

EXAMPLES

- 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
 * @param {?number} maxVideoBitrate - Max video bitrate
 * @returns {CancelablePromise<Room>}
 */
function connectWithBandwidthConstraints(token, roomName) {
  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 are being updated
 * @param {?number} maxAudioBitrate - Max audio bitrate
 * @param {?number} maxVideoBitrate - Max video bitrate
 */
function updateBandwidthConstraints(room, maxAudioBitrate) {
  room.localParticipant.setParameters({
    maxAudioBitrate: maxAudioBitrate,
    maxVideoBitrate: maxVideoBitrate
  });
}

exports.connectWithBandwidthConstraints = connectWithBandwidthConstraints;
exports.updateBandwidthConstraints = updateBandwidthConstraints;

```

Participant's Media



Select Bandwidth Constraints

Max Audio Bitrate (bps)

16000

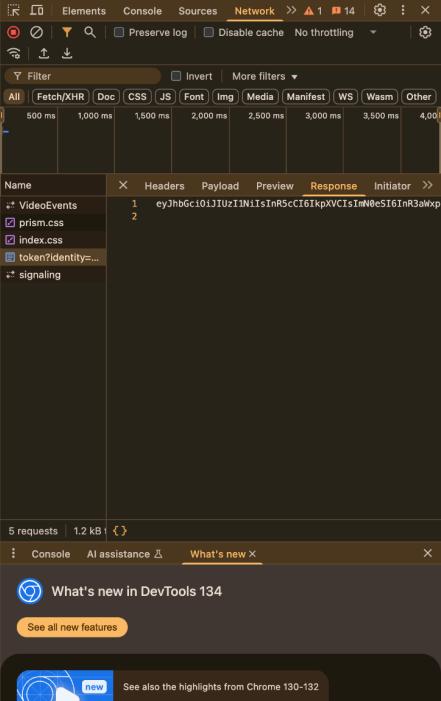


Max Video Bitrate (bps)

64000



Disconnect from Room



- Local Video Snapshot

localhost:3000/localvideosnapshot

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
 * in the HTMLCanvasElement.
 * @param {HTMLVideoElement} video
 * @param {LocalVideoTrack} localVideoTrack
 * @param {HTMLCanvasElement|HTMLImageElement} snapshot
 */
function takeLocalVideoSnapshot(video, localVideoTrack, if (window.ImageCapture) {
  const imageCapture = new ImageCapture(localVideoTrack);
  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

Snapshot

The screenshot shows a local video feed in a browser window. A blue button labeled "Take Snapshot" is visible below the video frame. To the right of the video, there is a developer tools Network tab showing network requests for the page, including files like "localvideosnapshot.js" and "index.css".

- Media device Selection

localhost:3000/mediadevices/

Media Device Selection

```
'* eslint-disable no-console */  
'use strict';

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

/**
 * Get the list of available media devices of the given kind
 * @param {Array<MediaDeviceInfo>} deviceInfos
 * @param {string} kind - One of 'audioinput', 'audiooutput'
 * @returns {Array<MediaDeviceInfo>} - Only those media devices
 */
function getDevicesOfKind(deviceInfos, kind) {
  return deviceInfos.filter(function(deviceInfo) {
    return deviceInfo.kind === kind;
  });
}

/**
 * Apply the selected audio output device.
 * @param {string} deviceId
 * @param {HTMLAudioElement} audio
 * @returns {Promise<void>}
 */
function applyAudioOutputDeviceSelection(deviceId, audio) {
  return typeof audio.setSinkId === 'function'
    ? audio.setSinkId(deviceId)
    : Promise.reject('This browser does not support setting audio sink');
}

