WebRTC or Skype for Business / Lync).
sendApplicationMessage(obj, uuid)	Send a JavaScript object to other participants.
disconnect()	Disconnect participant.
disconnectcall()	Disconnect the A/V call in use, leaving the control-only participant connected.
addCall(call_type)	Escalate existing call to add video/presentation/screensharing. Typically used when currently in a call_type of "none" (roster-only view).
renegotiate(resend_sdp)	Apply changes in selected camera/microphone (via audio_source and video_source properties) to the call.
getPresentation()	Request the full-frame rate presentation stream.
present(call_type)	Activate or stop screen capture sharing.
getMediaStatistics()	Retrieve statistics of the media streams.
getSecureCheckCode()	Obtain the secure check code of a direct media call.
Conference control functions (unless stated otherwise, 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/unmute a participant's audio.
videoMuted(uuid)	Administratively mute a participant's video. Hosts can mute anybody, a Guest can only mute themself.
videoUnmuted(uuid)	Administratively unmute a participant's video. Hosts can unmute anybody, a Guest can only unmute themself.

Method	Description
setParticipantRxPresentation(uuid, setting)	Administratively disable the sending of a presentation to a participant.
setParticipantSpotlight(uuid, setting)	Enable or disable "spotlight" on a participant.
setParticipantText(uuid, text)	Change the participant name overlay text.
setPresentationInMix(state, uuid)	Controls whether or not the participant sees presentation in the layout mix (Adaptive Composition layout only).
setRole(uuid, setting)	Change the role of another participant.
unlockParticipant(uuid)	Let the specified participant into a locked conference.
sendDTMF(digits, uuid)	Send one or more DTMF digits to the conference.
sendFECC(action, axis, direction, target, timeout)	Send a Far End Camera Control message to a remote participant.
setBuzz()	Raise the local participant's hand.
clearBuzz(uuid)	Lower the previously raised hand.
clearAllBuzz()	Lower all previously raised hands. Only available to Hosts.
transformLayout(transforms)	Changes the conference layout, controls streaming content, and enables/disables indicators and overlay text.
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, idp_selection)	Initial setup is complete.
onAuth(redirect_url, idp_uuid, idp_name)	An Identity Provider has been selected.
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).
onConferenceUpdate(properties)	The conference properties have been updated.
onLayoutUpdate(view, participants, requested_layout)	The stage layout has changed.

Callback	Description
onPresentation(setting, presenter, uuid, presenter_source)	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.
onDirectMessage(message)	A direct, private chat message has been sent to the participant.
onApplicationMessage(message)	A message has been sent 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-frame rate 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.
onCallTransfer(alias)	Informs the client that the call has been transferred to a new conference with the given alias.
onFECC(signal)	Indicates a far-end camera control signal has been received by the client.
onSplashScreen(properties)	The client should display a splash screen (direct media calls only).

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)

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

Parameters:

node	string	Conferencing Node.
conference	string	Alias of conference to join.
name	string	Display name of the participant.
bandwidth	number	Maximum bandwidth (in kbps) for up and down stream (can be null).
call_type	string	Optional (default is to bring up a WebRTC video call): • "presentation": receive-presentation-only WebRTC call • "screen": share-screen-only WebRTC call • "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")

Return value: none.

(Successful callback will be onSetup.)

connect(pin, extension, idp)

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 SSO participant authentication is enabled the UUID of an Identity Provider is required.

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	string	User-supplied PIN (if required, otherwise null).
extension	string	Conference to connect to, when being used with a Virtual Reception (otherwise leave undefined).
idp	string	UUID of an Identity Provider if using SSO.

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	boolean	true = mute; false = unmute.

Return value: new setting.

muteVideo(setting)

Mute or unmute the local outgoing video stream.

Parameters:

setting	boolean	true = mute; false = unmute.

Return value: new setting.

sendChatMessage(message, uuid)

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

Parameters:

message	string	Text message to send.
uuid	string	The UUID of the target participant; leave undefined for a message to all participants.

Return value: none.

sendApplicationMessage(obj, uuid)

Send a JavaScript object to other participants.

Parameters:

obj	string	An Object which will be converted to JSON and sent to the target participant(s).
uuid	string	The UUID of the target participant; leave undefined for a message to all participants.

Return value: none.

disconnect()

Disconnect participant.

Parameters: none.

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

disconnectcall()

Disconnect the A/V call in use, leaving the control-only participant connected.

Parameters: none.

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

addCall(call_type)

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

Parameters:

call_type	string	As for makeCall, typically this will be null to add audio/video capability.

Return value: none.

