WebRTC

Protocols, API Tour, Sessions, SDP, PeerConnection, DataChannel, Media, Codecs, NAT, Security, Project

Real time communication (RTC) is intended for person-to-person live communication. WebRTC is built into modern standards-compliant web browsers, mobile devices and IoT devices. WebRTC implements RTC without needing browser extensions or plug-ins and browsers supporting it are already widely deployed. Forms of communication WebRTC supports include audio, video and data exchange (e.g. app data or docs).

WebRTC is the basis for RTC within many interactive apps that are widely used. What is less known is that WebRTC is a programming platform, and apps running in the browser & on mobile/IoT devices can leverage it. Developers striving to add unique features to their apps would be well advised to consider WebRTC, as once an understanding of how it works has been gained, it is not that time consuming to program. The results are quite powerful and much appreciated by users.

This course starts with an introduction to WebRTC, then looks at the protocols and API standards in detail, then looks at how to program it using TypeScript, paying close attention to everthing app developers need to know, both on the client and the server.

Target Audience

Experienced web application developers who require a deeper understanding of WebRTC and how to use it within their own web applications.

Prerequisites

All demo code and lab exercises uses TypeScript, so web programming experience using that is required, as is good all-round foundational networking knowledge.

No previous WebRTC experience required.

Contents of One-Day Training Course

WebRTC Intro

What it is trying to achieve - a community, a set of standard protocols, a set of standard APIs, a foundation upon which to build apps and an open source project

Architecture

Major components for audio, video, data
How transport works; role of codecs
Identity and security (e.g. DTLS)
How to use with TypeScript

Foundational IETF Protocols

RTP /SRTP: Realtime Transport Protocol
RTCP/SRTCP: RTP Control Protocol
SDP: Session Description Protocol
SCTP: Stream Control Transmission
Future: QUIC and WebRTC?

W3C Standards

W3C is working on many WebRTC stds
Review of WebRTC 1.0: Real-time Comms Between Browsers, Media Capture and Streams and WebRTC's Statistics

Connectivity Establishment

Interactive Connectivity Establishment
How to use ICE/SDP to programmatically connect with remote party [JSEP]
Detailed look at RTCPeerConnection
Works from DOM thread, not web worker

Data Channel Programming

Exchanging app data over WebRTC links
RTCDataChannel: creation and use
Using send() from Typescript
Typings already defined in lib.dom.d.ts (which Angular CLI adds to tsconfig.json)

MediaStream

MediaStream interface
MediaStreamTrack interface
Codec selection (look at AV1 and OPUS)

WebRTC Media API

Passing track info to remote parties
RTCRtp[Sender|Receiver|Transceiver]
Encoding / transmission / processing

WebRTC on the Server

Role of signaling (custom to server app)
Your server application exposes REST API
MCU, SFU, gateways, ...

WebRTC and STUN/TURN

Issues with some networks (NAT/firewall)
Navigating NATs with WebRTC (STUN)
Relays using TURN (e.g. JANUS)

WebRTC And Security

How to identify/locate participants
Securing comms links (DTLS, secure RTP)
Regulation: comms, GDPR, police, ..

Application Issues

Developer environment setup for WebRTC
Debugging (chrome://webrtc-internals/)
Error handling with WebRTC
WebRTC as part of a large application suite

Architecture

Reference architecture exploring how UI of a client app app can be shared via WebRTC's data channel

Project

Attendees work on joint project to add WebRTC functionality to a larger web application