# Understanding WebRTC Signaling & Socket.io for Voice Calls

WebRTC (Web Real-Time Communication) enables direct peer-to-peer communication for voice and video calls. However, before two users can connect, they need to exchange connection details using a process called signaling. In this project, we use WebSockets (Socket.io) for signaling.

## 1. What is WebRTC Signaling?

Signaling is the process of exchanging connection details between two users before a WebRTC session begins. This includes:

• Session Description Protocol (SDP) – Exchange of offer and answer for call setup.

• Interactive Connectivity Establishment (ICE) – Finding the best network path for communication.

## 2. How WebSockets Work in WebRTC

WebSockets provide a persistent connection between the client and server to enable real-time communication. In our case, we use WebSockets to exchange signaling messages between peers.

## 3. WebSocket Events Explained

### • Client Connection

When a user connects, they are assigned a unique ID.

```javascript  
io.on('connection', (socket) => {  
 console.log('New client connected:', socket.id);  
});  
```

### • Joining a Call (Room System)

Users must join a specific room before a call starts.

```javascript  
socket.on('join-call', (roomId) => {  
 socket.join(roomId);  
 socket.to(roomId).emit('user-joined', socket.id);  
});  
```

### • WebRTC Offer & Answer Exchange

WebRTC requires an offer-answer exchange to establish the call.

```javascript  
socket.on('offer', (data) => {  
 socket.to(data.roomId).emit('receive-offer', data);  
});  
socket.on('answer', (data) => {  
 socket.to(data.roomId).emit('receive-answer', data);  
});  
```

### • Exchanging ICE Candidates

ICE candidates help determine the best network route for the call.

```javascript  
socket.on('ice-candidate', (data) => {  
 socket.to(data.roomId).emit('receive-ice-candidate', data);  
});  
```

### • Handling Disconnections

When a user disconnects, it is logged and handled.

```javascript  
socket.on('disconnect', () => {  
 console.log('Client disconnected:', socket.id);  
});  
```

## 4. Learning Resources

• WebRTC Basics: https://www.youtube.com/watch?v=DvlyzDZDEq4

• WebRTC Signaling using Socket.io: https://www.youtube.com/watch?v=6fZ\_MZZnVd4

• Mozilla WebRTC API Docs: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC\_API