(Successful callback will be onConnect.)

renegotiate(resend_sdp)

Apply changes in selected camera/microphone (via audio_source and video_source properties) to the call.

Parameters:

resend_sdp	boolean	Use false or unspecified to change media devices. Use true to renegotiate at the protocol level, for example for changing bandwidth, or adding/removing media streams.

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

getPresentation()

Request the full-frame rate presentation stream.

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-frame rate video (this method) and JPEG images via the onPresentationReload callback.

present(call_type)

Activate or stop screen capture sharing.

Parameters:

call_type	string	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 }
  }
}
```

getSecureCheckCode()

Obtain the secure check code of a direct media call. Matching values with both parties verifies that the media session is end-to-end encrypted.

Parameters: none.

Return value: string.

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	string	Address to dial.
protocol	string	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) To successfully place calls via the 'auto' protocol option, suitable Call Routing Rules must be configured. To enable calls to be placed via the other protocols you must select **Enable legacy dialout API** (via **Platform > Global Settings > Connectivity**).
role	string	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	string	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	object	This is an object containing any of the following optional parameters:

presentation_uri	string	This additional parameter can be specified for RTMP calls to send the presentation stream to a separate RTMP destination.
streaming	boolean	Identifies the dialed participant as a streaming or recording device: • true: streaming/recording participant • false: not a streaming/recording participant Default: false
dtmf_sequence	string	An optional DTMF sequence to transmit after the call to the dialed participant starts.
call_type	string	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	string	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	string	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	string	Optional text to use instead of **remote_display_name** as the participant name overlay text.

For example:

```
{presentation_uri: "rtmp://foo/bar", streaming: true, 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	boolean	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	boolean	true = all Guests muted; false = all Guests unmuted.

Return value: none.

setParticipantMute(uuid, setting)

Administratively mute/unmute a participant's audio. Only available to users with "chair" (Host) rights.

Parameters:

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

Return value: none.

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

videoMuted(uuid)

Administratively mute a participant's video. Hosts can mute anybody, a Guest can only mute themself.

Parameters:

uuid	string	Optional UUID of the participant (from the participant list). If specified it affects the given participant. If not specified, it adjusts yourself.

Return value: none.

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

videoUnmuted(uuid)

Administratively unmute a participant's video. Hosts can unmute anybody, a Guest can only unmute themself.

Parameters:

uuid	string	Optional UUID of the participant (from the participant list). If specified it affects the given participant. If not specified, it adjusts yourself.

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	string	UUID of the participant (from the participant list).
setting	boolean	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, ahead of any current speakers. 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 usual.

Parameters:

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

Return value: none.

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

setParticipantText(uuid, text)

Change the participant name overlay text. Only available to users with "chair" (Host) rights.

The text is only applied if overlay text is enabled on a VMR. It can also change the text of an audio-only participant.

Parameters:

uuid	string	UUID of the participant (from the participant list).
text	string	Text to use as the participant name overlay text.

Return value: none.

setPresentationInMix(state, uuid)

Controls whether or not the participant sees presentation in the layout mix (Adaptive Composition layout only).

Parameters:

state	boolean	true = enable presentation in mix; false = disable presentation in mix
uuid	string	UUID of the participant (from the participant list).

Return value: none.

setRole(uuid, setting)

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

Parameters:

uuid	string	UUID of the participant (from the participant list).
setting	string	"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	string	UUID of the participant (from the participant list).

Return value: none.

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

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

Return value: none.

sendFECC(action, axis, direction, target, timeout)

Send a Far End Camera Control message to a remote participant.

Note that this does not send FECC to all participants; it can either be used in a gateway call or be sent to a specific participant identified by the target UUID (as seen in the participant list).

Parameters:

action	string	Either "start", "stop", or "continue".
axis	string	Either "pan", "tilt", or "zoom".
direction	string	Use "left", "right" for pan; "up", "down" for tilt; or "in", "out" for zoom.
target	string	UUID of the target participant (from the participant list). Leave undefined for a gateway call.
timeout	number	The duration for which to send the signal. Recommended values are 1000 (1 second) for initial "start" message; 200 for "continue" messages.

Return value: none.

setBuzz()

Raise the local participant's hand.

Parameters: none.

Return value: none.

clearBuzz(uuid)

Lower the previously raised hand.

Parameters:

uuid	string	If specified, lower this participant's hand (only available to Hosts); if unspecified, lower the local participant's hand.

Return value: none.

clearAllBuzz()

Lower all previously raised hands. Only available to Hosts.

Parameters: none.

Return value: none.

transformLayout(transforms)

Changes the conference layout, controls streaming content, and enables/disables indicators and overlay text.

