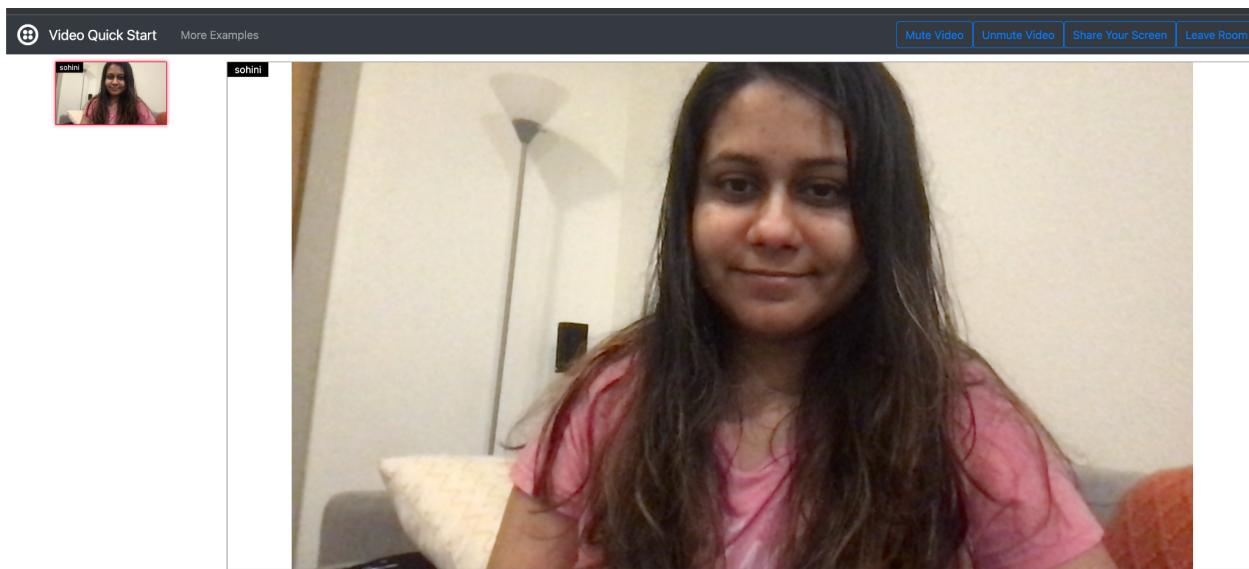


Sohini Video Wheels Off

Quickstart

- Set up the [JavaScript Video Quickstart](#) on your laptop.
- Start by [forking the Quickstart repo](#) into your own GH repo on the PUBLIC GH <https://github.com/> DO NOT USE code.hq.twilio. Then [clone the GH repo](#) to your laptop.
- Once you launch localhost:3000 click on More Examples in the Upper Left-hand Corner. Open the Javascript Console.



- Take Note of the Room SID - **VideoTestRoom - RM129ae172551b5a40675544d43291a69e**
- Create a screenshot of every "Example" you explore in the Quickstart, be sure to smile when on camera! Complete all Examples with the Javascript Console open and toggle all the options and dropdowns, connect, disconnect, etc. as the example guides you. Be sure to capture screenshots of the request inspector reporting the actions from the Quickstart room.

Bandwidth Constraints

Bandwidth Constraints

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the given bandwidth constraints.
 * @param {string} token - Token for joining the Room
 * @param {string} roomName - Room name
 * @param {number} maxAudioBitrate - Max audio bitrate (bps)
 * @param {number} maxVideoBitrate - Max video bitrate (bps)
 * @returns {Promise<Room>}
 */
function connectWithBandwidthConstraints(token, roomName, maxAudioBitrate) {
  return Video.connect(token, {
    maxAudioBitrate: maxAudioBitrate,
    maxVideoBitrate: maxVideoBitrate,
    name: roomName
  });
}

/**
 * Update the bandwidth constraints of a Room.
 * @param {Room} room - The Room whose bandwidth constraints have to be updated
 * @param {number} maxAudioBitrate - Max audio bitrate (bps)
 * @param {number} maxVideoBitrate - Max video bitrate (bps)
 */
function updateBandwidthConstraints(room, maxAudioBitrate, maxVideoBitrate) {
  room.localParticipant.setParameters({
    maxAudioBitrate: maxAudioBitrate,
    maxVideoBitrate: maxVideoBitrate
  });
}

function connectWithBandwidthConstraints = connectWithBandwidthConstraints;
exports.updateBandwidthConstraints = updateBandwidthConstraints;
```

Participant's Media



Local Video Snapshot

Local Video Snapshot

```
'use strict';

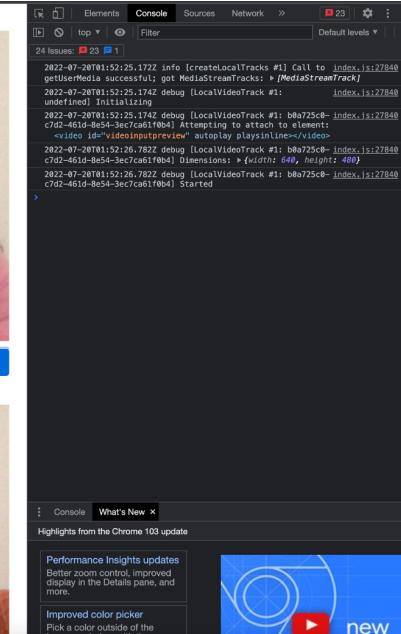
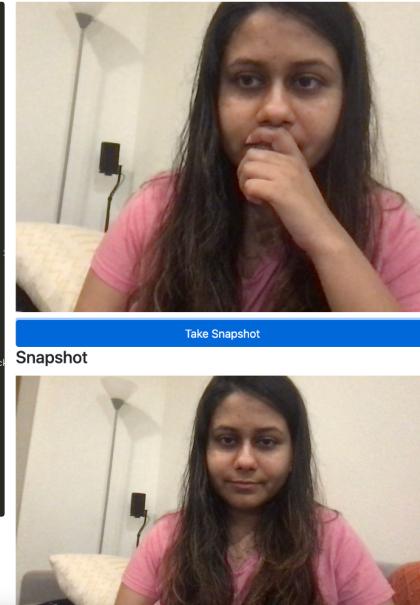
var Video = require('twilio-video');

/**
 * Display local video in the given HTMLVideoElement.
 * @param {HTMLVideoElement} video
 * @returns {Promise<void>}
 */
function displayLocalVideo(video) {
  return Video.createLocalVideoTrack().then(function(localTrack) {
    localTrack.attach(video);
    return localTrack;
  });
}

/**
 * Take snapshot of the local video from the HTMLVideoElement and render it in the HTMLCanvasElement.
 * @param {HTMLVideoElement} video
 * @param {LocalVideoTrack} localVideoTrack
 * @param {HTMLCanvasElement|HTMLImageElement} snapshot
 */
function takeLocalVideoSnapshot(video, localVideoTrack, snapshot) {
  if (window.ImageCapture) {
    const imageCapture = new ImageCapture(localVideoTrack.mediaStreamTrack);
    imageCapture.takePhoto().then(function(blob) {
      snapshot.src = URL.createObjectURL(blob);
    });
  } else {
    snapshot.getContext('2d').drawImage(video, 0, 0);
  }
}

module.exports.displayLocalVideo = displayLocalVideo;
module.exports.takeLocalVideoSnapshot = takeLocalVideoSnapshot;
```

Local Video



Codec Preferences

Codec Preferences

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the given codec preferences
 * @param {string} token - Token for joining the Room
 * @param {string} roomName - Room name
 * @param {Array<AudioCodecName>} preferredAudioCodecs - Preferred audio
 * @param {Array<VideoCodecName>} preferredVideoCodecs - Preferred video
 * @returns {CancelablePromise<Room>}
 */
function connectWithPreferredCodecs(token, roomName, preferredAudioCodecs, preferredVideoCodecs) {
    return Video.connect(token, {
        preferredAudioCodecs: preferredAudioCodecs,
        preferredVideoCodecs: preferredVideoCodecs,
        name: roomName
    });
}

exports.connectWithPreferredCodecs = connectWithPreferredCodecs;
```

Participant's Media



Select Codec Preferences

Preferred Audio Codec: Opus

Selected Audio Codec: opus

Preferred Video Codec: VP8

Selected Video Codec: VP8

Disconnect from Room

```
24 issues: 0 D 1
0bal.vss.twilio.com/signaling Incoming: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:19.860Z debug [LocalVideoTrack #2: 74a999c4] index.js:128201
0b14-47e6-9ed6-895fe878751d] Dimensions: > width: 640, height: 480
2022-07-20T01:53:19.860Z debug [LocalVideoTrack #2: 74a999c4] index.js:128201
0b14-47e6-9ed6-895fe878751d Started
2022-07-20T01:53:19.952Z debug [RemoteVideoTrack #3: index.js:128201]
MT422188d881568919ada7947506] Dimensions: > width: 320, height: 480
2022-07-20T01:53:19.952Z debug [RemoteVideoTrack #3: index.js:128201]
0b14-47e6-9ed6-895fe8787506 Started
2022-07-20T01:53:21.412Z info [connect #1] event index.js:128201
» (elapsedTime: 1440ms, group: "quality", level: "info", name: "stats-report", timestamp: 1658282091412, -)
2022-07-20T01:53:22.491Z debug [connect #1] event index.js:128201
» (elapsedTime: 1480ms, group: "quality", level: "info", name: "active-ice-can-detect-par", timestamp: 1658282091413, -)
2022-07-20T01:53:22.491Z debug [TwilioConnection #2: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Incoming: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:23.332Z debug [TwilioConnection #1: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:23.342Z debug [TwilioConnection #2: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:23.412Z debug [TwilioConnection #1: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:23.492Z debug [TwilioConnection #2: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Incoming: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:27.418Z debug [TwilioConnection #1: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Incoming: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:28.132Z debug [TwilioConnection #1: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:28.142Z debug [TwilioConnection #2: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:28.152Z debug [TwilioConnection #1: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:28.162Z debug [TwilioConnection #2: wss://128.199.10.100:443] index.js:128201
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 128201}
2022-07-20T01:53:28.172Z info [connect #2] event index.js:128201
» (elapsedTime: 1864ms, group: "quality", level: "info", name: "stats-report", timestamp: 1658282090545, -)
2022-07-20T01:53:28.582Z info [connect #2] event index.js:128201
» (elapsedTime: 1864ms, group: "quality", level: "info", name: "active-ice-can-detect-par", timestamp: 1658282090545, -)
3
```

Network Quality

Network Quality

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the Network Quality API enabled.
 * This API is available only in Small Group or Group Rooms.
 * @param {string} token - Token for joining the Room
 * @param {number} localVerbosity - Verbosity level of Network Quality report for the LocalParticipant (0 - 3)
 * @param {number} remoteVerbosity - Verbosity level of Network Quality report for the RemoteParticipant (0 - 3)
 * @returns {CancelablePromise<Room>}
 */
function connectToRoomWithNetworkQuality(token, localVerbosity, remoteVerbosity) {
    return Video.connect(token, {
        networkQuality: {
            local: localVerbosity,
            remote: remoteVerbosity
        }
    });
}

/**
 * Listen to changes in the Network Quality report of a Participant and update your application.
 * @param {Participant} participant - The Participant whose updates you want to receive.
 * @param {function} updateNetworkQualityReport - Updates the app UI with the Network Quality report of the Participant.
 * @returns {void}
 */
function setupNetworkQualityUpdatesForParticipant(participant, updateNetworkQualityReport) {
    participant.on('networkQualityLevelChanged', function() {
        updateNetworkQualityReport(participant);
    });
}

/**
 * Listen to changes in the Network Quality reports and update your application.
 * @param {Room} room - The Room you just joined
 * @param {function} updateNetworkQualityReport - Updates the app UI with the Network Quality reports of the Room.
 */
function setupNetworkQualityUpdatesForRoom(room, updateNetworkQualityReport) {
    room.on('networkQualityLevelChanged', function() {
        updateNetworkQualityReport(room);
    });
}
```

Set Network Quality Verbosity Levels

LocalParticipant: 2

RemoteParticipant(s): 1

Leave Room

Join Room:

RM549554ad56551d8290c3e83c8f8a88cd6

After creating the Room with the desired verbosity levels, please join the above Room using the [QuickStart App](#) or connect the below Participants to the Room. You can also change the verbosity levels and observe the change in the verbosity of the reported Network Quality stats.

Charlie	Disconnect
Alice	Disconnect
Bob	Disconnect
Mak	Disconnect

NQ Level (You):



```
1 issue: 0 D 1
0bal.vss.twilio.com/signaling Incoming: {"type": "heartbeat", "index": 20118}
2022-07-20T01:55:58.816Z debug [TwilioConnection #3: wss://q1] index.js:20118
0b14-47e6-9ed6-895fe878751d] Dimensions: > width: 640, height: 480
2022-07-20T01:55:59.392Z debug [TwilioConnection #4: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Incoming: {"type": "heartbeat", "index": 20118}
2022-07-20T01:55:59.392Z debug [TwilioConnection #3: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:23.412Z info [connect #1] event index.js:20118
» (elapsedTime: 1480ms, group: "quality", level: "info", name: "active-ice-can-detect-par", timestamp: 1658282091413, -)
2022-07-20T01:53:23.492Z debug [TwilioConnection #2: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:23.492Z debug [TwilioConnection #1: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:27.418Z debug [TwilioConnection #1: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Incoming: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:28.132Z debug [TwilioConnection #2: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:28.142Z debug [TwilioConnection #1: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:28.152Z debug [TwilioConnection #2: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:28.162Z debug [TwilioConnection #1: wss://q1] index.js:20118
0bal.vss.twilio.com/signaling Outgoing: {"type": "heartbeat", "index": 20118}
2022-07-20T01:53:28.172Z info [connect #2] event index.js:20118
» (elapsedTime: 1864ms, group: "quality", level: "info", name: "stats-report", timestamp: 1658282090545, -)
2022-07-20T01:53:28.582Z info [connect #2] event index.js:20118
» (elapsedTime: 1864ms, group: "quality", level: "info", name: "active-ice-can-detect-par", timestamp: 1658282090545, -)
3
```

Dominant Speaker

```
/*
 * Connect to a Room with the Dominant Speaker API enabled.
 * This API is available only in Small Group or Group Rooms.
 * @param {string} token - Token for joining the Room
 * @returns {CancelablePromise<Room>}
 */
function connectToRoomWithDominantSpeaker(token) {
  return Video.connect(token, {
    dominantSpeaker: true
  });
}

/**
 * Listen to changes in the dominant speaker and update your application.
 * @param {Room} room - The Room you just joined
 * @param {function} updateDominantSpeaker - Updates the app UI with the new
 * @returns {void}
 */
function setupDominantSpeakerUpdates(room, updateDominantSpeaker) {
  room.on('dominantSpeakerChanged', function(participant) {
    console.log(`A new RemoteParticipant is now the dominant speaker:`);
    updateDominantSpeaker(participant);
  });
}

exports.connectToRoomWithDominantsSpeaker = connectToRoomWithDominantSpeaker;
exports.setupDominantSpeakerUpdates = setupDominantSpeakerUpdates;
```

The screenshot shows a video conference interface with four participants. Each participant has a small video thumbnail and a name label below it. Alice's thumbnail is surrounded by a red border, signifying her as the current dominant speaker. The other participants are Charlie, Bob, and Mak.