/**
 * Apply the selected input device.
 * @param {string} deviceId
 * @param {LocalTrack} localTrack - LocalAudioTrack or LocalVideoTrack
 * @param {'audio' | 'video'} kind
 * @returns {Promise<LocalTrack>} - The created or restarted track
 */
function applyInputDeviceSelection(deviceId, localTrack, constraints = { deviceid: { exact: deviceId } }) {
  if (!localTrack) {
    return localTrack.restart(constraints).then(function() {
      return localTrack;
    });
  }
}
```

Preview Media

Select Media Devices

Audio Input

MacBook Pro Microphone (Built-in)

Apply

Video Input

FaceTime HD Camera (D288:CE50)

Apply

Audio Output

MacBook Pro Speakers (Built-in)

Apply

Connect

The screenshot shows a media device selection interface. It includes dropdown menus for selecting audio input (MacBook Pro Microphone), video input (FaceTime HD Camera), and audio output (MacBook Pro Speakers). A "Connect" button is at the bottom. To the right, there is a developer tools Network tab showing network requests for the page, including files like "mediadevices.js" and "index.css".

- Codec Preference

The screenshot shows a browser window at localhost:3000/codecpreferences/. On the left, a code editor displays a JavaScript file for 'Codec Preferences' using the 'twilio-video' library. It includes functions for connecting with preferred codecs and creating local video tracks. In the center, a video player titled 'Participant's Media' shows a person's face. Below it, a section titled 'Select Codec Preferences' allows setting 'Preferred Audio Codec' (PCMA) and 'Preferred Video Codec' (H264). A blue button labeled 'Disconnect from Room' is visible. To the right, the Chrome DevTools Network tab is open, showing network requests for 'VideoEvents' and 'token?identity=...'.

- Share your Screen

The screenshot shows a browser window at localhost:3000/screenshare/. On the left, a code editor displays a JavaScript file for 'Share Your Screen' using the 'twilio-video' library. It includes a function for creating a local video track from the user's screen. In the center, a video player titled 'Local Screen' shows a screenshot of the browser interface. Below it, a section titled 'Remote Screen' shows another screenshot of the browser interface. A blue button labeled 'Stop Screen Capture' is visible. To the right, the Chrome DevTools Network tab is open, showing network requests for various files like 'bootstrap.min.css', 'jquery-3.4.1.slim.min.js', and 'popper.min.js', along with 'VideoEvents' and 'token?identity=' requests.

- Dominant Screen

The screenshot shows a browser window at localhost:3000/dominantspeaker/. The page displays a Twilio Video API code snippet for connecting to a room with a dominant speaker. Below the code, a message says: "Please join the above Room using the [QuickStart App](#) or connect the below Participants to the Room and unmute only one of them." It lists participants Alice, Charlie, Mak, and Bob, each with a "Disconnect" and "Mute" button. Below the participant list are four video thumbnails: Alice (highlighted with a red border), Charlie, Mak, and Bob. The Network tab in the DevTools panel shows network requests for the page resources.

- Reconnection

The screenshot shows a browser window at localhost:3000/reconnection/. The page displays a Twilio Video SDK code snippet for handling reconnection events. It includes a "Reconnection States and Events" section and a "Room state is:" section. The "Room state is:" section shows three states: "Connected" (green background), "Disconnected" (white background), and "Reconnecting" (light blue background). A "Create Room" button is present. Below the states, a text block explains the reconnection process and a "Leave Room" button is shown. The Network tab in the DevTools panel shows network requests for the page resources.

- Network Quality

The screenshot shows a web application interface for monitoring network quality. On the left, there is a code editor displaying JavaScript code related to network quality reporting and participant management. On the right, a list of participants (Alice, Bob, Charlie, Mak) is shown with their current network quality levels. Below each participant is a small video thumbnail and a 'Disconnect' button. A 'NQ Level (You):' section at the bottom shows a video thumbnail of the user and a 'NQ Stats' box. To the right of the participant list is a detailed network timeline showing various requests and their performance metrics.

- Local Media controls

The screenshot displays a web application for managing local media controls. On the left, a code editor shows JavaScript functions for enabling and disabling audio and video tracks. On the right, a 'Local Media Controls' section features 'Disable Audio' and 'Disable Video' buttons. Below this is a 'RemoteParticipant View' section showing a video thumbnail of a participant. A network timeline on the far right tracks the performance of various resources and requests.

- **Datatracks**

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}
 * @param {string} message - The message to be sent
 */
function sendChatMessage(dataTrack, message) {
  dataTrack.send(message);
}

/*
 * Receive chat messages from RemoteParticipants in the 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 === message.trackName) {
        publication.track.on('message', function(msg) {
          onMessageReceived(msg);
        });
      }
    });
  });
}

function receiveChatMessages(room, onMessageReceived) {
  room.participants.forEach(function(participant) {
    participant.dataTracks.forEach(function(publication) {
      if (publication.isSubscribed && publication.trackName === message.trackName) {
        publication.track.on('message', function(msg) {
          onMessageReceived(msg);
        });
      }
    });
  });
}
```