Parameters:

transforms — This is an object containing any of the following optional parameters:

layout host_layout guest_layout	string	In VMRs the layout for Hosts and Guests is controlled by the **layout** parameter. In Virtual Auditoriums the Host layout is controlled by the **host_layout** parameter and the Guest layout is controlled by the **guest_layout** parameter. The layout options are: • "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": 2 main speakers and up to 21 previous speakers • "1:33": 1 small main speaker and up to 33 other speakers • "4:0": 2x2 layout, up to a maximum of 4 speakers • "9:0": 3x3 layout, up to a maximum of 9 speakers • "16:0": 4x4 layout, up to a maximum of 16 speakers • "25:0": 5x5 layout, up to a maximum of 25 speakers • "ac": Adaptive Composition layout • "teams": Teams-like layout (this is a technology preview feature and is only recommended for use with Microsoft Teams gateway calls) Note that the **layout** parameter is an alias for **host_layout**, and that an attempt to set **guest_layout** in a service that is not a Virtual Auditorium will return a "400 Bad Request" error.
enable_extended_ac *	boolean	This enables an extended Adaptive Composition (AC) layout that can contain more video participants than the standard AC layout. In the standard AC layout, a maximum of 12 video participants are shown across up to three rows (2 participants on the first row / 3 on the second row / 7 on the bottom row). In the extended layout up to 23 video participants may be shown, initially across three rows (2/3/7 extending to 2/5/7 and then 3/5/7) and then across four rows (3/5/7/8) when required. Note that this setting only has an effect in a conference that is already using AC, so the conference either needs to be already configured to use AC, or you also need to pass `"layout": "ac"` or `"host_layout": "ac"` to enable AC simultaneously, for example: `{"transforms": {"layout": "ac","enable_extended_ac": true}}` * Technology preview only
streaming_indicator	boolean	Determines whether the streaming indicator icon is disabled (**false**) or enabled (**true**).
recording_indicator	boolean	Determines whether the recording indicator icon is disabled (**false**) or enabled (**true**).
transcribing_indicator	boolean	Determines whether the transcribing indicator icon is disabled (**false**) or enabled (**true**).
enable_active_speaker_indication	boolean	Determines whether active speaker indication is disabled (**false**) or enabled (**true**).
enable_overlay_text	boolean	Determines whether participant names overlay text is disabled (**false**) or enabled (**true**).
plus_n_pip_enabled	boolean	Controls whether the "plus n indicator" i.e. the number of participants present in the conference, is displayed (**true**) or not (**false**).
streaming	object	This can be used to specifically control the content sent to a streaming participant, for example to send a different layout to the stream from that which is seen by standard participants. It is an object containing any of the following optional parameters:

layout	string	Sets the layout seen by the streaming participant (regardless of Host or Guest role). ℹ You cannot use this layout parameter if the main stream is using Adaptive Composition

Return value: none.

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	string	UUID of the participant (from the participant list).
destination	string	Target conference alias.
role	string	Either "guest" or "chair" (Host). Default is "chair" if unspecified.
pin	string	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 presentation and 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	string	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, idp_selection)

Initial setup is complete.

Parameters:

stream	string	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	string	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	string	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: • "standard": for a regular, Microsoft Teams or Google Meet Virtual Reception. • "mssip": for a Lync / Skype for Business Virtual Reception.
idp_selection	object	An optional array of the available Identity Providers (IDPs) for the service, consisting of {uuid: string, name: string, img: string}, for example: `"idp_selection": [ { "name": "Azure AD”, "img": "", "uuid": "4fe19459-67b4-4e84-abb4-a3765a0a7e09” }, { "name": "ADFS”, "img": "", "uuid": "7e43610f-1c82-4726-899a-af801748846c” } ]`

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

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.

onAuth(redirect_url, idp_uuid, idp_name)

An Identity Provider has been selected.

Parameters:

redirect_url	string	The SAML SSO URL including the AuthNRequest message.
idp_uuid	string	The UUID of the selected Identity Provider (IDP). This is important for retries as a new AuthNRequest message should be generated and the IDP UUID is needed to avoid prompting the user again (i.e. so there can just be a retry button without needing to go through IDP selection again).
idp_name	string	The name of the selected Identity Provider.

onConnect(stream)

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

Parameters:

stream	string	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	string	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	string	An explanation for the disconnection.

onConferenceUpdate(properties)

The conference properties have been updated.

Parameters:

properties	This is an object containing the following fields. All values are true/false.		
	guests_muted	boolean	Specifies if all Guests are muted.
	locked	boolean	Specifies if the conference is locked.
	started	boolean	Specifies if the conference has been started.
	direct_media	boolean	Specifies if the conference is using direct media.

onLayoutUpdate(view, participants, requested_layout)

The stage layout has changed.