```
NO Status: -----
{
  "level": 5,
  "audio": 1,
  "video": 1,
  "sendstats": null,
  "recvstats": null
},
"video": {
  "send": 5,
  "recv": 5,
  "sendstats": null
}
```

The screenshot shows the Network tab of the browser developer tools. It lists several requests and responses from the 'twilio.vss.twilio.com/signaling' domain, all related to heartbeats and connection management. The requests are mostly 'GET' and 'POST' methods with URLs like '/ws/v1/rooms/...'. The responses show JSON payloads containing connection IDs and heartbeat status.

```
/*
 * @param {function} updateNetworkQualityReport - Updates the app UI with
 * a Network Quality report of a Participant.
 * @returns {void}
 */
function setupNetworkQualityUpdates(room, updateNetworkQualityReport) {
  // Listen to changes in Network Quality level of the LocalParticipant.
  setupNetworkQualityUpdatesForParticipant(room.localParticipant, updateNetworkQualityReport);
  // Listen to changes in Network Quality levels of RemoteParticipants
  // in the Room.
  room.participants.forEach(function(participant) {
    setupNetworkQualityUpdatesForParticipant(participant, updateNetworkQualityReport);
  });
  // Listen to changes in Network Quality levels of RemoteParticipants
  // that join the Room in the future.
  room.on('participantConnected', function(participant) {
    setupNetworkQualityUpdatesForParticipant(participant, updateNetworkQualityReport);
  });
}

/**
 * Change the local and remote Network Quality verbosity levels after join.
 * @param {Room} room - The Room you just joined
 * @param {number} localVerbosity - Verbosity level of Network Quality report
 * for the LocalParticipant [1 - 3]
 * @param {number} remoteVerbosity - Verbosity level of Network Quality report
 * for the RemoteParticipant(s) [0 - 3]
 * @returns {void}
 */
function setNetworkQualityConfiguration(room, localVerbosity, remoteVerbosity) {
  room.localParticipant.setNetworkQualityConfiguration({
    local: localVerbosity,
    remote: remoteVerbosity
  });
}

exports.connectToRoomWithNetworkQuality = connectToRoomWithNetworkQuality;
exports.setupNetworkQualityUpdates = setupNetworkQualityUpdates;
exports.setNetworkQualityConfiguration = setNetworkQualityConfiguration;
```

The screenshot shows a video conference interface with four participants. Alice is highlighted with a red border, indicating she is the dominant speaker. The other participants are Charlie, Bob, and Mak.

```
NO Status: -----
null
```

The screenshot shows the Network tab of the browser developer tools. It lists several requests and responses from the 'twilio.vss.twilio.com/signaling' domain, all related to heartbeats and connection management. The requests are mostly 'GET' and 'POST' methods with URLs like '/ws/v1/rooms/...'. The responses show JSON payloads containing connection IDs and heartbeat status.

Video Track Manual Controls

Video Track Manual Controls

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with 'manual' mode. The default mode is 'auto'.
 * @param {string} token - AccessToken for joining the Room
 * @returns {Room}
 */
function joinRoom(token) {
  return Video.connect(token, {
    name: 'my-cool-room',
    bandwidthProfile: {
      video: {
        contentPreferencesMode: 'manual',
        clientTrackSwitchOnControl: 'manual'
      }
    }
  });
}

/**
 * Switch on the RemoteVideoTrack.
 * @param {RemoteVideoTrack} track - The RemoteVideoTrack you want to switch on
 * @returns {RemoteVideoTrack}
 */
function switchOn(track) {
  return track.switchOn();
}

/**
 * Switch off the RemoteVideoTrack.
 * @param {RemoteVideoTrack} track - The RemoteVideoTrack you want to switch off
 * @returns {RemoteVideoTrack}
 */
function switchOff(track) {
  return track.switchOff();
}
```

Remote Video Controls

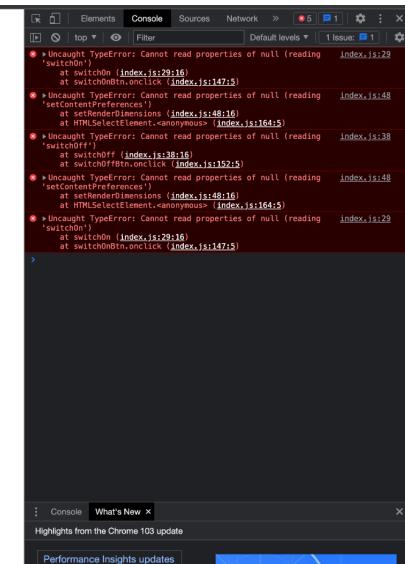
Switch Off Control:

Content Preferences:

Remote Video Track



Video Bitrate



Data Tracks

Data Tracks

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to the given Room with a LocalDataTrack.
 * @param {string} token - AccessToken for joining the Room
 * @returns {CancelablePromise<Room>}
 */
async function connectToRoomWithDataTrack(token, roomName) {
  const localDataTrack = new Video.LocalDataTrack({
    name: 'chat',
  });

  const room = await Video.connect(token, {
    name: roomName,
    tracks: [localDataTrack],
  });

  return room;
}

/**
 * Send a chat message using the given LocalDataTrack.
 * @param {LocalDataTrack} dataTrack - The {@link LocalDataTrack} to send
 * @param {string} message - The message to be sent
 */
function sendChatMessage(dataTrack, message) {
  dataTrack.send(message);
}

/**
 * Receive chat messages from RemoteParticipants in the given Room.
 * @param {Room} room - The Room you are currently in
 * @param {Function} onMessageReceived - Updates UI when a message is received
 */
function receiveChatMessages(room, onMessageReceived) {
  room.participants.forEach(function(participant) {
    participant.dataTracks.forEach(function(publication) {
      if (publication.isSubscribed && publication.trackName === 'chat') {
        publication.on('message', function(msg) {
          onMessageReceived(msg, participant);
        });
      }
    });
  });
}
```

P1

P2: has connected
P2: has disconnected

Hello

Submit

P2

Hi

Submit

Remote Participant Reconnections States and Events

```

    * @returns {void}
  */
function handleRemoteParticipantReconnectionUpdates(room, updateParticipantState, participant) {
  room.on('participantReconnecting', function(participant) {
    updateParticipantState(participant.state);
  });

  room.on('participantReconnected', function(participant) {
    updateParticipantState(participant.state);
  });
}

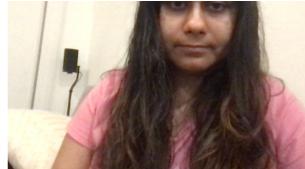
/**
 * Listen to LocalParticipant reconnection events and update the UI accordingly
 * @param {Room} room - The Room you have joined
 * @param {function} updateRoomState - Updates the app UI with the new state
 * @returns {void}
 */
function handleLocalParticipantReconnectionUpdates(room, updateParticipantState) {
  const localParticipant = room.localParticipant;

  localParticipant.on('reconnecting', function() {
    updateParticipantState(localParticipant.state);
  });

  localParticipant.on('reconnected', function() {
    updateParticipantState(localParticipant.state);
  });
}

exports.handleLocalParticipantReconnectionUpdates = handleLocalParticipantReconnectionUpdates;
exports.handleRemoteParticipantReconnectionUpdates = handleRemoteParticipantReconnectionUpdates;

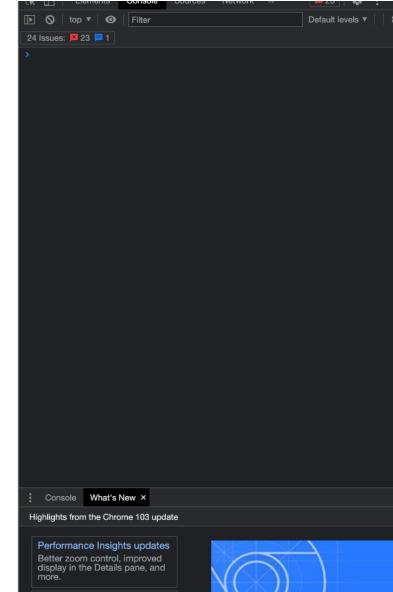
```



Simulate Reconnection

State
Connected
Reconnecting

Remote View



Remote Participant Reconnection States

```

'use strict';

/**
 * Listen to RemoteParticipant reconnection events and update the UI accordingly
 * @param {Room} room - The Room you have joined
 * @param {function} updateRoomState - Updates the app UI with the new state
 * @returns {void}
 */
function handleRemoteParticipantReconnectionUpdates(room, updateParticipantState, participant) {
  room.on('participantReconnecting', function(participant) {
    updateParticipantState(participant.state);
  });

  room.on('participantReconnected', function(participant) {
    updateParticipantState(participant.state);
  });
}

/**
 * Listen to LocalParticipant reconnection events and update the UI accordingly
 * @param {Room} room - The Room you have joined
 * @param {function} updateRoomState - Updates the app UI with the new state
 * @returns {void}
 */
function handleLocalParticipantReconnectionUpdates(room, updateParticipantState) {
  const localParticipant = room.localParticipant;

  localParticipant.on('reconnecting', function() {
    updateParticipantState(localParticipant.state);
  });

  localParticipant.on('reconnected', function() {
    updateParticipantState(localParticipant.state);
  });
}

exports.handleLocalParticipantReconnectionUpdates = handleLocalParticipantReconnectionUpdates;
exports.handleRemoteParticipantReconnectionUpdates = handleRemoteParticipantReconnectionUpdates;

```

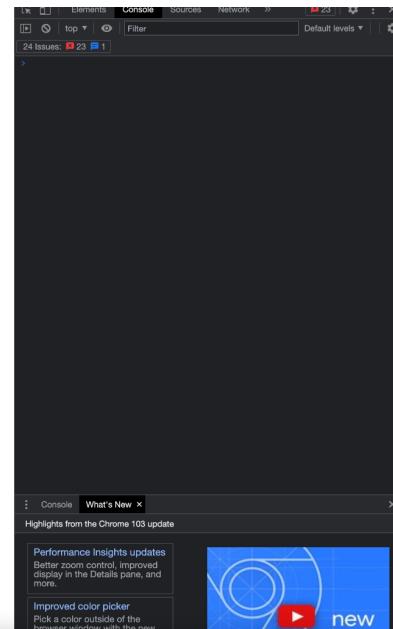
Local View



Simulate Reconnection

State
Connected
Reconnecting

Remote View



Reconnection States and Events

Reconnection States and Events

```
'use strict';

/**
 * Listen to Room reconnection events and update the UI accordingly.
 * @param {Room} room - The Room you have joined
 * @param {function} updateRoomState - Updates the app UI with the new Room state
 * @returns {void}
 */
function setupReconnectionUpdates(room, updateRoomState) {
  room.on('disconnected', (room, error) => {
    if (error.code === 29104) {
      console.log(`Signaling reconnection failed due to expired AccessToken`);
    } else if (error.code === 53000) {
      console.log(`Signaling reconnection attempts exhausted`);
    } else if (error.code === 53204) {
      console.log(`Signaling reconnection took too long!`);
    }
    updateRoomState(room.state);
  });

  room.on('reconnected', function() {
    console.log(`Reconnected to the Room!`);
    updateRoomState(room.state);
  });

  room.on('reconnecting', function(error) {
    if (error.code === 53001) {
      console.log(`Reconnecting your signaling connection!`, error.message);
    } else if (error.code === 53405) {
      console.log(`Reconnecting your media connection!`, error.message);
    }
    updateRoomState(room.state);
  });
}

exports setupReconnectionUpdates = setupReconnectionUpdates;
```

Room state is:

Connected

Disconnected

Reconnecting

Create Room

After you have created the Room, please either turn off your internet network for a little while and then turn it back on, or switch between networks. The Twilio Video SDK will try to re-establish connection to the Room, which will transition to the "connecting" state. Once reconnection is complete, the Room will transition back to the "connected" state. You can join other user to the room using button below.

Join Room

