RTCIceCandidatePairStats

The WebRTC **`RTCIceCandidatePairStats`** dictionary reports statistics which provide insight into the quality and performance of an `RTCPeerConnection` while connected and configured as described by the specified pair of ICE candidates.

If a `RTCStats`-based object's `type` is `candidate-pair`, it's an `RTCIceCandidatePairStats` object.

Properties

`RTCIceCandidatePairStats` is based upon `RTCStats` and inherits its properties. In addition, it adds the following new properties:

`availableIncomingBitrate` Optional

Provides an informative value representing the available inbound capacity of the network by reporting the total number of bits per second available for all of the candidate pair's incoming RTP streams. This does not take into account the size of the IP overhead, nor any other transport layers such as TCP or UDP.

`availableOutgoingBitrate` Optional

Provides an informative value representing the available outbound capacity of the network by reporting the total number of bits per second available for all of the candidate pair's outoing RTP streams. This does not take into account the size of the IP overhead, nor any other transport layers such as TCP or UDP.

`bytesReceieved` Optional

The total number of payload bytes received (that is, the total number of bytes received minus any headers, padding, or other administrative overhead) on this candidate pair so far.

`bytesSent` Optional

The total number of payload bytes sent (that is, the total number of bytes sent minus any headers, padding, or other administrative overhead) so far on this candidate pair.

`circuitBreakerTriggerCount` Optional

An integer value indicating the number of times the circuit-breaker has been triggered for this particular 5-tuple (the set of five values comprising a TCP connection: source IP address, source port number, destination IP address, destination port number, and protocol). The circuit breaker is triggered whenever a connection times out or otherwise needs to be automatically aborted.

`consentExpiredTimestamp` Optional

A `DOMHighResTimeStamp` value indicating the time at which the most recent STUN binding response expired. This value is `undefined` if no valid STUN binding responses have been sent on the candidate pair; this can only happen if `responsesReceived` is 0.

`consentRequestsSent` Optional

The total number of consent requests that have been sent on this candidate pair.

`currentRoundTripTime` Optional

A floating-point value indicating the total time, in seconds, that elapsed elapsed between the most recently-sent STUN request and the response being received. This may be based upon requests that were involved in confirming permission to open the connection.

`firstRequestTimestamp` Optional

A `DOMHighResTimeStamp` value which specifies the time at which the first STUN request was sent from the local peer to the remote peer for this candidate pair.

`lastPacketReceivedTimestamp` Optional

A `DOMHighResTimeStamp` value indicating the time at which the last packet was received by the local peer from the remote peer for this candidate pair. Timestamps are not recorded for STUN packets.

`lastPacketSentTimestamp` Optional

A `DOMHighResTimeStamp` value indicating the time at which the last packet was sent from the local peer to the remote peer for this candidate pair. Timestamps are not recorded for STUN packets.

`lastRequestTimestamp` Optional

A `DOMHighResTimeStamp` value which specifies the time at which the last (most recent) STUN request was sent from the local peer to the remote peer for this candidate pair.

`lastResponseTimestamp` Optional

A `DOMHighResTimeStamp` value that specifies the time at which the last (most recent) STUN response was received by the local candidate from the remote candidate in this pair.

`localCandidateId` Optional

The unique ID string corresponding to the `RTCIceCandidate` from the data included in the `RTCIceCandidateStats` object providing statistics for the candidate pair's local candidate.

`nominated` Optional

A Boolean value which, if `true`, indicates that the candidate pair described by this object is one which has been proposed for use, and will be (or was) used if its priority is the highest among the nominated candidate pairs. See RFC 5245, section 7.1.3.2.4 for details.

`packetsReceived` Optional

The total number of packets received on this candidate pair.

`packetsSent` Optional

The total number of packets sent on this candidate pair.

`remoteCandidateId` Optional

The unique ID string corresponding to the remote candidate from which data was taken to construct the `RTCIceCandidateStats` object describing the remote end of the connection.

`requestsReceived` Optional

The total number of connectivity check requests that have been received, including retransmissions. This value includes both connectivity checks and STUN consent checks.