Parameters:

view	string	The layout currently seen by the participant: • "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": 2 main speakers and up to 21 previous speakers • "1:33": 1 small main speaker and up to 33 other speakers • "4:0": 2x2 layout, up to a maximum of 4 speakers • "9:0": 3x3 layout, up to a maximum of 9 speakers • "16:0": 4x4 layout, up to a maximum of 16 speakers • "25:0": 5x5 layout, up to a maximum of 25 speakers • "5:7": Adaptive Composition (AC) layout • "ac_presentation_in_mix": AC and viewing a presentation in the layout mix (single person presenter) • "ac_presentation_in_mix_group": AC and viewing a presentation in the layout mix (group presenter) • "teams": Teams-like layout (this is a technology preview feature and is only recommended for use with Microsoft Teams gateway calls)
participants	array	An array of UUIDs for the participants, in order, starting from the main speaker position. Note that an API-only participant (no audio or video) always receives an empty participants UUID list.
requested_layout	object	Indicates the last requested layout. For example, when using a Virtual Auditorium as a Host, the view is "2:21" whereas the requested layout can be {"primary_screen": {"chair_layout": "2:21", "guest_layout": "4:0"}}.

For example:

```
{view: "1:7", participants: ["a0196175-b462-48a1-b95c-f322c3af57c1", "65b4af2f-657a-4081-98a8-b17667628ce3”],
requested_layout: {"primary_screen": {"chair_layout": "2:21", "guest_layout": "4:0"}}}
```

onPresentation(setting, presenter, uuid, presenter_source)

A presentation has started or stopped.

Parameters:

setting	boolean	true = presentation has started; false = presentation has stopped.
presenter	string	The name of the presenter (only given when **setting** = true, else null).
uuid	string	The UUID of the presenter.
presenter_source	string	The presentation source ("video" or "static").

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-frame rate video stream, or listen for onPresentatonReload to fetch JPEG frames.

onPresentationReload(url)

A new presentation frame is available in JPEG format.

Parameters:

url	string	The URL of the new presentation frame.

onRosterList(roster)

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

onParticipantCreate(participant)

A new participant has been added.

Parameters:

participant	object	The new participant object, as for onRosterList.

onParticipantUpdate(participant)

A participant has been updated.

Parameters:

participant	object	The updated participant object, as for onRosterList.

onParticipantDelete(participant)

A participant has been deleted.

Parameters:

participant	object	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	string	Name of the sending participant.
	uuid	string	UUID of the sending participant.
	payload	string	Message contents.

onDirectMessage(message)

A direct, private chat message has been sent to the participant.

Parameters:

message	This is an object containing the following fields:		
	origin	string	Name of the sending participant.
	uuid	string	UUID of the sending participant.
	payload	string	Message contents.

onApplicationMessage(message)

A message has been sent to the conference.

Parameters:

message	This is an object containing the following fields:		
	origin	string	Name of the sending participant.
	uuid	string	UUID of the sending participant.
	direct	boolean	Indicates if this message was sent direct to the participant (true) or broadcast to the conference (false).
	payload	string	JSON-encoded object.

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	string	The UUID of the participant.
	stage_index	number	The index of the participant on the "stage". 0 is most recent speaker, 1 is the next most recent etc.
	vad	number	Audio speaking indication. 0 = not speaking, 100 = speaking.

onPresentationConnected(stream)

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

Parameters:

stream	string	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	string	An explanation for the disconnection.

onScreenshareConnected(stream)

The outgoing screenshare has been set up correctly.

Parameters:

stream	string	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	string	An explanation for the disconnection.

onCallTransfer(alias)

Informs the client that the call has been transferred to a new conference with the given alias.

Parameters:

alias	string	The alias of the new conference.

onFECC(signal)

Indicates a far-end camera control signal has been received by the client.

Parameters:

signal — This is an object in the form `{'action': action, 'movement': [{'axis': axis, 'direction': direction}], 'timeout': timeout};`

It contains the following fields.

action	string	Either "start", "stop", or "continue".
movement	object	An array of movements, consisting of:
timeout	number	The signal duration e.g. 1000 (1 second).

movement:

axis	string	Either "pan", "tilt", or "zoom".
direction	string	Use "left", "right" for pan; "up", "down" for tilt; or "in", "out" for zoom.

onSplashScreen(properties)

The client should display a splash screen (direct media calls only).