```
2022-07-21T02:24:54.848Z debug [connect #1] Transport state changed:disconnected->connected
2022-07-21T02:24:54.849Z debug [connect #1] Transport state changed:connected->disconnected
2022-07-21T02:24:54.849Z debug [connect #1] Creating a new RemoteParticipant for room: Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.849Z debug [connect #1] Setting up RemoteParticipant creation for all subsequent ParticipantSignalings that connect to the RoomSignalng
2022-07-21T02:24:54.849Z info [Room #1] Room created: Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.849Z debug [connect #1] Room created a new Room: Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.849Z debug [connect #1] Initial RemoteParticipants: []
2022-07-21T02:24:54.849Z debug [connect #1] Creating a new Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.850Z info [connect #1] Connected to Room: Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.850Z info [Room #1] Room name: Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.850Z debug [connect #1] Room: Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.850Z debug [connect #1] Room: Room {id: 'RM008beb4b705acd526013824255de92497', name: 'LocalParticipant', peerConnections: Map(0), ...}
2022-07-21T02:24:54.851Z debug [TwilioConnection #: wss://gl.ohai.vss.twilio.com/signaling] Incoming: {"body": {"type": "peer_update", "peer_connections": [{"id": "1518f2e3-719f-4d84-b14c-2f590ab4343b", "status": "closed", "revision": 39}, {"id": "90080beb4b705acd526013824255de92497", "status": "open", "revision": 1}], "participants": [{"id": "94859a709464ba42e75b3e1d2d5939745", "name": "Juliet Jenova Utica", "tracks": [], "revision": 1}, {"id": "94859a709464ba42e75b3e1d2d5939745", "name": "Juliet Jenova Utica", "tracks": [], "revision": 1}], "state": "connected", "media_warnings": []}, "type": "msg"}
2022-07-21T02:24:54.851Z debug [TwilioConnection #: wss://gl.ohai.vss.twilio.com/signaling] Incoming: {"type": "heartbeat"}
2022-07-21T02:24:58.797Z debug [TwilioConnection #: wss://gl.ohai.vss.twilio.com/signaling] Outgoing: {"type": "heartbeat"}
```

Media Selection

Select Media Devices

Audio Input
 Default - MacBook Pro Microphone (Built-in)

Video Input
 FaceTime HD Camera (Built-in) (05ac:8514)

Audio Output
 MacBook Pro Speakers (Built-in)

Highlights from the Chrome 103 update

Performance Insights updates
 Better zoom control, improved display in the details pane, and more.

Improved color picker
 Pick a color outside of the browser window with the new



Enabling and Disabling Tracks

Enabling and Disabling Tracks

```
use strict';

/**
 * Mute/unmute your media in a Room.
 * @param {Room} room - The Room you have joined
 * @param {'audio'|'video'} kind - The type of media you want to mute/unmute
 * @param {'mute'|'unmute'} action - Whether you want to mute/unmute
 */
function muteOrUnmuteYourMedia(room, kind, action) {
  const publications = kind === 'audio'
    ? room.localParticipant.audioTracks
    : room.localParticipant.videoTracks;

  publications.forEach(function(publication) {
    if (action === 'mute') {
      publication.track.disable();
    } else {
      publication.track.enable();
    }
  });
}

/**
 * Mute your audio in a Room.
 * @param {Room} room - The Room you have joined
 * @returns {void}
 */
function muteYourAudio(room) {
  muteOrUnmuteYourMedia(room, 'audio', 'mute');
}

/**
 * Mute your video in a Room.
 * @param {Room} room - The Room you have joined
 * @returns {void}
 */
function muteYourVideo(room) {
  muteOrUnmuteYourMedia(room, 'video', 'mute');
}

/**
 * Unmute your audio in a Room.
 * @param {Room} room - The Room you have joined
 */


```

Local Media Controls

Enable Audio

Enable Video

RemoteParticipant View

```
24 issues: 0 23 1
("revision":1,"rules":[{"type":"include","all":true}]),"subscribed": {"revision":1,"tracks":[{"id":"708846c-c173-486d-b6b-03439355486c","sid": "MT5e563af0f10803535483c4e07293293946"}, {"id": "4dec5636-21ce-4ab3-9fb0-03439355486c"}], "published": [{"track": "audio", "type": "audio", "id": "4dec5636-21ce-4ab3-9fb0-03439355486c"}]}

2022-07-21T02:25:36.642Z debug [RemoteAudioTrack #3: index.js:27991 MT5e563af0f10803535483c4e07293293946] Attempting to detach from elements: > [audio]
2022-07-21T02:25:36.985Z debug [TwilioConnection #2: wss://gl1.index.js:27991 obal.vss.twilio.com/signaling] Outgoing: {"type": "heartbeat"}
2022-07-21T02:25:39.572Z debug [TwilioConnection #1: wss://gl1.index.js:27991 obal.vss.twilio.com/signaling] Incoming: {"type": "heartbeat"}
2022-07-21T02:25:40.115Z debug [TwilioConnection #2: wss://gl1.index.js:27991 obal.vss.twilio.com/signaling] Incoming: {"type": "heartbeat"}
2022-07-21T02:25:40.115Z debug [TwilioConnection #1: wss://gl1.index.js:27991 MT469fc03aa74302dd6d46f5ba0e17cf] Dimensions changed: > {width: 400, height: 300}
2022-07-21T02:25:41.404Z debug [TwilioConnection #1: wss://gl1.index.js:27991 obal.vss.twilio.com/signaling] Outgoing: {"type": "heartbeat"}
2022-07-21T02:25:41.404Z info [connect #1] event: index.js:27991 > {elapsedTime: 10745, group: "quality", level: "info", name: "stats-report", timestamp: 1658703416091, ...}
2022-07-21T02:25:41.492Z info [connect #1] event: index.js:27991 > {elapsedTime: 10745, group: "active-ice-can-didate-par", level: "info", name: "active-ice-can-didate-par", timestamp: 1658703416052, ...}
2022-07-21T02:25:41.778Z debug [TwilioConnection #2: wss://gl1.index.js:27991 obal.vss.twilio.com/signaling] Outgoing: {"type": "heartbeat"}
2022-07-21T02:25:41.852Z info [connect #2] event: index.js:27991 > {elapsedTime: 10745, group: "quality", level: "info", name: "stats-report", timestamp: 1658703424205, ...}
2022-07-21T02:25:42.466Z info [connect #2] event: index.js:27991 > {elapsedTime: 10745, group: "active-ice-can-didate-par", level: "info", name: "active-ice-can-didate-par", timestamp: 1658703424206, ...}
2022-07-21T02:25:43.672Z debug [TwilioConnection #1: wss://gl1.index.js:27991 obal.vss.twilio.com/signaling] Incoming: {"type": "heartbeat"}
2022-07-21T02:25:44.216Z debug [TwilioConnection #2: wss://gl1.index.js:27991 obal.vss.twilio.com/signaling] Incoming: {"type": "heartbeat"}
```

Console What's New Issues

- Audit usage of navigator.userAgent, navigator.appVersion, and navigator.platform
- Indicate whether to send a cookie in a cross-site request by specifying its SameSite attribute

Video Track Automatic Controls

VIDEO TRACK AUTOMATIC CONTROLS

```
'use strict';

var Video = require('twilio-video');

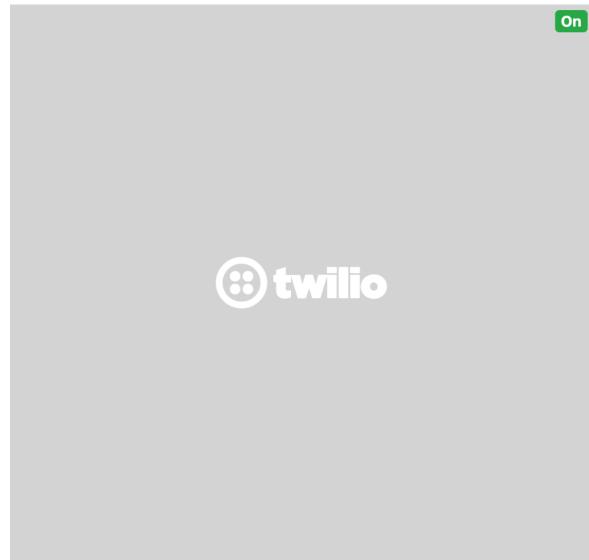
/**
 * Connect to a Room with 'auto' mode. This is the default mode.
 * @param {string} token - AccessToken for joining the Room
 * @returns {Room}
 */
function joinRoom(token) {
  return Video.connect(token, {
    name: 'my-cool-room',
    bandwidthProfile: {
      video: {
        contentPreferencesMode: 'auto',
        clientTrackSwitchOffControl: 'auto'
      }
    }
  });
}