P1 **P2**

P2: has connected
P2: hi
P1: hola

Message Message

Submit Submit

Network Tab (DevTools)

| Name | Status | Type | Initiator | Size | Time |
|-------------------------------|--------|------------|----------------|------------|---------|
| videoEvents | 101 | webso... | index.js:25/24 | U B | Pending |
| datatracks/ | 200 | docum... | datatracks | 266 B | 3 ms |
| datatracks | 304 | docum... | Other | (disk c... | Pending |
| jquery-3.4.1.slim.min.js | 200 | script | datatracks/277 | (memo... | 0 ms |
| popper.min.js | 200 | script | datatracks/66 | (memo... | 0 ms |
| bootstrap.min.js | 200 | script | datatracks/278 | (memo... | 0 ms |
| bootstrap.min.css | 200 | stylesheet | datatracks/6 | (disk c... | 3 ms |
| prism.css | 304 | stylesheet | datatracks/7 | 265 B | 4 ms |
| index.css | 304 | stylesheet | datatracks/8 | 265 B | 4 ms |
| index.js | 304 | script | datatracks/64 | 268 B | 5 ms |
| css?family=Roboto+Mono:3... | 200 | stylesheet | index.css:1 | (memo... | 0 ms |
| helpers.js | 304 | fetch | index.js:343 | 265 B | 2 ms |
| LuxDF-4xIVMF-BR8bXMH... | 200 | font | shared.js:28 | (memo... | 0 ms |
| data:image/svg+xml;... | 200 | svg+xml | shared.js:28 | (memo... | 0 ms |
| token?identity=Creative%20... | 200 | fetch | index.js:327 | 654 B | 3 ms |
| token?identity=Kindly%20SS... | 200 | fetch | index.js:327 | 650 B | 3 ms |
| +> signaling | 101 | webso... | index.js:23417 | 0 B | Pending |
| +> signaling | 101 | webso... | index.js:23417 | 0 B | Pending |
| +> VideoEvents | 101 | webso... | index.js:25784 | 0 B | Pending |

19 requests | 2.6 kB transferred | 1.7 MB resources | Finish: 2.83 s | DOMContentLoaded

Console AI assistance What's new X

What's new in DevTools 134

See all new features

Cloudinary icon See also the highlights from Chrome 130-132

- **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',
        clientTrackSwitchOffControl: 'manual'
      }
    }
  });
}

