WebRTC Statistics API

Draft
This page is not complete.

The WebRTC API has a vast array of statistics available, covering the entire breadth of the WebRTC connectivity system, from sender to receiver and peer to peer.

Collecting statistics

You can collect statistics at various levels throughout the WebRTC hierarchy of objects. Most broadly, you can call `getStats()` on an `RTCPeerConnection` to get statistics for the connection overall. In this example, a new `RTCPeerConnection` is created, and then `setInterval()` is used to set the function `getConnectionStats()` to be called every second.

That function, in turn, uses `getStats()` to obtain statistics for the connection and to make use of that data.

```
try {
  myPeerConnection = new RTCPeerConnection(pcOptions);

  statsInterval = window.setInterval(getConnectionStats, 1000);
  /* add event handlers, etc */
} catch(err) {
  console.error("Error creating RTCPeerConnection: " + err);
}

function getConnectionStats() {
  myPeerConnection.getStats(null).then(stats => {
    var statsOutput = "";
```

```
    stats.forEach(report => {
      if (report.type === "inbound-rtp" && report.kind === "video") {
        Object.keys(report).forEach(statName => {
          statsOutput += `<strong>${statName}:</strong> ${report[statN
        });
      }
    });

    document.querySelector(".stats-box").innerHTML = statsOutput;
  });
}
```

When the promise returned by `getStats()` is fulfilled, the resolution handler receives as input an `RTCStatsReport` object containing the statistics information. This object contains a `Map` of named dictionaries based on `RTCStats` and its affiliated types.

This example specifically looks for the report whose `type` is `inbound-rtp` and whose `kind` is `video`. This way, we look only at the video-related statistics for the local `RTCRtpReceiver` responsible for receiving the streamed media.

Commonly used statistics

Reference

The `RTCStatsReport` object contains a map of named objects based one of the `RTCStats` dictionary's subclasses. Upon looking up a statistic category by name, you get an object containing the corresponding data. The table below shows the statistic categories and the corresponding dictionaries; for each statistic category, the full hierarchy of `RTCStats`-based dictionaries are listed, so you can easily find all the available values.

Mapping of statistic category names to the dictionaries they implement

Statistic category name (`RTCStatsType`)	Description	Dictionaries implemented

Statistic category name (`RTCStatsType`)	Description	Dictionaries implemented
`candidate-pair`	Statistics describing the change from one `RTCIceTransport` to another, such as during an ICE restart.	`RTCIceCandidatePairStats` └ `RTCStats`
`certificate`	Statistics about a certificate being used by an `RTCIceTransport`.	`RTCCertificateStats` └ `RTCStats`
`codec`	Statistics about a specific codec being used by streams being sent or received by this connection.	`RTCCodecStats` `RTCStats`
`csrc`	Statistics for a single contributing source (CSRC) that contributed to one of the connection's inbound RTP streams.	`RTCRtpContributingSourceStats` └ `RTCStats`
`data-channel`	Statistics related to one `RTCDataChannel` on the connection.	`RTCDataChannelStats` └ `RTCStats`
`ice-server`	Statistics about the connection to an ICE server (STUN or TURN.	`RTCIceServerStats` └ `RTCStats`

`inbound-rtp`	Statistics describing the state of one of the connection's inbound data streams.	`RTCInboundRtpStreamStats`
		└ `RTCReceivedRtpStreamStats`
		  └ `RTCRtpStreamStats`
		    └ `RTCStats`

Statistic category name (`RTCStatsType`)	Description	Dictionaries implemented
`local-candidate`	Statistics about a local ICE candidate associated with the connection's `RTCIceTransport`s.	`RTCIceCandidateStats` └ `RTCStats`
`media-source`	Statistics about the media produced by the `MediaStreamTrack` attached to an RTP sender. The dictionary this key maps to depends on the track's `kind`.	`RTCAudioSourceStats` *or* `RTCVideoSourceStats` └ `RTCMediaSourceStats` └ `RTCStats` `RTCAudioSourceStats` *or* `RTCVideoSourceStats`

`outbound-rtp`	Statistics describing the state of one of the outbound data streams on this connection.	`RTCOutboundRtpStreamStats`
		└ `RTCSentRtpStreamStats`
		  └ `RTCRtpStreamStats`
		    └ `RTCStats`

`peer-connection`	Statistics describing the state of the `RTCPeerConnection`.	`RTCPeerConnectionStats` └ `RTCStats`