module.exports.joinRoom = joinRoom;
```

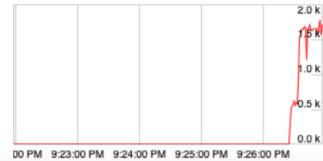
REMOTE VIDEO CONTROLS

Toggle Visibility: [Show](#) [Hide](#)
Video Size: [960x540](#) ▾

Remote Video Track



Video Bitrate



Screen Share

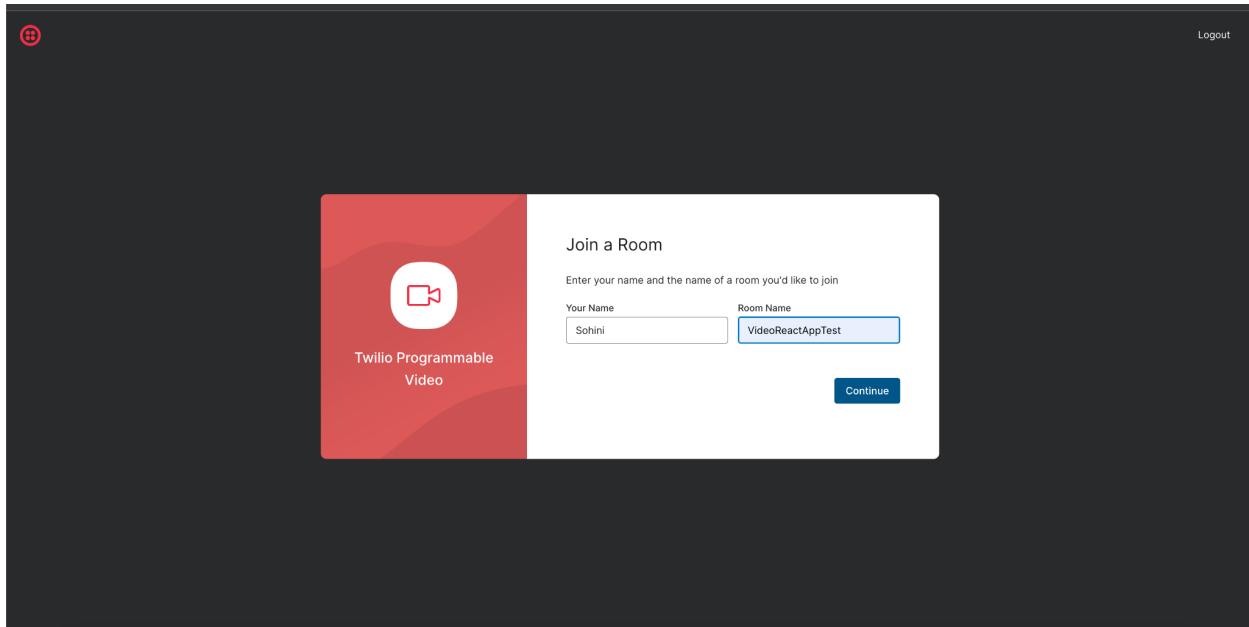
The screenshot shows a development environment with the following components:

- Code Editor:** Displays a snippet of JavaScript from the `twilio-video.js` file, specifically the `createScreenTrack` function.
- Local Screen Tab:** A browser tab titled "Local Screen" showing a video feed of the local desktop.
- Remote Screen Tab:** A browser tab titled "Remote Screen" showing a video feed of a remote desktop, with a message at the bottom stating "<http://localhost:3000> is sharing your screen." and buttons for "Stop sharing" and "Hide".
- Developer Tools:** An open sidebar in the browser showing the "Console" tab with 24 issues and 23 warnings.

- Now you are going to make changes to the actual Quickstart by implementing the code from the Examples.
- Once all the examples have been written and tested on your local Quickstart repo, merge the changes into your repo on GH and share with the Supportability Engineer.

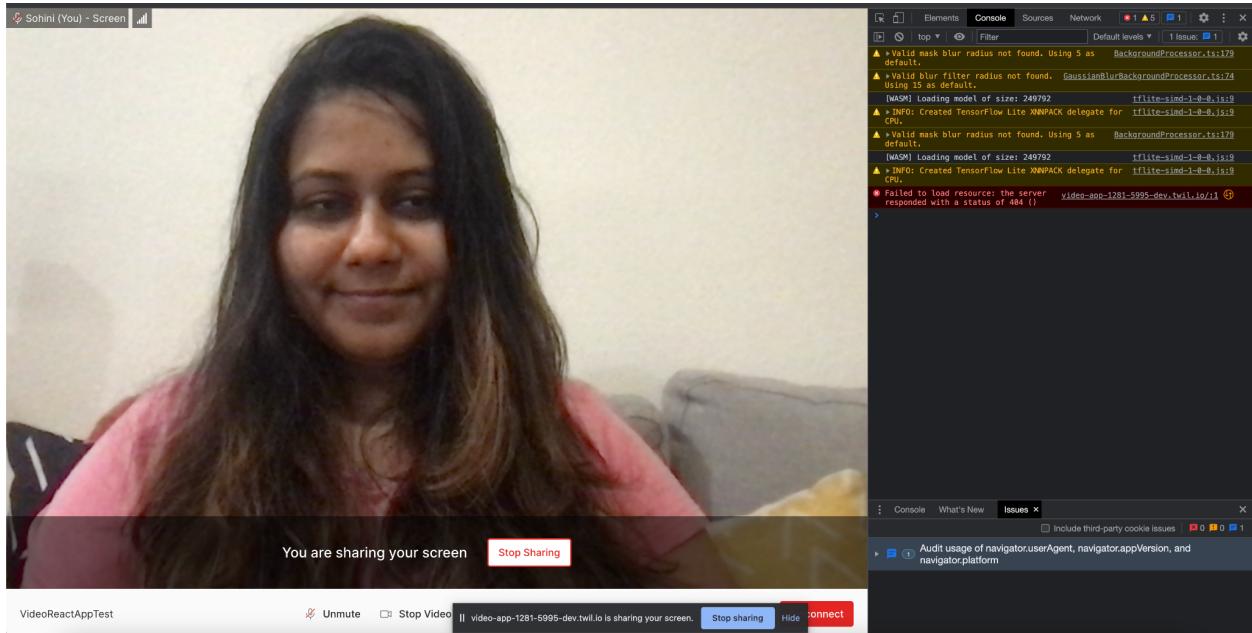
Ahoy! Demo App

- Set up the [Video Demo React App](#) on your laptop.
- **VideoReactAppTest - RM6ce863d590d0bbeb1c58abd46198547c**

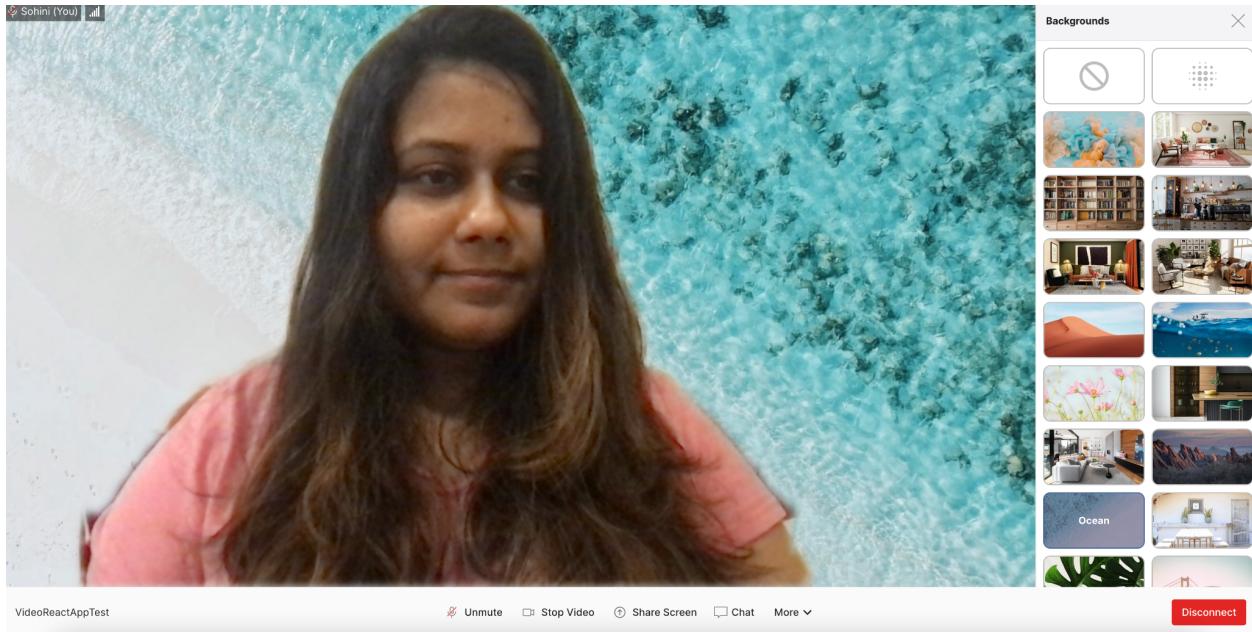


- Start with a Group Room. While in the rooms be sure to have the Javascript Console and the Request Inspector Open, and that you are capturing screen shots of the output from both the JS Console and the Request Inspector.
- As with the Quickstart, take note of the Room SIDs, and take many screenshots showing various actions with the Demo App. Be sure to smile when on camera! Upload the screenshots into GH and create a file in GH that has descriptions for the screenshot filenames and their respective SIDs.
- Share the Demo App URL and passcode with a friend and create a Group room.
 - Web App URL:
<https://video-app-1281-5995-dev.twil.io?passcode=52008012815995>
 - Passcode: 520 080 1281 5995
 - Expires: Tue Jul 26 2022 12:43:06 GMT-0500
 - Room Type: group
- While in the various rooms, each of you should toggle mute both video and audio tracks, toggle recording on and off. Send some chats to one another. If possible quickly disconnect your network and reconnect to see how the room handles the reconnect. Test the device input dropdowns, bonus points for adding [krisp.ai](#) to the audio input selector. Test the virtual background feature.
- Create at minimum two Group Rooms with two or more participants to generate logs and recordings.

Testing Screenshare



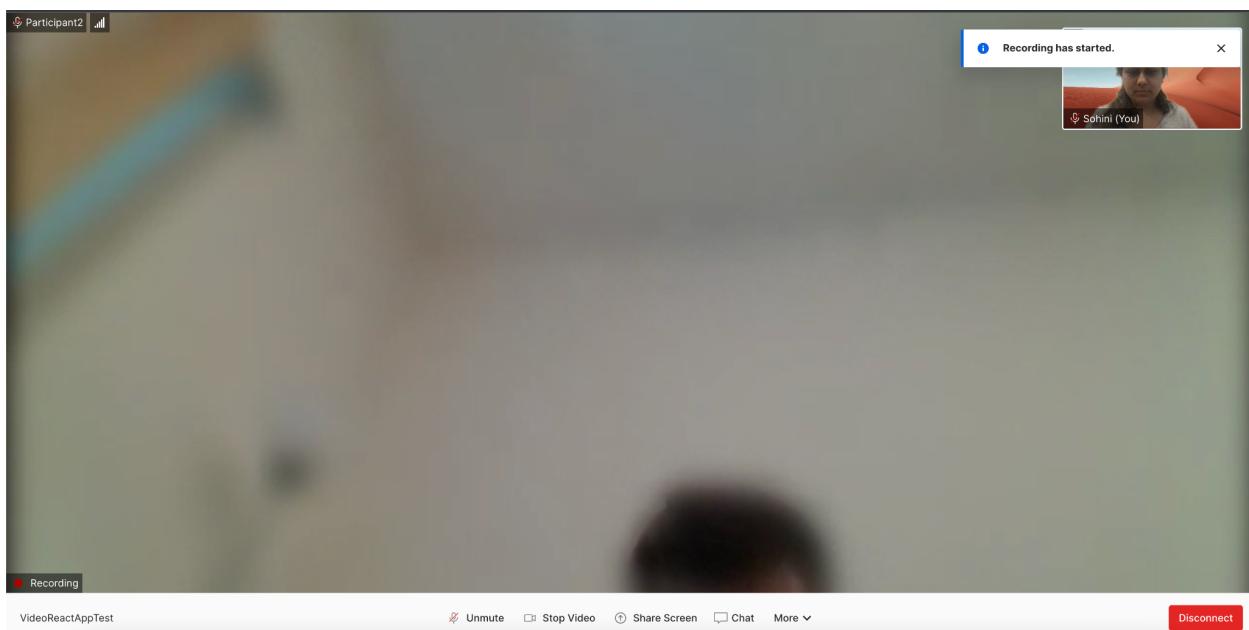
Testing Virtual Backgrounds



Testing the Chat feature with another participant



Testing Recording



Testing Video Stats/Info

Twilio Video Room Monitor Info Stats X

Room information:

- Room Name: VideoReactAppTest
- SID: RM5e69a0def15a6aa6ebd2597142268059
- State: connected
- Dominant Speaker: null
- Media Region: us1
- Is Recording: false
- Total Sent Bandwidth: 0.3kbps
- Total Received Bandwidth: 0kbps

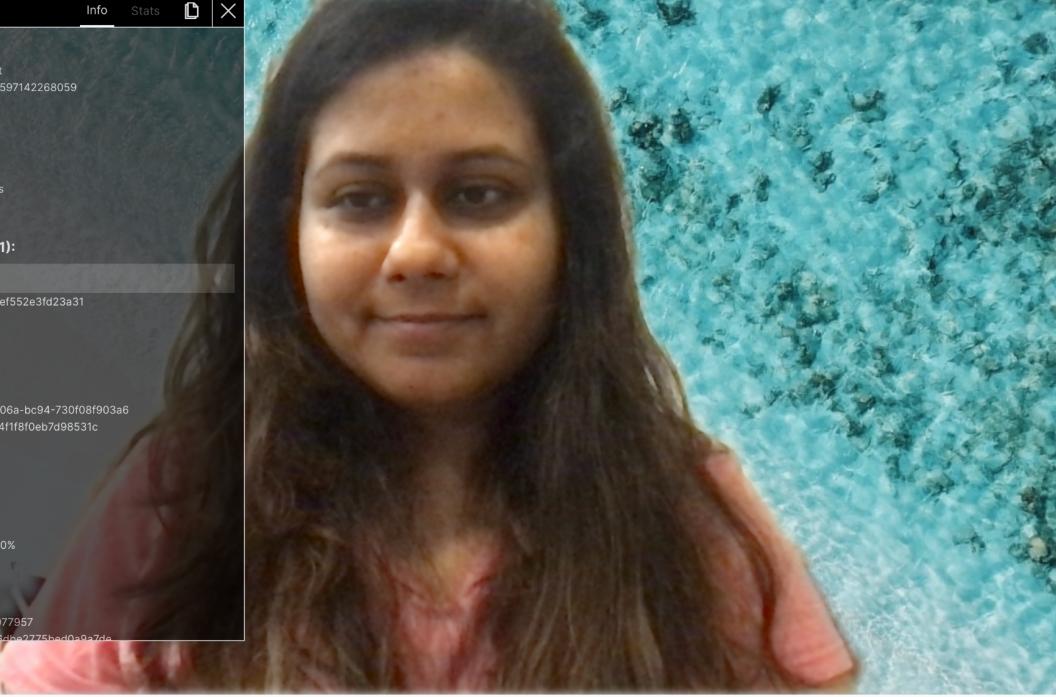
▶ Connection Options

Participant Information (1):

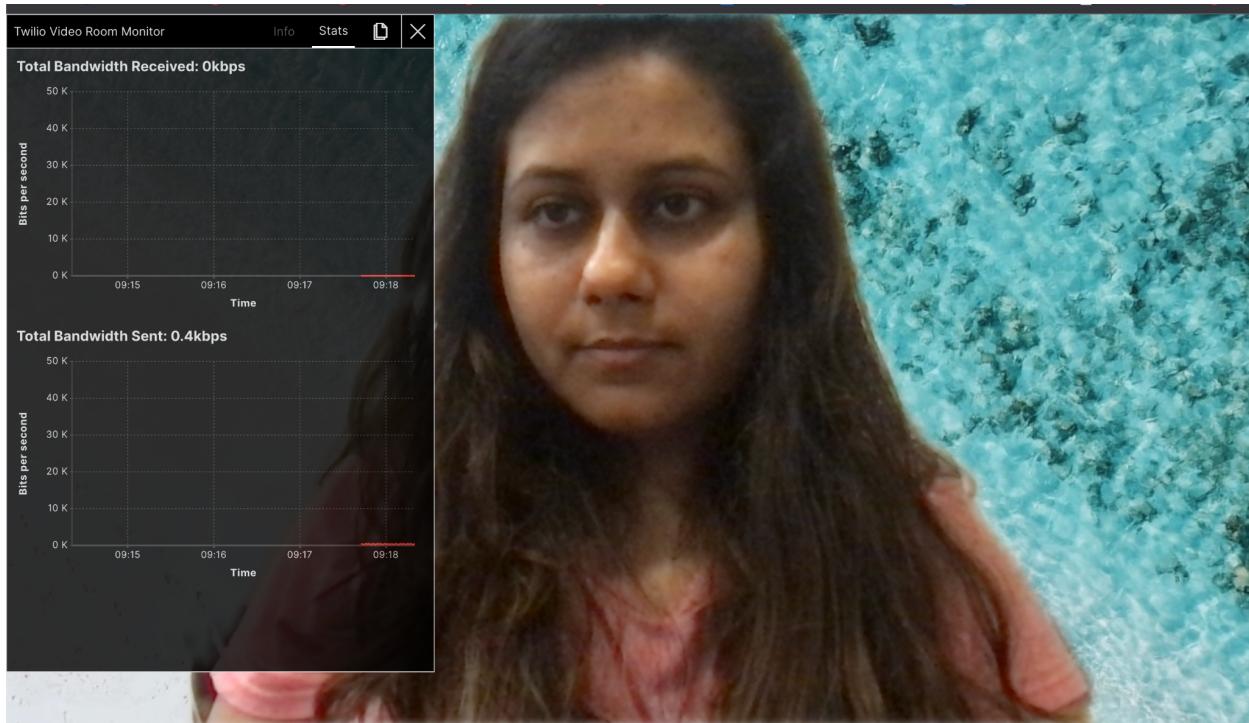
- Sohini

SID: PA3132fad3418cac319aeef552e3fd23a31
isReconnecting: false
networkQualityLevel: 5

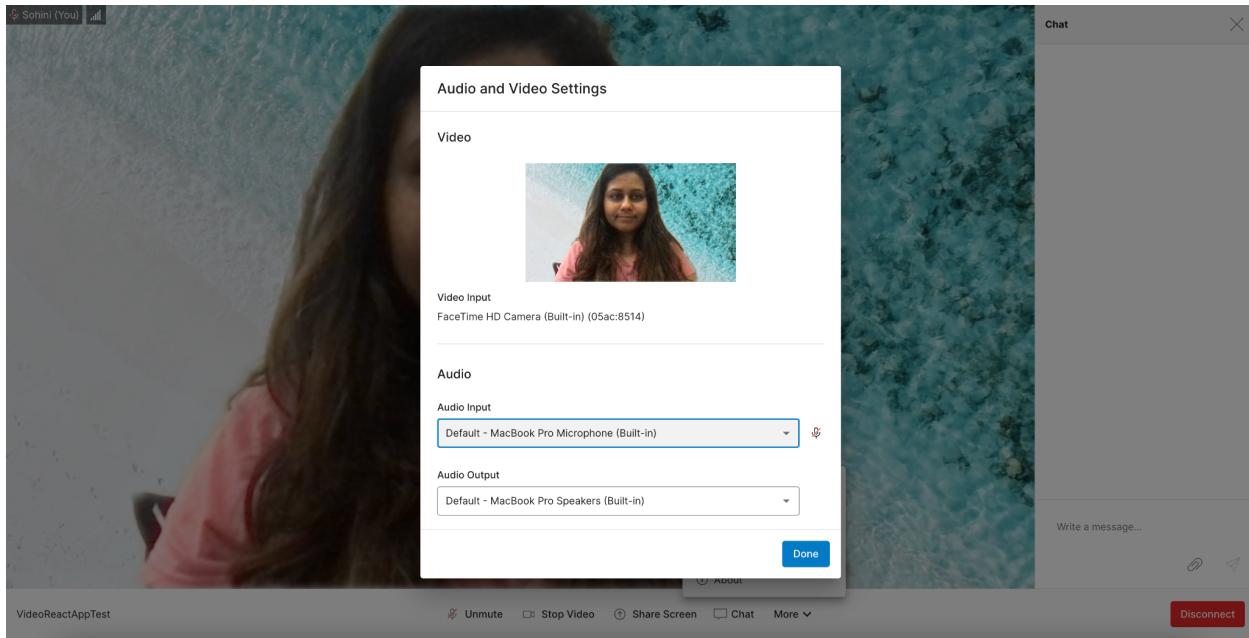
- ▶ Data Tracks (0)
- ▶ Audio Tracks (1)
 - Name: 69df6679-a375-406a-bc94-730f08f903a6
SID: Mtaaf0bfab05156e4ff1bf0eb7d98531c
isSubscribed: true
isEnabled: false
Bandwidth: 0kbps
Codec: opus
Jitter: 0
Packets Lost: 0
Packet Loss Percentage: 0%
- ▶ Media Stream Track
- ▶ Video Tracks (1)
 - Name: camera-1658283077957
SID: MT185bcfd4027b16dbe2775bed0a9a7de



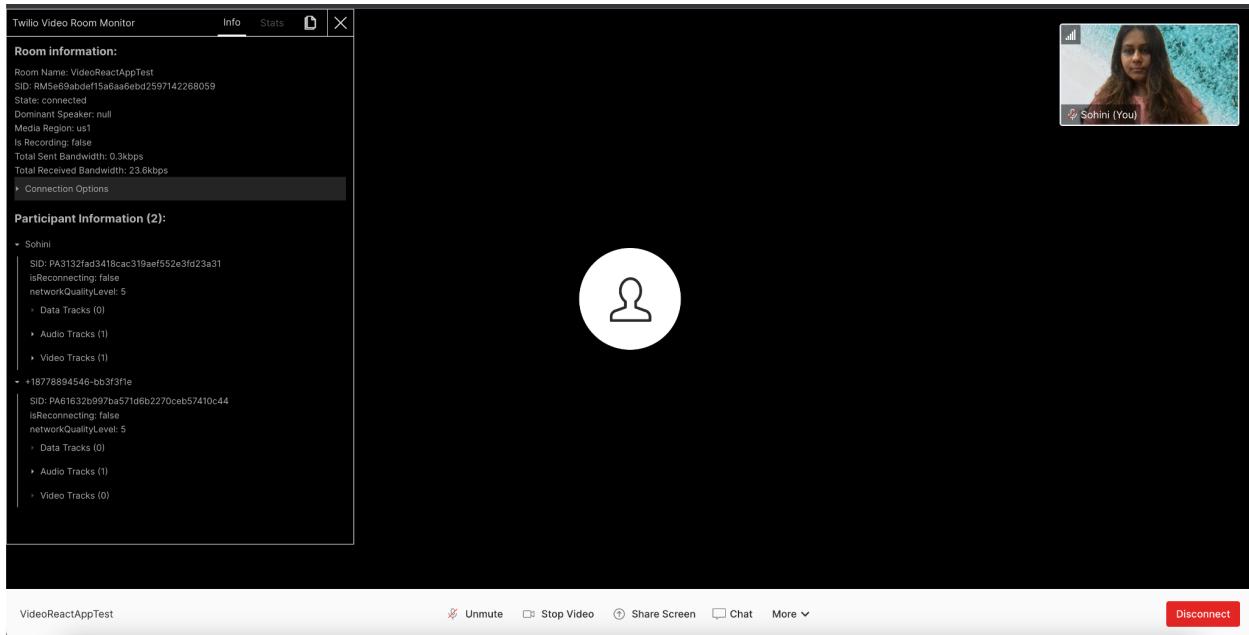
VideoReactAppTest ⚡ Unmute 📹 Stop Video 🖥 Share Screen 💬 Chat More ▾



Changing Audio/Video Inputs



- Add one PSTN participant to a room following [Adding Programmable Voice participants to Video Rooms](#)
- Twilio number used - **+15075785525**



Create one Peer-to-Peer room