For VMRs with direct media enabled, it is the clients' responsibility to render local screens. The **screen_key** uniquely identifies the type of screen the client should display. On receiving the event with no data, the splash screen should be cleared.

Parameters:

screen_key	string	The **screen_key** uniquely identifies the type of screen the client should display. The screen keys may be **direct_media_welcome**, **direct_media_waiting_for_host**, **direct_media_other_participants_audio_only**, **direct_media_escalate**, or **direct_media_deescalate**.
text	string	Suggested text to display.
background	string	Suggested background URL to display.

Instance variables

A few additional configuration changes can be undertaken via instance variables on the PexRTC object, before calling makeCall. Where indicated, some of the instance variables can be changed mid-call by setting and then using renegotiate.

Instance variable		Description	Changeable in-call
audio_source / video_source	string	Can be set to: • null: default sources • false: do not request • a uuid of a media source gathered through device enumeration (Chrome only) The use of **audio_source** / **video_source** and **user_media_stream** are mutually exclusive; you cannot change which style you are using during a call.	✓
autoGainControl	boolean	Indicates if autoGainControl should be enabled for audio. This only applies if **user_media_stream** is not in use. Default: true	
recv_audio / recv_video	boolean	Booleans to specify whether to request receiving audio or video. Defaults: true	✓
bandwidth_in / bandwidth_out	number	Bandwidth settings, to allow different bandwidths for incoming and outgoing directions (kbps). Defaults: 1280; overridden by makeCall if **makeCall** specifies a bandwidth.	✓
call_tag	string	An optional call tag to assign to a participant.	
default_stun	string	The default STUN server to use, to be added to those presented by the Pexip Conferencing Node. Default: none	
echoCancellation	boolean	Indicates if echoCancellation should be enabled for audio. This only applies if **user_media_stream** is not in use. Default: true	
fecc_supported	boolean	Indicates if the client supports far-end camera control signals. Default: false	✓
h264_enabled	boolean	Indicates if the H.264 codec is enabled. Default: true	
noiseSuppression	boolean	Indicates if noiseSuppression should be enabled for audio. This only applies if **user_media_stream** is not in use. Default: true	
screenshare_fps	number	Sets the frame rate in fps for the presentation stream. Default: 5	
turn_server	object	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	

Instance variable		Description	Changeable in-call
user_media_stream	object	A MediaStream object to use instead of PexRTC calling getUserMedia. The use of **audio_source** / **video_source** and **user_media_stream** are mutually exclusive; you cannot change which style you are using during a call.	✓
user_presentation_stream	object	A MediaStream object to use for presentation instead of PexRTC calling getDisplayMedia.	✓
vp8_enabled	boolean	Indicates if the VP8 codec is enabled. Default: true	
vp9_enabled	boolean	Indicates if the VP9 codec is enabled. Default: true	

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
chat_enabled	boolean	Whether the chat functionality is administratively enabled or disabled on the Pexip system.
current_service_type	string	Your current service_type, i.e. "conference" / "waiting_room" / "gateway" / etc, as given in roster list updates.
role	string	"HOST" or "GUEST", depending on which role the participant holds in the conference.
service_type	string	Either "conference", "gateway" or “test_call” depending on whether this is a VMR, gateway or Test Call Service respectively.
uuid	string	The participant UUID of yourself in the conference.
version	string	The version of the Pexip server being communicated with, e.g. "20 (45400.0.0)".

Changelog

Changes in version 31:

- New **getSecureCheckCode** client control function.
- New **sendApplicationMessage** client control function.
- New **onSplashScreen**, **onDirectMessage** and **onApplicationMessage** callbacks.
- New **uuid** parameter in **sendChatMessage** client control function.
- New **direct_media** parameter in **onConferenceUpdate** callback.
- There are 3 new instance variables: **h264_enabled**, **vp8_enabled** and **vp9_enabled**.

Changes in version 30:

- New **presenter_source** parameter in **onPresentation** callback.
- New **plus_n_pip_enabled** argument to the **transformLayout** function.
- New **requested_layout** parameter in **onLayoutUpdate** callback.

Changes in version 29:

- Removed the **stopPresentation** client control function.
- New **idp_selection** parameter in the **onSetup** callback.
- New **onAuth** callback.
- New **idp** argument to the **connect** function.

Changes in version 28:

The flash object parameters have been removed from the **makeCall** and **addCall** functions.

Changes in version 27:

- New conference control function: **setPresentationInMix**.
- New callback: **onFECC**.
- New instance variables: **fecc_supported**, **autoGainControl**, **echoCancellation**, **noiseSuppression**.

© 2023 Pexip AS

More information

For more information about using this API, contact your Pexip authorized support representative.