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

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

/*
 * Set the encoder dimensions of the RemoteVideoTrack
 */

```

Remote Video Controls

Switch Off Control: **Switch On** **Switch Off**

Content Preferences: **1280x720**

Remote Video Track

Video Bitrate

On

2.0k

18:20:00 18:21:00 18:22:00 18:23:00 18:24:00

Network Tab (DevTools)

| Name | Status | Type | Initiator | Size | Time |
|------------------------------|--------|------------|------------------|------------|---------|
| videoEvents | 101 | webso... | index.js:204/25 | U B | Pending |
| manualrenderhin/ | 200 | docum... | manualrenderhin | 4.0 kB | 4 ms |
| manualrenderhin | 200 | docum... | Other | (disk c... | Pending |
| bootstrap.min.css | 200 | stylesheet | manualrenderhin | (memo... | 0 ms |
| jquery-3.4.1.slim.min.js | 200 | script | manualrenderhin | (memo... | 0 ms |
| popper.min.js | 200 | script | manualrenderhin | (memo... | 0 ms |
| bootstrap.min.js | 200 | script | manualrenderhin | (memo... | 0 ms |
| all.css | 200 | stylesheet | manualrenderhin | (disk c... | 8 ms |
| prism.css | 200 | stylesheet | manualrenderhin | 2.3 kB | 8 ms |
| index.css | 200 | stylesheet | manualrenderhin | 1.9 kB | 8 ms |
| index.js | 200 | script | manualrenderhin | 1.4 kB | 18 ms |
| css?family=Roboto+Mono:3... | 200 | stylesheet | datatracks/index | (memo... | 0 ms |
| fa-solid-900.wof2 | 200 | font | all.css | (memo... | 0 ms |
| LuxDF-4xIVMF-BR8bXMH... | 200 | font | css | (memo... | 0 ms |
| helpers.js | 200 | fetch | index.js:235 | 1.8 kB | 2 ms |
| token?identity=Lucky%20Ta... | 200 | fetch | index.js:219 | 649 B | 1 ms |
| token?identity=Tall%20Dan... | 200 | fetch | index.js:219 | 650 B | 1 ms |
| +> signaling | 101 | webso... | index.js:24058 | 0 B | Pending |
| +> signaling | 101 | webso... | index.js:24058 | 0 B | Pending |

20 requests | 1.4 MB transferred | 1.8 MB resources | Finish: 84 ms | DOMContentLoaded

Console AI assistance What's new X

What's new in DevTools 134

See all new features

Cloudinary icon See also the highlights from Chrome 130-132

- Video Track Automatic Controls

localhost:3000/autorenderhint/

Video Track Automatic Controls

```
'use strict';

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

/**
 * Connect to a Room with 'auto' mode. This is the default
 * @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: 176x144

Remote Video Track

On

Video Bitrate

Network

Fetch/XHR Doc CSS JS Font Img Media Manifest WS

| Name | Status | Type | Initiator | Size |
|------------------------------|--------|------------|------------------|------------|
| autorenderhint/ | 200 | document | autorenderhint | 4 |
| autorenderhint | 200 | document | Other | (disabled) |
| bootstrap.min.css | 200 | stylesheet | autorenderhint/ | (media) |
| all.css | 200 | stylesheet | autorenderhint/ | (media) |
| jquery-3.4.1.slim.min.js | 200 | script | autorenderhint/ | (media) |
| popper.min.js | 200 | script | autorenderhint/ | (media) |
| bootstrap.min.js | 200 | script | autorenderhint/ | (media) |
| prism.css | 200 | stylesheet | autorenderhint/ | 2 |
| index.css | 200 | stylesheet | autorenderhint/ | 2 |
| index.js | 200 | script | autorenderhint/ | 1.4 |
| css?family=Roboto+Mono:3... | 200 | stylesheet | datatracks/index | (media) |
| fa-solid-900.woff2 | 200 | font | all.css | (media) |
| L0xuDF4xIVMF-BfR8bXMh... | 200 | font | css | (media) |
| helpers.js | 200 | fetch | index.js:207 | 8 |
| token?identity=Golden%20E... | 200 | fetch | index.js:191 | 6 |
| token?identity=Tall%20Cam... | 200 | fetch | index.js:191 | 6 |
| +* signaling | 101 | websocket | index.js:24030 | |
| +* signaling | 101 | websocket | index.js:24030 | |
| +* VideoEvents | 101 | websocket | index.js:26397 | |

20 requests | 1.4 MB transferred | 1.8 MB resources | Finish: 85 ms | DOMContentLoaded: 18:23:50

Console AI assistance What's new

What's new in DevTools 134