```
"ERROR CODE 50013 FROM TWILIO. ROOM EXISTS. See https://www.twilio.com/docs/api/video/v1/rooms#create"
smitra@C02FC4MUMD6M Downloads % twilio api:video:v1:rooms:create \
    --unique-name PeerToPeerTest \
    --type peer-to-peer
SID                               Unique Name      Status
RMf2d66c074b3259155fa2aa63e404433c  PeerToPeerTest  in-progress
smitra@C02FC4MUMD6M Downloads %
```

 **Sohini Mitra** 1:28 PM
RMa50ace1b43d7e4a92e87a2edd16e3639

 **bot** APP 1:28 PM
Room Analysis for RMa50ace1b43d7e4a92e87a2edd16e3639

Room Info	
AccountSid	Created On
AC29f38c1506bf188995edc6c312c84d31	2022-07-20 18:28:42.990
Ended On	Room Name
	PeerToPeer
VideoMediaServerSid	Status Callback
	POST https://webhook.site/68ab1fb3-6180-4359-978d-2be3f02a3263
Room Type	Unique Participants
peer-to-peer	0
Unique Identities	Turn Enabled
0	True
Recordings Enabled	IsAdhoc
None	False
Completed via REST API	Media Region
No	
Video Codecs	Status
	in-progress
Maximum Participants	Maximum Participant Duration
10	14400

No participant found for Room sid: RMa50ace1b43d7e4a92e87a2edd16e3639.

Explain the differences in output in a document in GH between Group rooms and P2P rooms.

Peer to Peer Rooms -

Media is encrypted end-to-end (E2E) using WebRTC security protocols.

Twilio does not mediate the media exchange, which takes place through direct communication among Participants. The only exception is when media exchange requires TURN. In that case, a TURN server will blindly relay the encrypted media bits to guarantee connectivity. The TURN server cannot decrypt or manipulate the media.

As Twilio does not intercept the media in P2P Rooms, it is not possible to record or transcode the media or make it interoperate with other RTC services.

Despite not being in the media path, Twilio manages the signaling path, making it possible for Participants to discover each other and negotiate the communications in agreement with the application and SDK requirements. Hence, signaling connectivity to Twilio's cloud is still necessary.

Group Rooms -

Participants publish media to a Twilio Selective Forwarding Unit (SFU). An SFU is a Media Server that decrypts the media, processes, re-encrypts, and routes the media tracks to the correct destinations.

Media is not E2E encrypted, as the SFU keeps media unencrypted in memory to process it.

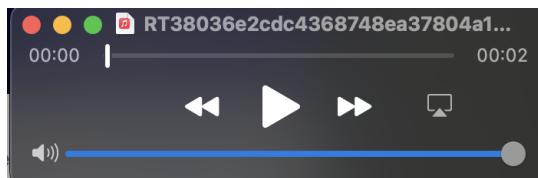
Services such as recordings and public switched telephone network (PSTN) interoperability can be added, as Twilio acts as media middleware.

Media is exchanged directly between participants in P2P rooms.

Recordings

- Download and Play an audio recording and a video recording.

Audio Recording



Recording Log

Properties

Recording SID:	RT38036e2cdc4368748ea37804a1019392	Duration:	3 sec
Room SID:	RMfd9c3802a5051b87490a157f5ba6ea36	Status:	Completed
Date:	02:04:41 UTC 2022-07-20	Size:	0.0 MB
Source SID:	MT9318265f07426c1382ab2e913b6d15e8	Type:	audio
Participant SID:	PA6557bb99fb020d3dbbf16e42ce08b75	Codec:	opus

Media

[Download](#)

Video Recording

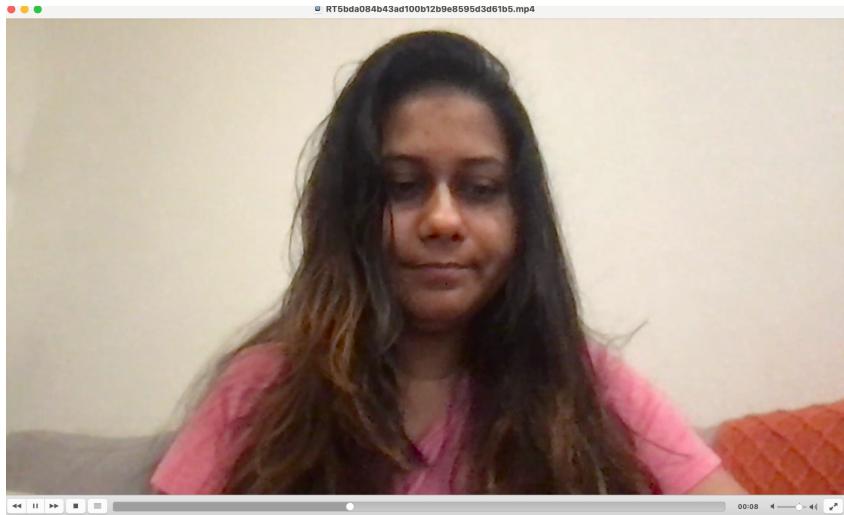
Recording Log

Properties

Recording SID:	RT5bda084b43ad100b12b9e8595d3d61b5	Duration:	23 sec
Room SID:	RM5e69abdef15a6aa6ebd2597142268059	Status:	Completed
Date:	02:11:25 UTC 2022-07-20	Size:	5.5 MB
Source SID:	MT185bcfd4027ab16dbe2775bed0a9a7de	Type:	video
Participant SID:	PA3132fad3418cac319aef552e3fd23a31	Codec:	vp8

Media

[Download](#)



Compositions

- Create an audio-only composition.

```
twilio api:video:v1:compositions:create \
--audio-sources PA6557bb99fb020d3dbbf16e42ce08b75 \
--status-callback
https://webhook.site/68ab1fb3-6180-4359-978d-2be3f02a3263 \
--format mp4 \
--room-sid RMfd9c3802a5051b87490a157f5ba6ea36
```

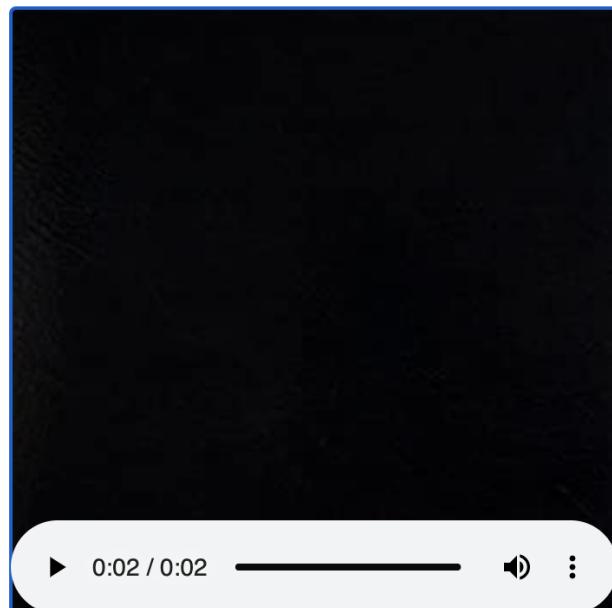
Sid - CJa01d0459253b607873bb9e14d9c55565

Request Details		Permalink	Raw content	Export as
POST	https://webhook.site/68ab1fb3-6180-4359-978d-2be3f02a3263			
Host	54.146.234.88 whois			
Date	07/19/2022 10:27:38 PM (a few seconds ago)			
Size	389 bytes			
ID	4260ca0e-bc0b-49bf-9c72-9f87bc4faf1a			
Files				
Query strings	(empty)			
Headers				
connection	close			
user-agent	TwilioProxy/1.1			
content-length	389			
x-home-region	us1			
accept	*/*			
x-twilio-signature	HNPn3wUoQbdDxblK9zmw0Hb0=			
i-twilio-idempotency-token	15099d99-924d-48cf-9982-c4d5c1bdcc91			
content-type	application/x-www-form-urlencoded; charset=UTF-8			
host	webhook.site			
Form values				
MediaUri	/v1/Compositions/CJa01d0459253b607873bb9e14d9c55565/Media			
RoomSid	RMfd9c3802a5051b87490a157f5ba6ea36			
Size	46355			
CompositionSid	CJa01d0459253b607873bb9e14d9c55565			
Duration	3			
StatusCallbackEvent	composition-available			
Timestamp	2022-07-20T03:27:38.141Z			
AccountSid	AC29f38c1506bf188995edc6c312c84d31			
CompositionUri	/v1/Compositions/CJa01d0459253b607873bb9e14d9c55565			
Raw Content				
<pre>MediaUri=%2Fv1%2FCompositions%2FCJa01d0459253b607873bb9e14d9c55565%2FMedia&RoomSid=RMfd9c3802a5051b87490a157f5ba6ea36&Size=46355&CompositionSid=CJa01d0459253b607873bb9e14d9c55565&Duration=3&StatusCallbackEvent=composition-available&Timestamp=2022-07-20T03:27:38.141Z&AccountSid=AC29f38c1506bf188995edc6c312c84d31&CompositionUri=%2Fv1%2FCompositions%2FCJa01d0459253b607873bb9e14d9c55565</pre>				

Properties

Composition SID: CJa01d0459253b607873bb9e14d9c55565	Duration: 3 sec
Room SID: RMfd9c3802a5051b87490a157f5ba6ea36	Status: Completed
Date: 2022-07-20	Size: 0.0 MB
Start Time: 03:27:32 UTC 2022-07-20	Format: mp4
End Time: 03:27:38 UTC 2022-07-20	Encrypted: False
Location: Twilio	

Media



- Create a video grid composition.