`receiver`	Statistics related to a specific `RTCRtpReceiver` and the media associated with that receiver. The specific type of object representing the statistics depends on the media `kind`.	`RTCAudioReceiverStats` *or* `RTCVideoReceiverStats`
		└ `RTCAudioHandlerStats` *or* `RTCVideoHandlerStats`
		  └ `RTCMediaHandlerStats`
		    └ `RTCStats`

Statistic category name (`RTCStatsType`)	Description	Dictionaries implemented
`remote-candidate`	Statistics about a remote ICE candidate associated with the connection's `RTCIceTransport`s.	`RTCIceCandidateStats` └ `RTCStats`

`remote-inbound-rtp`	Statistics describing the state of the inbound data stream from the perspective of the remote peer.	`RTCRemoteInboundRtpStreamStats`
		└ `RTCReceivedRtpStreamStats`
		  └ `RTCRtpStreamStats`
		    └ `RTCStats`

`remote-outbound-rtp`	Statistics describing the state of the outbound data stream from the perpsective of the remote peer.	`RTCRemoteOutboundRtpStreamStats`
		└ `RTCSentRtpStreamStats`
		  └ `RTCRtpStreamStats`
		    └ `RTCStats`

`sctp-transport`	Statistics about an `RTCSctpTransport`.	`RTCSctpTransportStats` └ `RTCStats`

`sender`	Statistics related to a specific `RTCRtpSender` and the media associated with that sender. The type of object representing this statistic depends on the `kind` of the media.	`RTCAudioSenderStats` *or* `RTCVideoSenderStats`
		└ `RTCAudioHandlerStats` *or* `RTCVideoHandlerStats`
		  └ `RTCMediaHandlerStats`
		    └ `RTCStats`

Statistic category name (`RTCStatsType`)	Description	Dictionaries implemented
`stream` 🗑	Statistics about a particular media `MediaStream`. This has been obsoleted since the transition to WebRTC being track-based rather than stream-based.	`RTCMediaStreamStats` └ `RTCStats`
`track` 🗑	Statistics related to a specific `MediaStreamTrack`'s attachment to one of the connection's senders or receivers. The referenced object's type depends on the track type. These statistics have all been moved to `media-source`, `sender`, `receiver`, `outbound-rtp`, and `inbound-rtp`, and this statistic category type is thus obsolete and shouldn't be used anymore.	`RTCSenderVideoTrackAttachmentStats` *or* `RTCSenderAudioTrackAttachmentStats` *or* `RTCReceiverVideoTrackAttachmentStats` *or* `RTCReceiverAudioTrackAttachmentStats` └ `RTCMediaHandlerStats` └ `RTCStats`
`transceiver`	Statistics related to a specific `RTCRtpTransceiver`.	`RTCRtpTransceiverStats` └ `RTCStats`

Statistic category name (`RTCStatsType`)	Description	Dictionaries implemented
`transport`	Statistics about a transport used by the connection.	`RTCTransportStats` └ `RTCStats`

Specifications

Specification	Status	Comment
Identifiers for WebRTC's Statistics API The definition of 'WebRTC statistics types' in that specification.	CR Candidate Recommendation	Compatibility for individual statistic types
WebRTC 1.0: Real-time Communication Between Browsers The definition of 'RTCStatsReport' in that specification.	CR Candidate Recommendation	Compatibility of statistic reporting

Browser compatibility

No compatibility data found. Please contribute data for "api.RTCStatsType" (depth: 1) to the MDN compatibility data repository.

Last modified: Jul 13, 2020, by MDN contributors

- Introduction to WebRTC protocols
- WebRTC connectivity
- Lifetime of a WebRTC session
- Signaling and video calling
- Using WebRTC data channels
- Using DTMF with WebRTC
- Improving compatibility using WebRTC adapter.js
- Taking still photos with WebRTC
- A simple RTCDataChannel sample

▼ Interfaces

- `RTCPeerConnection`
- `RTCSessionDescription`
- `RTCIceCandidate`
- `RTCPeerConnectionIceEvent`
- `RTCPeerConnectionIceErrorEvent`
- `RTCCertificate`
- `RTCRtpSender`
- `RTCRtpReceiver`
- `RTCRtpTransceiver`
- `RTCDtlsTransport`
- `RTCIceTransport`
- `RTCTrackEvent`
- `RTCSctpTransport`
- `RTCDataChannel`
- `RTCDataChannelEvent`
- `RTCDTMFSender`
- `RTCDTMFToneChangeEvent`
- `RTCStatsReport`
- `RTCStatsEvent`
- `RTCErrorEvent`

▼ Properties

- `Navigator.mediaDevices`

▼ Methods

`MediaDevices.getUserMedia()`