[peerconnection](https://www.npmjs.com/package/peerconnection)

Parte superior do formulário

Parte inferior do formulário

PeerConnection implementation for Node.js. This allows Node.js server to receive WebRTC calls from the browsers. Only Chrome is tested for now. Firefox might work when the **[interop notes](http://www.webrtc.org/interop" \o "WebRTC interop notes)** are taken into account.

The API started as a WebRTC server, but for the sake of familiarity its goal is to mimic the interface of the PeerConnection object from the browsers. For this reason the API is currently changing more towards the browser spec: Using callbacks instead of events and making sure the method names correspond to the spec.

**Example**

var connections = [];

socket.on('offer', function( sdp ) {

console.log( "New offer:" );

console.log( sdp.sdp );

// Create the PeerConnection and setRemoteDescription from the offer.

var c = new nodertc.PeerConnection();

c.setRemoteDescription( sdp );

connections.push( c );

// File for writing the audio.

var stream = fs.createWriteStream( socket.id + ".pcm" );

c.on( 'icecandidate', function( evt ) {

// New ice candidate from the local socket.

// Emit it to the browser.

socket.emit( 'icecandidate', evt );

});

c.on( 'answer', function( evt ) {

// Answer from the local socket.

// Emit it to the browser.

socket.emit( 'answer', evt );

});

c.on( 'audio', function( evt ) {

// Audio from the local socket.

// Write it to a f ile.

stream.write( evt.data );

});

socket.on( 'icecandidate', function( evt ) {

// icecandidate from the remote connection,

// add it to the local connection.

c.addIceCandidate( evt );

});

});

**Platform support**

Supported platforms:

* Linux x64

The platform restrictions are due to the native **[libjingle](http://code.google.com/p/libjingle/" \o "libjingle; Google Talk Voice and P2P Interoperability Library)** bindings. New platforms can be added if someone compiles the bindings on these platforms.

Compile libjingle on a new platform using

gclient config http://libjingle.googlecode.com/svn/trunk

gclient sync

cd trunk

ninja -C out/Release peerconnection\_client

After this the bindings can be compiled with node-gyp rebuild. The binding.gyp file assumes that libjingle is located next to node-peerclient:

../node-peerclient

../libjingle/trunk

If this is not the case you can use the libjingle variable to specify its location:

node-gyp rebuild --libjingle=/usr/sdks/libjingle/trunk

**Thanks**

* Google and rest of the libjingle team for the WebRTC implementation.
* João Martins for the original idea on using libjingle.