```
twilio api:video:v1:compositions:create \
--audio-sources * \
--video-layout "{\"grid\": {\"video_sources\": [\"*\"]}}" \
--status-callback
https://webhook.site/68ab1fb3-6180-4359-978d-2be3f02a3263 \
--format mp4 \
--room-sid RMfd9c3802a5051b87490a157f5ba6ea36
```

Sid - **CJ04ebc749dffff2f42c5d3d9ad41c19b4**

Request Details		Permalink	Raw content	Export as
POST	https://webhook.site/68ab1fb3-6180-4359-978d-2be3f02a3263			
Host	3.90.140.18 whois			
Date	07/19/2022 10:25:09 PM (a few seconds ago)			
Size	390 bytes			
ID	e2799bfc-bf9b-4684-bd14-57378c86330f			
Files				
Query strings				
(empty)				
Headers				
connection	close			
user-agent	TwilioProxy/1.1			
content-length	390			
x-home-region	us1			
accept	/*/*			
x-twilio-signature	yfRrWYyRUTjQijEf7u1od5+ym4=			
i-twilio-idempotency-token	e625e891-24b8-4020-869b-37d93ba2b030			
content-type	application/x-www-form-urlencoded; charset=UTF-8			
host	webhook.site			
Form values				
MediaUri	/v1/Compositions/CJ04ebc749dffff2f42c5d3d9ad41c19b4/Media			
RoomSid	RHfd9c3802a5051b87490a157f5ba6ea36			
Size	123000			
CompositionSid	CJ04ebc749dffff2f42c5d3d9ad41c19b4			
Duration	2			
StatusCallbackEvent	composition-available			
Timestamp	2022-07-20T03:25:09.070Z			
AccountId	AC29f38c1506bf188995edc6c312c84d31			
CompositionUri	/v1/Compositions/CJ04ebc749dffff2f42c5d3d9ad41c19b4			
<input checked="" type="checkbox"/> Format JSON <input checked="" type="checkbox"/> Word-Wrap <input type="button" value="Copy"/>				
<pre>MediaUri=%2Fv1%2FCompositions%2FCJ04ebc749dffff2f42c5d3d9ad41c19b4%2FMedia&RoomSid=RMfd9c3802a5051b87490a157f5ba6ea36&Size=123000&CompositionSid=CJ04ebc749dffff2f42c5d3d9ad41c19b4&Duration=2&StatusCallbackEvent=composition-available&Timestamp=2022-07-20T03%3A25%3A09.070Z&AccountId=AC29f38c1506bf188995edc6c312c84d31&CompositionUri=%2Fv1%2FCompositions%2FCJ04ebc749dffff2f42c5d3d9ad41c19b4</pre>				
Raw Content				
<pre>MediaUri=%2Fv1%2FCompositions%2FCJ04ebc749dffff2f42c5d3d9ad41c19b4%2FMedia&RoomSid=RMfd9c3802a5051b87490a157f5ba6ea36&Size=123000&CompositionSid=CJ04ebc749dffff2f42c5d3d9ad41c19b4&Duration=2&StatusCallbackEvent=composition-available&Timestamp=2022-07-20T03%3A25%3A09.070Z&AccountId=AC29f38c1506bf188995edc6c312c84d31&CompositionUri=%2Fv1%2FCompositions%2FCJ04ebc749dffff2f42c5d3d9ad41c19b4</pre>				

Programmable-Video Bot (a Slack App)

- Type "help" into the @Programmable-Video Bot - an app you need to add in slack. Use the commands based on your own video logs based on SIDs from the video react demo app.
- Please use every command at least twice with various switches. Take screenshots and/or send the pop-out shareable (make sure you allow it) URLs; add these to the GH Repo and file.

AC29f38c1506bf188995edc6c312c84d31

 Sonini Mitra 10:38 PM
AC29f38c1506bf188995edc6c312c84d31 Today

 bot APP 10:38 PM
Account Info for AC29f38c1506bf188995edc6c312c84d31

Account Info	
Created On	Alias
2022-06-13T00:09:12.280Z	Video
Modified On	Configuration
2022-07-20T02:31:24.214Z	VSd157efaa757f8897c29078ebbe798844
Adhoc Room	Turn
enabled	enabled
Media Region	Record Participants on connect
us1	enabled
Room Timeout	Room Type
300	group
Status Callback URL	Status Callback Method
https://webhook.site/68ab1fb3-6180-4359-978d-2be3f02a3263	POST
Video Codecs	Account Read Rate Limit
VP8,H264	80
Account Write Rate Limit	Rooms Concurrent High
20	5
Participants Concurrent High	Concurrent Max Limit
6	10000

 Programmable-Video Bot APP 10:38 PM
Recently created Rooms ▾

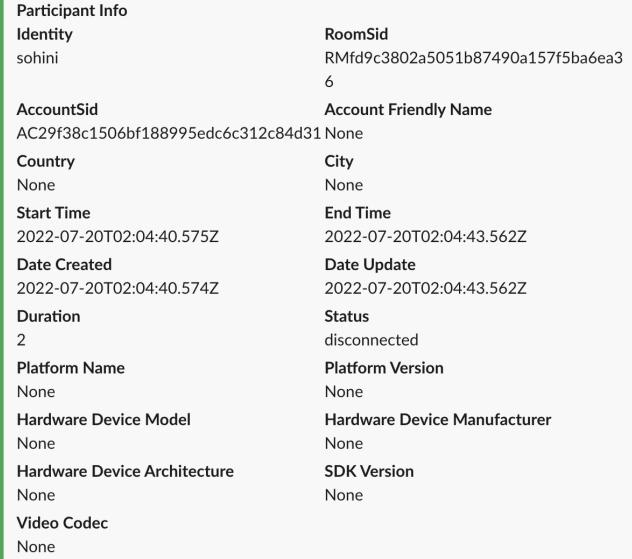
1	timestamp	room sid	room name	media
	region	room type	video codecs	
2	-	-	-	-

PA6557bb99fb020d3dbbf16e42ce08b75

 **Sohini Mitra** 10:39 PM
PA6557bb99fb020d3dbbf16e42ce08b75

Today 

 **bot** APP 10:39 PM
Participant Analysis for PA6557bb99fb020d3dbbf16e42ce08b75



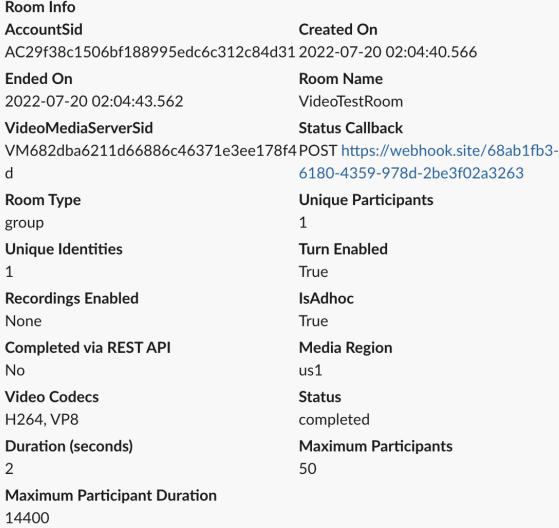
Participant Info	
Identity	RoomSid
sohini	RMfd9c3802a5051b87490a157f5ba6ea3
	6
AccountSid	Account Friendly Name
AC29f38c1506bf188995edc6c312c84d31	None
Country	City
None	None
Start Time	End Time
2022-07-20T02:04:40.575Z	2022-07-20T02:04:43.562Z
Date Created	Date Update
2022-07-20T02:04:40.574Z	2022-07-20T02:04:43.562Z
Duration	Status
2	disconnected
Platform Name	Platform Version
None	None
Hardware Device Model	Hardware Device Manufacturer
None	None
Hardware Device Architecture	SDK Version
None	None
Video Codec	
None	

 **Programmable-Video Bot** APP 10:39 PM
Participant Summary Report 

RMfd9c3802a5051b87490a157f5ba6ea36

 **Sohini Mitra** 10:44 PM
RMfd9c3802a5051b87490a157f5ba6ea36

 **bot** APP 10:44 PM
Room Analysis for RMfd9c3802a5051b87490a157f5ba6ea36



Room Info	
AccountSid	Created On
AC29f38c1506bf188995edc6c312c84d31	2022-07-20 02:04:40.566
Ended On	Room Name
2022-07-20 02:04:43.562	VideoTestRoom
VideoMediaServerSid	Status Callback
VM682dba6211d66886c46371e3ee178f4	POST https://webhook.site/68ab1fb3-d180-4359-978d-2be3f02a3263
Room Type	Unique Participants
group	1
Unique Identities	Turn Enabled
1	True
Recordings Enabled	IsAdhoc
None	True
Completed via REST API	Media Region
No	us1
Video Codecs	Status
H264, VP8	completed
Duration (seconds)	Maximum Participants
2	50
Maximum Participant Duration	
14400	

 **Programmable-Video Bot** APP 10:44 PM
Participants Info 

Sohini Mitra 10:50 PM

compositions --sid CJa01d0459253b607873bb9e14d9c55565 --account-sid AC29f38c1506bf188995edc6c312c84d31

Programmable-Video Bot APP 10:50 PM

Compositions metadata report ▾

1	-----	
2	Metadata of CJa01d0459253b607873bb9e14d9c55565	
3	-----	
4	AccountSid	AC29f38c1506bf188995edc6c312c84d31
5	AudioSources	['PA6557bb99fb020d3dbbf16e42ce08b75']
6	AudioSourcesExcluded	[]
7	Bitrate	134
8	DateCompleted	2022-07-20T03:27:38.141Z
9	DateCreated	2022-07-20T03:27:32.232Z
10	DateDeleted	-
11	Duration	3
12	Format	mp4
13	HostedByTwilio	True
14	MediaExternalLocation	-
15	Resolution	640x480
16	RoomSid	RMfd9c3802a5051b87490a157f5ba6ea36
17	Sid	CJa01d0459253b607873bb9e14d9c55565
18	Size	46355
19	Status	completed
20	StatusCallback	https://webhook.site/68ab1fb3-6180-4359-978d-2be3f02a3263
21	StatusCallbackMethod	POST
22	Trim	True
23	Uri	/v1/Compositions/CJa01d0459253b607873bb9e14d9c55565
24	Encrypted	No
25	-----	

 **Sohini Mitra** 10:49 PM
compositions --room-sid RMfd9c3802a5051b87490a157f5ba6ea36 --account-sid AC29f38c1506bf188995edc6c312c84d31

 **Programmable-Video Bot** APP 10:49 PM
Compositions for AC29f38c1506bf188995edc6c312c84d31 and room RMfd9c3802a5051b87490a157f5ba6ea36 ▾

1	-----
2	Compositions for room RMfd9c3802a5051b87490a157f5ba6ea36
3	-----
4	CJa01d0459253b607873bb9e14d9c55565
5	CJ04ebc749dffff2f42c5d3d9ad41c19b4
6	-----

 **Sohini Mitra** 10:51 PM
vms --room-sid RMfd9c3802a5051b87490a157f5ba6ea36

 **Programmable-Video Bot** APP 10:51 PM
Session for Room: RMfd9c3802a5051b87490a157f5ba6ea36 ▾

```
1 -----  
2 | Account: AC29f38c1506bf188995edc6c312c84d31  
3 | VMS: VM682dba6211d66886c46371e3ee178f4d  
4 | Host: H087c0fb5a26b689b6adba8bca98d51135  
5 | Region: us-east-1  
6 -----  
7 2022-07-20T02:04:40.584759Z - VMS process created  
8 2022-07-20T02:05:44.069198Z - VMS process terminated
```



Sohini Mitra 10:53 PM

vms --vms-sid VM682dba6211d66886c46371e3ee178f4d



Programmable-Video Bot APP 10:53 PM

Session for VMS: VM682dba6211d66886c46371e3ee178f4d ▾

```
1 -----
2 | Account: AC29f38c1506bf188995edc6c312c84d31
3 | Room: RMfd9c3802a5051b87490a157f5ba6ea36
4 | Host: H087c0fb5a26b689b6adba8bca98d51135
5 | Region: us-east-1
6 -----
7     2022-07-20T02:04:40.584759Z - VMS process created
8     2022-07-20T02:05:44.069198Z - VMS process terminated
```



Sohini Mitra 10:54 PM

recordings --room-sid RMfd9c3802a5051b87490a157f5ba6ea36



Programmable-Video Bot APP 10:54 PM

Recording for Room: RMfd9c3802a5051b87490a157f5ba6ea36 ▾

