**webRTC**

* webRTC means Web Real Time Communication
* It helps to connect pear to pear for real time
* webRTC is used to make video & audio share in real time
* Also used to file share or transfer
* Its opensource & free
* We just need to use webRTC supported browser like chrome, mozilla, opera, edge, android etc
* It is aslo available mobile application

**webRTC API**

* getUserMedia
* getDisplayMedia
* RTCPeerConnection
* RTCDataChannel
* GetStats

**webRTC Signaling**

* To control communication session between two clients



**RTCPeerConnection**

* In Every client form other devices are called peer
* To connect all clients in one peer we use RTCPeerConnection
* Now create a RTCPeerConnection   
  const peerConn = new RTCPeerConnection( configuration );
* To setup peer connection we need two type of server like stun server or turn server, those are called iceServer ( interactive connectivity Establishment )
* RTCPeerConnection configuration   
  let servers = {  
   iceServers : [  
   urls : [ 'stun:stun1.1.google.com:19302',   
   'stun:stun2.1.google.com:19302'

]  
 ]

}

**RTCPeerConnection API method**

* createOffer
* createAnswer
* setLocalDescription
* localDescription
* setRemoteDescription
* removeDescription
* onicecandidate
* ontrack
* getTracks
* getAudioTracks
* getVideoTracks
* addStream
* addIceCandidate

**Create SDP Offer**

* To create SDP ( Session Description protocall ) Offer we can follow this   
  let offer = peerConnection.createOffer();  
  peerConnection.setLocalDescription( offer );

**Create SDP Answer**

* To create SDP ( Session Description protocall ) Answer we can follow this   
  let answer = peerConnection.createAnswer();  
  peerConnection.setLocalDescription( answer );

**ICE Candidate**

* Ice candidate is a user in our peer request.
* We can check is any user are here in ice candidate then we can accept or send any action   
    
  peerConnection.onicecandidate = ( event ) => {  
   if( event.candidate ){  
   // take action with this candidate   
   }  
  }