`requestsSent` Optional

The total number of connectivity check requests that have been sent, *not* including retransmissions.

`responsesReceived` Optional

The total number of connectivity check responses that have been received.

`responsesSent` Optional

The total number of connectivity check responses that have been sent. This includes both connectivity check requests and STUN consent requests.

`retransmissionsReceived` Optional

The total number of times connectivity check request retransmissions were received. A retransmission is a connectivity check request whose `TRANSACTION_TRANSMIT_COUNTER` attribute's `req` field is greater than 1.

`retransmissionsSent` Optional

The total number of times connectivity check request retransmissions were sent.

`state` Optional

A `RTCStatsIceCandidatePairState` object which indicates the state of the connection between the two candidates.

`totalRoundTripTime` Optional

A floating-point value indicating the total time, in seconds, that has elapsed between sending STUN requests and receiving responses to them, for all such requests made to date on this candidate pair. This includes botyh connectivity check and consent check requests. You can compute the average round trip time (RTT) by dividing this value by `responsesReceived`.

`transportId` Optional

A `DOMString` that uniquely identifies the `RTCIceTransport` that was inspected to obtain the transport-related statistics (as found in `RTCTransportStats`) used in generating this object.

Obsolete properties

The following properties have been removed from the specification and should no longer be used. You should update any existing code to avoid using them as soon as is practical. Check the compatibility table for details on which browsers support them and in which versions.

`priority` Optional

An integer value indicating the candidate pair's priority.

`readable` Optional

A Boolean value indicating whether or not data can be sent over the connection described by the candidate pair.

`writable` Optional

A Boolean value indicating whether or not data can be received on the connection described by the candidate pair.

Non-standard properties

`selected` ⚠ Optional

A Firefox-specific Boolean value which is `true` if the candidate pair described by this object is the one currently in use. The spec-compliant way to determine the selected candidate pair is to look for a stats object of type `transport`, which is an `RTCTransportStats` object. That object's `selectedCandidatePairId` property indicates whether or not the specified transport is the one being used.

Usage notes

The currently-active ICE candidate pair—if any—can be obtained by calling the `RTCIceTransport` method `getSelectedCandidatePair()`, which returns an `RTCIceCandidatePair` object, or `null` if there isn't a pair selected. The active candidate pair describes the current configuration of the two ends of the `RTCPeerConnection`.

Any candidate pair that isn't the active pair of candidates for a transport gets deleted if the `RTCIceTransport` performs an ICE restart, at which point the `state` of the ICE transport returns to `new` and negotiation starts once again. For more information, see ICE restart in Lifetime of a WebRTC session.

Example

This example computes the average time elapsed between connectivity checks if the `RTCStats` object `rtcStats` is an `RTCIceCandidatePairStats` object.

```
if (rtcStats && rtcStats.type === "candidate-pair") {
  let elapsed = (rtcStats.lastRequestTimestamp - rtcStats.firstReques
                 / rtcStats.requestsSent;

  log("Average time between ICE connectivity checks: " + elapsed + "
}
```

The code begins by looking at `rtcStats` to see if its `type` is `candidate-pair`. If it is, then we know that `rtcStats` is in fact an `RTCIceCandidatePairStats` object. If so, we compute the average time elapsed between STUN connectivity checks and log that information.

Specifications

Specification	Status	Comment
Identifiers for WebRTC's Statistics API The definition of 'RTCIceCandidatePairStats' in that specification.	CR Candidate Recommendation	Initial specification.

Browser compatibility

Update compatibility data on GitHub

`RTCIceCandidatePairStats`	
Chrome	56
Edge	≤79
Firefox	29
IE	No
Opera	?
Safari	?
WebView Android	56
Chrome Android	56
Firefox Android	29
Opera Android	?
Safari iOS	?
Samsung Internet Android	6.0

`availableIncomingBitrate`	
Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`availableOutgoingBitrate`	
Chrome	Yes
Edge	≤79
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	Yes
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

`bytesReceived`	
Chrome	Yes

Edge	≤79
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	Yes
Firefox Android	56
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

`bytesSent`

Chrome	Yes
Edge	≤79
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	Yes
Firefox Android	56
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

`circuitBreakerTriggerCount`

Chrome	No
Edge	No
Firefox	No

IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`consentExpiredTimestamp`

Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`consentRequestsSent`

Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?

Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`currentRoundTripTime`

Chrome	71
Edge	≤79
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	71
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	10.0

`firstRequestTimeStamp`

Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No

Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`lastPacketReceivedTimestamp`

Chrome	No
Edge	No
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	56
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`lastPacketSentTimestamp`

Chrome	No
Edge	No
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	56

Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`lastReponseTimestamp`

Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`lastRequestTimestamp`

Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?

Samsung Internet Android	No

`localCandidateId`

Chrome	No
Edge	No
Firefox	29
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	29
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`nominated`

Chrome	No
Edge	No
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	56
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`packetsReceived`

Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`packetsSent`

Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`priority` ⚠

Chrome	No
Edge	No

Firefox	42
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	42
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`readable` ⚠

Chrome	No
Edge	No
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	56
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`remoteCandidateId`

Chrome	No
Edge	No
Firefox	29
IE	No

Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	29
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`requestsReceived`

Chrome	Yes
Edge	≤79
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	Yes
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

`requestsSent`

Chrome	Yes
Edge	≤79
Firefox	No
IE	No
Opera	?
Safari	?

WebView Android	Yes
Chrome Android	Yes
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

`responsesReceived`

Chrome	Yes
Edge	≤79
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	Yes
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes

`responsesSent`

Chrome	Yes
Edge	≤79
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	Yes

Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes
`retransmissionsReceived`	
Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	No
`retransmissionsSent`	
Chrome	No
Edge	No
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	?

Safari iOS	?
Samsung Internet Android	No
`state`	
Chrome	No
Edge	No
Firefox	29
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	29
Opera Android	?
Safari iOS	?
Samsung Internet Android	No
`totalRoundTripTime`	
Chrome	71
Edge	≤79
Firefox	No
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	71
Firefox Android	No
Opera Android	?
Safari iOS	?
Samsung Internet Android	10.0

`transportId`	
Chrome	No
Edge	No
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	No
Chrome Android	No
Firefox Android	56
Opera Android	?
Safari iOS	?
Samsung Internet Android	No

`writable` ⚠	
Chrome	Yes★
Edge	≤79★
Firefox	56
IE	No
Opera	?
Safari	?
WebView Android	Yes
Chrome Android	Yes★
Firefox Android	56
Opera Android	?
Safari iOS	?
Samsung Internet Android	Yes★

Full support

 No support

 Compatibility unknown

⚠ Non-standard. Expect poor cross-browser support.

* See implementation notes.

Uses a non-standard name.

Last modified: Apr 4, 2019, by MDN contributors

Related Topics

WebRTC API

`RTCIceCandidatePairStats`

▼ Properties

`availableOutgoingBitrate`

`bytesReceived`

`circuitBreakerTriggerCount`

`consentExpiredTimestamp`

`firstRequestTimestamp`

`lastPacketReceivedTimestamp`

`lastPacketSentTimestamp`

`lastRequestTimestamp`

`lastResponseTimestamp`

`localCandidateId`

`nominated`

`packetsReceived`

`packetsSent`

`priority`

`readable`

`remoteCandidateId`

`requestsReceived`

requestsSent

responsesReceived

responsesSent

retransmissionsReceived

state

totalRoundTripTime

transportId

writable

Related pages for WebRTC

MediaDevices.getUserMedia()

Navigator.mediaDevices

RTCCertificate

RTCDTMFSender

RTCDTMFToneChangeEvent

RTCDataChannel

RTCDataChannelEvent

RTCDtlsTransport

RTCErrorEvent

RTCIceCandidate

RTCIceTransport

RTCPeerConnection

RTCPeerConnectionIceErrorEvent

RTCPeerConnectionIceEvent

RTCRtpReceiver

RTCRtpSender

RTCRtpTransceiver

RTCSctpTransport

RTCSessionDescription

RTCStatsEvent

RTCStatsReport

RTCTrackEvent