```
1 -----
2 | Account: AC29f38c1506bf188995edc6c312c84d31
3 | Participant: PA6557bb99fb020d3dbbf816e42ce08b75
4 | VMS: VM682dba6211d66886c46371e3ee178f4d
5 | Host: H087c0fb5a26b689b6adba8bca98d51135
6 | Region: us-east-1
7 -----
8     2022-07-20T02:04:40.638362Z - Recording rules changed
9 Track: MT9318265f07426c1382ab2e913b6d15e8
10 -----
11 | Recording: RT38036e2cdc4368748ea37804a1019392
12 | Codec: audio/opus
13 -----
14     2022-07-20T02:04:40.709758Z - Recording started
15     2022-07-20T02:04:40.946740Z - Decodable state changed
16     2022-07-20T02:04:43.564907Z - Recording stopped - Duration: 2593
17 Track: MT853a61ae337fd84b8a93e56a2cabe00d
18 -----
19 | Recording: RT55cd28f381952eb4144563249d39d9cc
20 | Codec: video/VP8
21 -----
22     2022-07-20T02:04:40.711378Z - Recording started
23     2022-07-20T02:04:42.002626Z - Decodable state changed
24     2022-07-20T02:04:42.518860Z - Decodable state changed
25     2022-07-20T02:04:43.564917Z - Recording stopped - Duration: 1488
```



Sohini Mitra 10:56 PM

recordings --participant-sid PA6557bb99fb020d3dbbf16e42ce08b75



Programmable-Video Bot APP 10:56 PM

Recording for Participant: PA6557bb99fb020d3dbbf16e42ce08b75 ▾

```
1 -----
2 | Account: AC29f38c1506bf188995edc6c312c84d31
3 | Room: RMfd9c3802a5051b87490a157f5ba6ea36
4 | VMS: VM682dba6211d66886c46371e3ee178f4d
5 | Host: H087c0fb5a26b689b6adba8bca98d51135
6 | Region: us-east-1
7 -----
8 Track: MT9318265f07426c1382ab2e913b6d15e8
9 -----
10 | Recording: RT38036e2cdc4368748ea37804a1019392
11 | Codec: audio/opus
12 -----
13 2022-07-20T02:04:40.709758Z - Recording started
14 2022-07-20T02:04:40.946740Z - Decodable state changed
15 2022-07-20T02:04:43.564907Z - Recording stopped - Duration: 2593
16 Track: MT853a61ae337fd84b8a93e56a2cabef00d
17 -----
18 | Recording: RT55cd28f381952eb4144563249d39d9cc
19 | Codec: video/VP8
20 -----
21 2022-07-20T02:04:40.711378Z - Recording started
22 2022-07-20T02:04:42.002626Z - Decodable state changed
23 2022-07-20T02:04:42.518860Z - Decodable state changed
24 2022-07-20T02:04:43.564917Z - Recording stopped - Duration: 1488
```



Sohini Mitra 10:57 PM

billing --account-sid AC29f38c1506bf188995edc6c312c84d31



Programmable-Video Bot APP 10:57 PM

Please wait...

New

No data available

 **Sohini Mitra** 10:58 PM
vms2 rooms impact recording --since yesterday --sid AC29f38c1506bf188995edc6c312c84d31

 **Programmable-Video Bot** APP 10:58 PM
Please wait...

---- ROOMS RECORDING ERROR REPORT ----

Since: 2022-07-19 03:58:45.455Z

Until: 2022-07-20 03:58:45.460Z

---- RECORDINGS ----

TOTAL	:	47 (100.00%)
SUCCESS	:	41 (87.23%)
FAILED	:	0 (0.00%)
WARNING	:	6 (12.77%)

---- RECORDING ERRORS ----

Code	Name	Count	Percent
ABSENT_NO_MEDIA		0 (%)
ABSENT_NO_DECODEABLE_MEDIA		0 (%)

---- RECORDING WARNINGS ----

Code	Name	Count	Percent
ABSENT_DISABLED		2 (%)
MEDIA_GAP		4 (%)

Media Recordings Bot (a Slack App)

- Type "help" into the @Media-Recordings bot - an app you need to add in slack. Use the commands based on your own video logs based on SIDs from the video react demo app.
- Please use every command at least twice with various switches. Take screenshots and/or send the pop-out shareable (make sure you allow it) URLs; add these to the GH Repo and file.

Sohini Mitra 1:06 PM
compositions --account-sid AC29f38c1506bf188995edc6c312c84d31

Media-Recordings bot APP 1:06 PM
Recent compositions for AC29f38c1506bf188995edc6c312c84d31 ▾

1	-----
2	Compositions for AC29f38c1506bf188995edc6c312c84d31
3	-----
4	CJa01d0459253b607873bb9e14d9c55565
5	CJ04ebc749dffff2f42c5d3d9ad41c19b4
6	CJ13f3c05f70ad246170cffccf7554b20a7
7	CJ5fe1f6ceebb75a2d178f37f623c58afa
8	-----

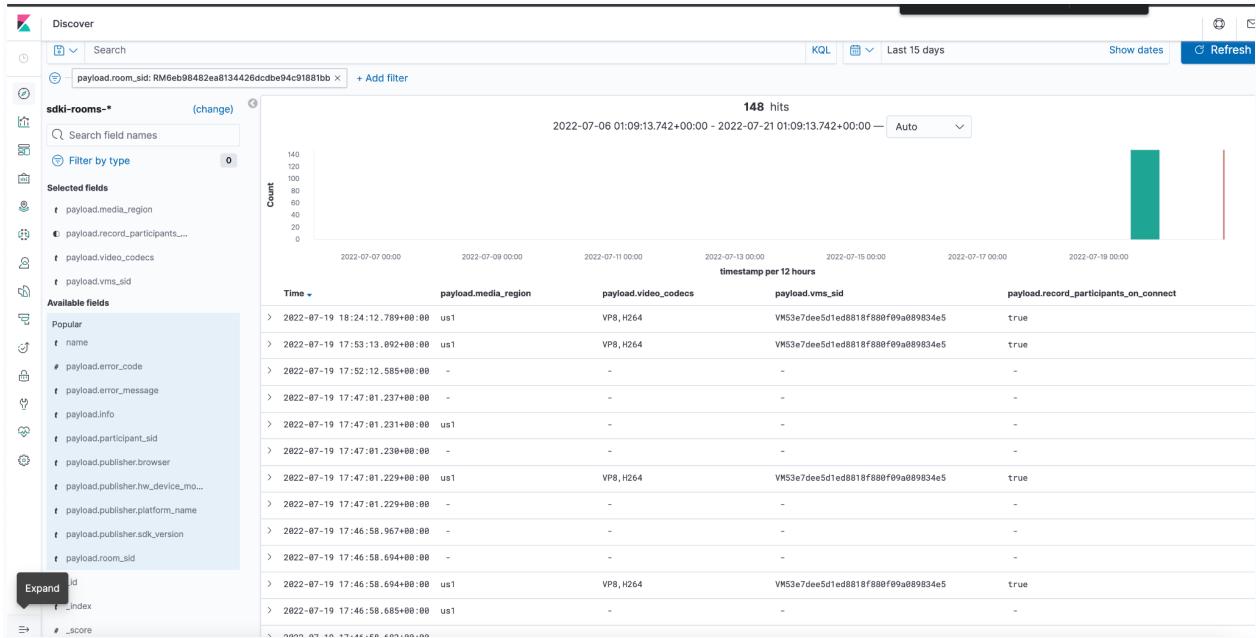
Sohini Mitra 1:09 PM
compositions --account-sid AC29f38c1506bf188995edc6c312c84d31 --room-sid RMfd9c3802a5051b87490a157f5ba6ea36

Media-Recordings bot APP 1:09 PM
Compositions for AC29f38c1506bf188995edc6c312c84d31 and room RMfd9c3802a5051b87490a157f5ba6ea36 ▾

1	-----
2	Compositions for room RMfd9c3802a5051b87490a157f5ba6ea36
3	-----
4	CJa01d0459253b607873bb9e14d9c55565
5	CJ04ebc749dffff2f42c5d3d9ad41c19b4
6	-----

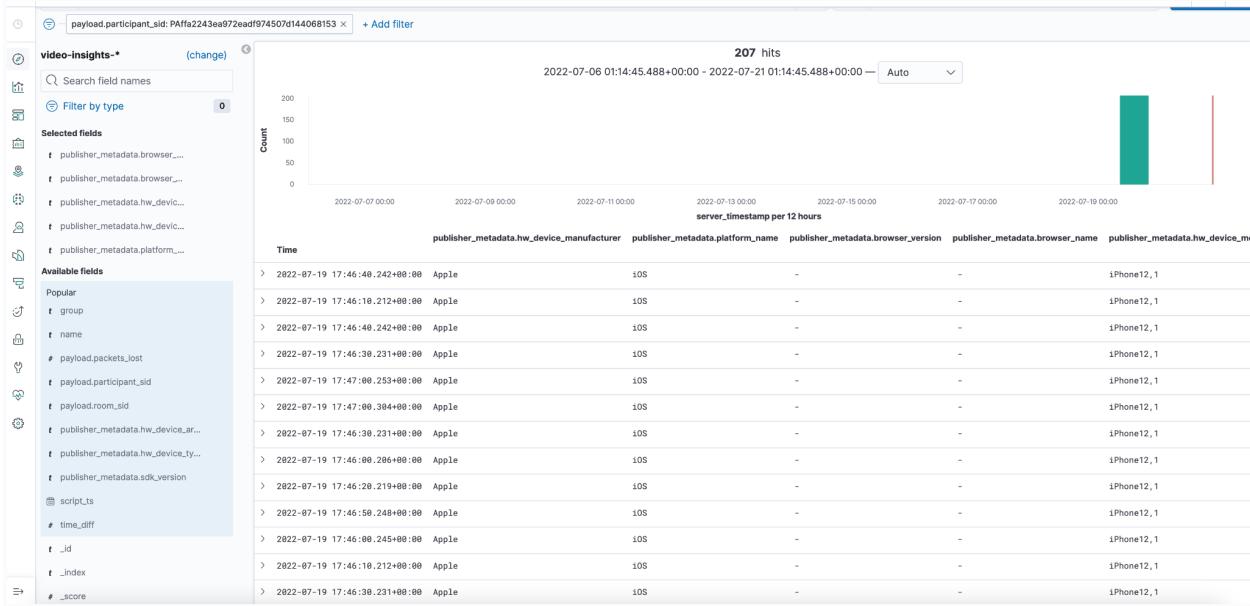
Kibana

- Ensure you have access to Kibana
<https://kibana.us1.eak.twilio.com/video/app/kibana#/discover>
- Change Index to sdki-rooms* create a "Discover URL" that displays media region, codecs, vms server sid, whether the participant was "recorded on connect" for all of the participants in one of your group rooms. Add shareable short URL for Kibana Link to Document and take a screenshot of using Kibana and add to GitHub Document.
- <https://kibana.us1.eak.twilio.com/video/goto/56c4e46d5f4aa85440349577adf17dbb>



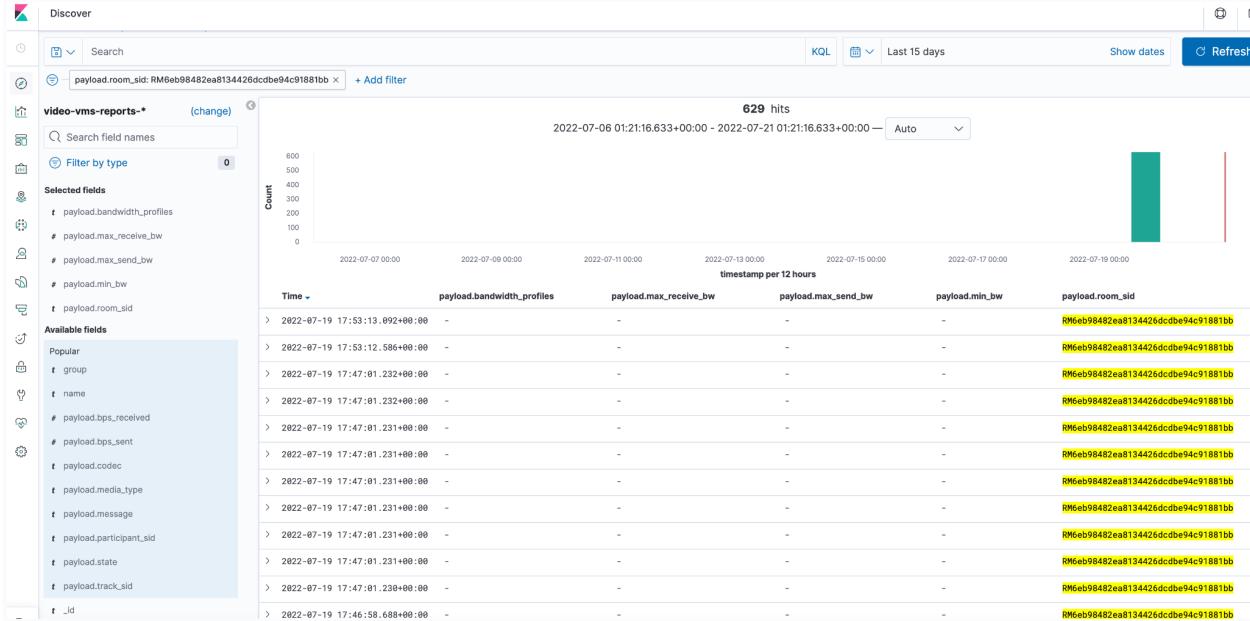
- Change Index to video-insights* create a "Discover URL" that displays all device, manufacture, OS, browser version, SDK version information for the participants. Add shareable short URL for Kibana Link to Document and take a screenshot of using Kibana and add to GitHub Document.

<https://kibana.us1.eak.twilio.com/video/goto/0c36e35e1cccaf7b2c160e25456d0b15>



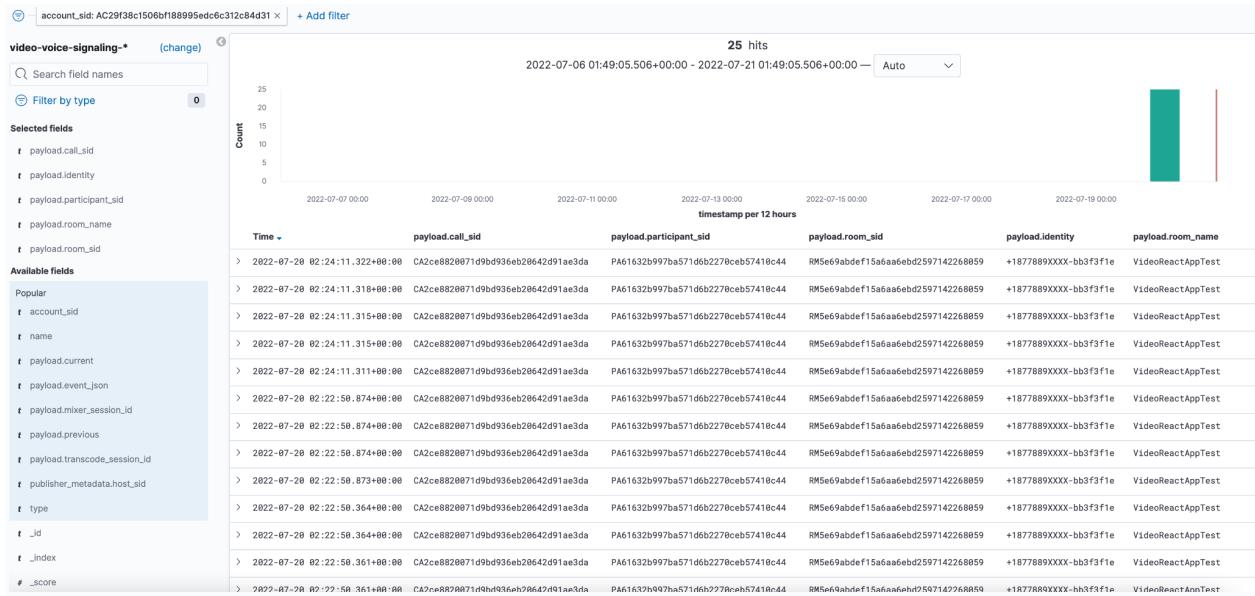
- Change Index to video-vms-reports* create a "Discover URL" that displays all actual bandwidth for each participant in one room. Add shareable short URL for Kibana Link to Document and take a screenshot of using Kibana and add to GitHub Document.

<https://kibana.us1.eak.twilio.com/video/goto/3d946aeed504666b3fd94e16d7dfda74>



- Change Index to video-voice-signaling* create a "Discover URL" that displays Participant SID, Call SID, Room SID, Identity, Room Name for the PSTN participant in one room. Add shareable short URL for Kibana Link to Document and take a screenshot of using Kibana and add to GitHub Document.

<https://kibana.us1.eak.twilio.com/video/goto/cc0f13ffa6ea94261c1bca2962680d38>



- Create a "Discover URL" that displays all the Network Quality scores for all participants in one of your group rooms. Add shareable short URL for Kibana Link to Document and take a screenshot of using Kibana and add to GitHub Document.

<https://kibana.us1.eak.twilio.com/video/qoto/a6d432c53850c262fe4449fed6fc3f3f>

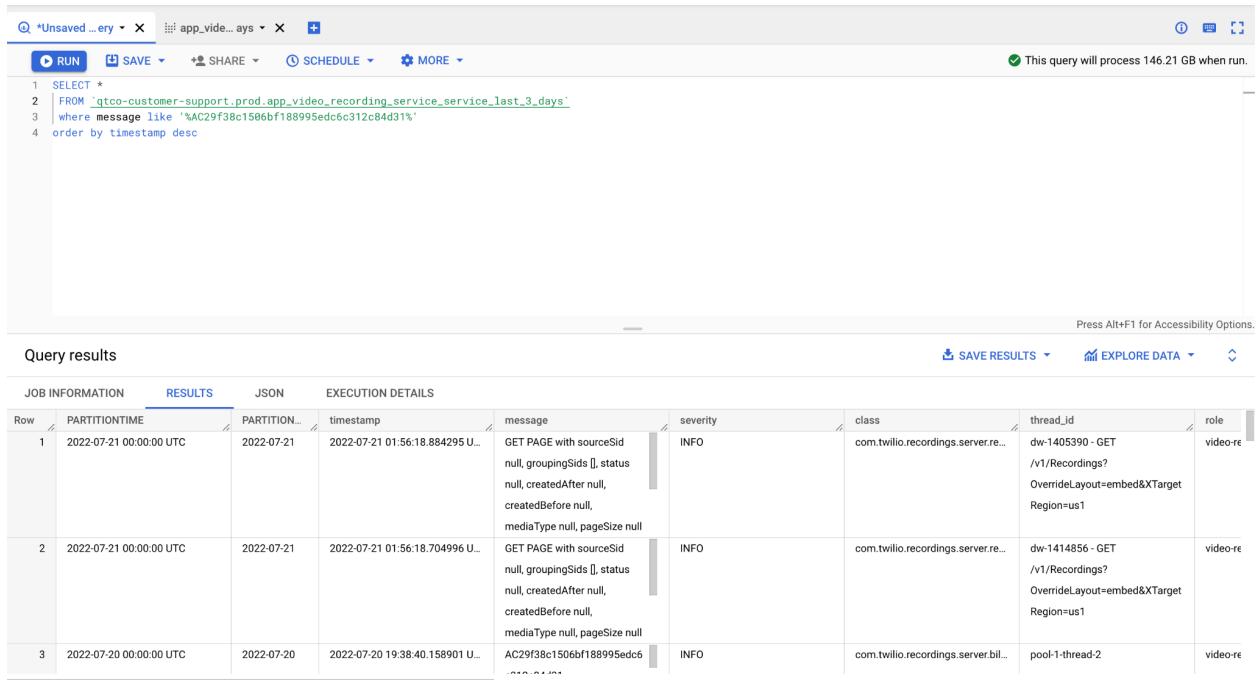
- Create visualization for the Network Quality Scores. Add shareable short URL for Kibana Link to file and take a screenshot of using Kibana and add to GitHub Document.

<https://kibana.us1.eak.twilio.com/video/goto/e428db9ee6e7a555fbf66d5b9113879c>

Quantico

- Ensure you have access to Quantico
<https://console.cloud.google.com/bigquery?project=qtco-customer-support>
- Using a SQL Query, find your API Request for the compositions you created. Add SQL Query and a shareable link to Google spreadsheet with the results from Quantico to a GitHub Document.

