

Ongoing conference session handling



browser

Dev Server

APIGW

SBEEnabler for webRTC

User is registered and does have a **session ID**

1. Accept/Answer an media conference

HTTP PUT /mediaConferences/{confId}(sdp,header:conference-answer)

Accept the conference (session params)

"**session-open**" - sent to conferenceer over the **event channel** , if present

Accepted (state:accepted)

HTTP 204 No Content (state:accepted)

1. User is registered and does have a **session ID**
2. state of conference is "**session-open**".

2. Initiate audio video session modifications

HTTP PUT /mediaConferences/{confId}(sdp,header:initiate-conference-mod)

Modify Session (session params)

modified SDP - sent to the peer over the **event channel** , if present

(state:mod-sent)

HTTP 204 No Content (state:mod-sent,modID)

1. User is registered and does have a **session ID**
2. state of conference is "**session-open**".

3. Initiate audio video conference HOLD

HTTP PUT /mediaConferences/{confId}(sdp->a:recvonly,header:initiate-conference-hold)

Modify Session (session params)

modified SDP[a:recvonly] - sent to the peer over the **event channel** , if present

(state:mod-sent)

HTTP 204 No Content (state:mod-sent,modID)

1. User is registered and does have a **session ID**
2. state of conference is "**session-open**".

4. Initiate audio video conference RESUME

HTTP PUT /mediaConferences/{confId}(sdp->a:sendrecv,header:initiate-conference-resume)

Modify Session (session params)

modified SDP[a:sendrecv] - sent to the peer over the **event channel** , if present

(state:mod-sent)

HTTP 204 No Content (state:mod-sent,modID)

1. User is registered and does have a **session ID**
2. state of conference is "**session-open**".

5. Accept conference modification

HTTP PUT /mediaConferences/{confId}(modID,sdp,header:accept-conference-mod)

Modify Session (session params)

modified SDP,modID - sent to the peer over the **event channel** , if present

(state:mod-accepted)

HTTP 204 No Content (state:mod-accepted)

1. User is registered and does have a **session ID**
2. state of conference is "**session-open**".

6. Reject/Cancel conference modification

HTTP PUT /mediaConferences/{confId}(modID,sdp,header:rejectCancel-conference-mod)

Modify Session (session params)

Reject modID - sent to the peer over the **event channel** , if present

(Modification Rejected or Deleted)

HTTP 204 No Content

1. User is registered and does have a **session ID**
2. state of call is "**session-open**".

7. Move media conference session to different end point

STEP 1 --> PUT CALL ON HOLD

HTTP PUT /mediaConferences/{confId}(sdp->a:recvonly,header:initiate-call-hold)

Modify Session (session params)

modified SDP[a:recvonly] - sent to the peer over the **event channel** , if present

(state:mod-sent)

HTTP 204 No Content (state:mod-sent,modID)

1. Sessions are active on another endpoint to get media conference move invitation.
2. Termination of media conference session should happen in a specified duration (less than 5 sec).

STEP 2 --> TERMINATE MEDIA CONFERENCE SESSION

HTTP DELETE /mediaConferences/{confId}

Delete media conference Session (session params)

(state:mod-sent)

HTTP 200 OK

1. Other Endpoints having active session registered with same username/VTN/ICMN starts ringing.
2. Conference is moved to the end point that answers the call first.