```
SELECT *  
  
FROM `qtco-customer-support.prod.app_video_recording_service_service_last_3_days`  
  
where message like '%AC29f38c1506bf188995edc6c312c84d31%'  
  
order by timestamp desc
```



The screenshot shows the Google BigQuery interface. At the top, there's a toolbar with buttons for RUN, SAVE, SHARE, SCHEDULE, and MORE. A note indicates that the query will process 146.21 GB when run. Below the toolbar, the query code is displayed:

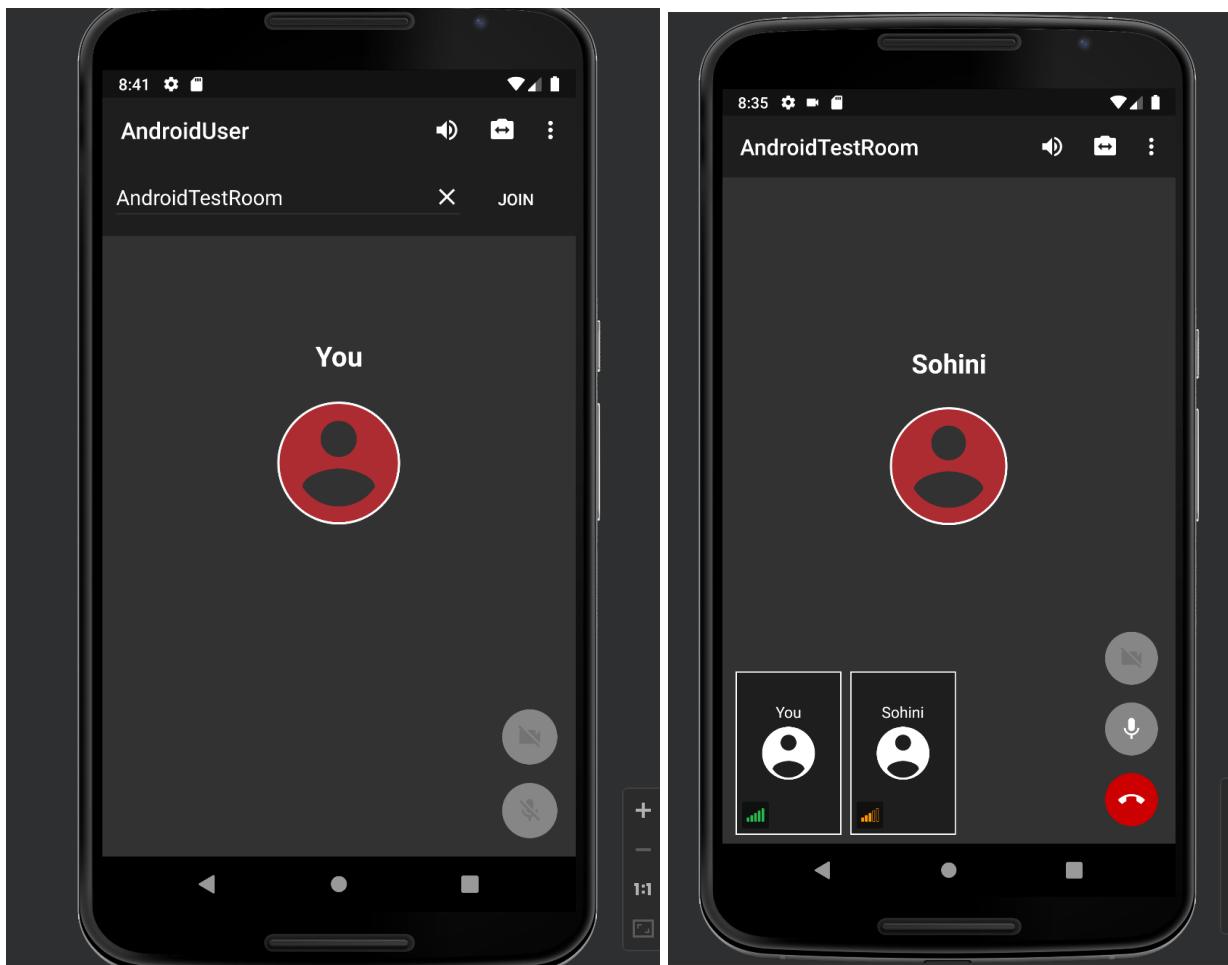
```
1 SELECT *  
2 FROM `qtco-customer-support.prod.app_video_recording_service_service_last_3_days`  
3 where message like '%AC29f38c1506bf188995edc6c312c84d31%'  
4 order by timestamp desc
```

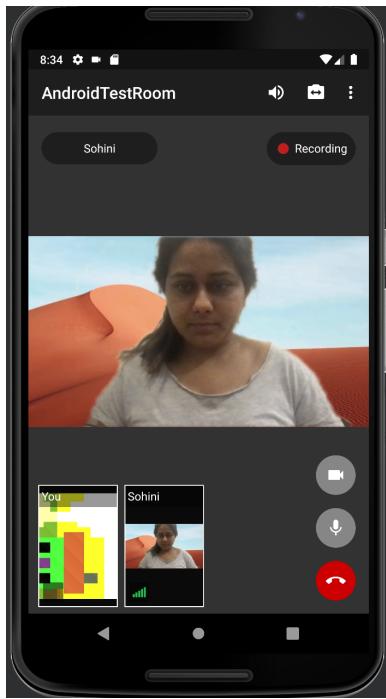
Below the code, a note says "Press Alt+F1 for Accessibility Options." Under "Query results", there are tabs for JOB INFORMATION, RESULTS (which is selected), JSON, and EXECUTION DETAILS. The results table has columns: Row, PARTITIONTIME, PARTITION_, timestamp, message, severity, class, thread_id, and role. The data shows three rows of results:

Row	PARTITIONTIME	PARTITION_	timestamp	message	severity	class	thread_id	role
1	2022-07-21 00:00:00 UTC	2022-07-21	2022-07-21 01:56:18.884295 U...	GET PAGE with sourceSid null, groupingSids [], status null, createdAfter null, createdBefore null, mediaType null, pageSize null	INFO	com.twilio.recordings.server.re...	dw-1405390 - GET /v1/Recordings? OverrideLayout=embed&XTarget Region=us1	video-re
2	2022-07-21 00:00:00 UTC	2022-07-21	2022-07-21 01:56:18.704996 U...	GET PAGE with sourceSid null, groupingSids [], status null, createdAfter null, createdBefore null, mediaType null, pageSize null	INFO	com.twilio.recordings.server.re...	dw-1414856 - GET /v1/Recordings? OverrideLayout=embed&XTarget Region=us1	video-re
3	2022-07-20 00:00:00 UTC	2022-07-20	2022-07-20 19:38:40.158901 U...	AC29f38c1506bf188995edc6...	INFO	com.twilio.recordings.server.bi...	pool-1-thread-2	video-re

Android Studio

- Take a screenshot with [Android Studio](#) set up and the <https://github.com/twilio/twilio-video-app-android> Android SDK Ahoy! Demo App loaded. Taking screenshots as you go, launch the emulator, join a group room, and join the room created by the android app from your browser using the <https://github.com/twilio/twilio-video-app-react> JS SDK Ahoy! Demo App. Post SIDs and screenshots to the GitHub Document.





XCode

- Take a screenshot with XCode set up and the <https://github.com/twilio/twilio-video-app-ios> iOS SDK Ahoy! Demo App loaded. Taking screenshots as you go, launch the emulator, join a group room, and join the room created by the iOS app from your browser using the <https://github.com/twilio/twilio-video-app-react> JS SDK Ahoy! Demo App. Post SIDs and screenshots to the GitHub Document.

