WebRTC School

Learn and Qualify

WSQD™ Certification

Exam Objectives

The **WebRTC School Qualified Developer (WSQD™)** is designed to test your skills and knowledge on the both the technologies that WebRTC utilizes as well as the APIs that are required for you to start creating your own WebRTC applications.

Everything that you need to cover in order to pass this test is in the **WebRTC School Qualified Developer** program but if you decide to learn about WebRTC elsewhere then these are the topics that you should learn about in order to be prepared for the test.

This list is the same as the 'course topics' list also found under the 'outline' button next to the course name in the Catalog.

Please note that if you go along an alternate training path it is possible that you may get a question that may not have been covered in that path. It's up to you!

Please view the following pages for the complete topic list….

Introduction to the course

Topics to study:

- ‘Goals’ of the course
- Topics covered and the ‘approach’
- Be a WebRTC ‘Chameleon’

Introduction to WebRTC

Topics to study:

- *Real-Time* Communication on the Internet
- WebRTC is “Skype™ in the browser”
- What's New?
- A Short History of WebRTC
- WebRTC Support of Multiple Media
- WebRTC Triangle
- WebRTC Trapezoid
- WebRTC and SIP
- WebRTC and Jingle
- WebRTC and PSTN

WebRTC API Preview

Topics to study:

- WebRTC APIs ‘flowchart’
- Obtain Local Media
- Set Up Peer Connections
- Attach Media or Data
- Exchange Offer / Answer
    - createOffer()
    - createAnswer()
    - setLocalDescription()
    - setRemoteDescription()
- API Flow

Media Flows in WebRTC

Topics to study:

- Media Flows in WebRTC
- Media without WebRTC
- Peer-to-Peer Media with WebRTC
- NAT Complicates Peer-to-Peer Media
- What is a NAT?
- NAT Example
- NATs and Applications
- Peer-to-Peer Media ‘through’ NAT
- ICE Connectivity Checks
- P2P Media Can Stay Local to NAT
- ICE Servers
- Browser Queries STUN Server
- TURN Server Can Relay Media
- NAT and IPv6

IETF Protocols

Topics to study:

- IETF Protocols
- WebRTC: A Joint Standards Effort
- IETF Standards
- The WebRTC Protocol Stack
- WebRTC Protocols
- Internet Protocols
- Transmission Control Protocol
- User Datagram Protocol
- Transport Layer Security
- Datagram Transport Layer Security
- Stream Control Transport Protocol
- Hypertext Transport Protocol
- The WebSocket Protocol
- Secure Real-Time Transport Protocol
- Session Description Protocol
- Session Traversal Utilities for NAT
- Traversal Using Relays around NAT
- Interactive Connectivity Establishment
- What about SIP?

WebRTC Media Handling APIs

Topics to study:

- W3C WebRTC standards work
- Standards process and conformance
- Two API modules
- Local Media Handling Example
- Sources
- Tracks
- MediaStreamTrack API
- The constraint approach
- Constraints
- States
- Capabilities
- SourceInfo
- MediaStreamTrack API
- MediaStreamTrack subclasses
- Streams
- MediaStream API
- getUserMedia()

WebRTC Media Transmission APIs

Topics to study:

- A Peer Connection
- WebRTC Triangle ‘Review’
- A Peer Connection is
- About ICE
- WebRTC NAT Traversal
- ICE Call Flow
- ICE Gathering State Machine
- ICE Connection State Machine
- Offer/Answer
- Offer/Answer State Machine
- Media Description
- MediaStreamTrack API
- RTCPeerConnection Core API
- “Extra” APIs
- DTMF
- DTMF in WebRTC
- DTMF API
- Data Channel
- Data Channel API
- Identity API
- Statistics

Simple WebRTC Pseudo code example

Topics to study:

- Pseudo Code
- Mobile browser code outline
- Mobile browser 'streams' example
- function getMedia()
- function createPC()
- Function handleIncomingStream()
- Function show_av(st)
- Mobile browser code outline
- function attachMedia() [1]
- function call()
- How do we get the SDP answer?
- Laptop browser stream example
- Signaling channel message is trigger
- Function prepareForIncomingCall()
- Function answer()
- Function handleIncomingStream()
- Function getMedia() [1]
- Function attachMedia()
- In real code . .

The Signaling Channel

Topics to study:

- WebRTC signaling
- Role of Signaling
- Why Signaling is Not Standardized
- Server Chooses Signaling Protocol
- Some Signaling is Needed
- Signaling in WebRTC
    - Signaling State Machine
    - Signaling Transport Options
    - Signaling Transport Example: WebSockets
    - Signaling Transport Example: HTTP
    - Signaling Transport Example: Data Channel
    - Signaling Protocol Options
    - Signaling Example: HTTP Polling
    - Signaling Example: WebSockets
- WebSocket code outline
- Standardized protocols
- Open Source JavaScript SIP Stacks
- Open Source JavaScript XMPP
- Comparison of Approaches

Basic ‘Real code’ example

Topics to study:

- Overview
- Structure of module
- Web server
- Node.js
- Index and Server
- Signaling channel
- Signaling channel interface
- Server Signaling
- Client Signaling
- Web application
- What’s Next?

Security and Privacy

Topics to study:

- What are Security and Privacy?
- Is WebRTC Secure and Private?
- Web Security and Privacy Model
- Browser/Web Security Model
- Browser/Web Privacy Model
- How WebRTC Changes Browser Privacy
- Browser Prompts for Permission
- How WebRTC Changes Browser Security
- New Attacks
- WebRTC API Attacks
- Signaling Channel Attacks
- Security of WebRTC Media Sessions
- Website Identity
- Browser User Identity
- Secure Real Time Protocol
    - Secure Profile of RTP (SRTP)
    - SRTP Encryption and Authentication
    - SRTP Key Management
    - Key Exchange in Signaling
    - Generating a Key in Media Path*
    - SDP Security Descriptions
    - Generating a Key in Media Path
    - SRTP Call Flow
- DTLS-SRTP Key Agreement
- DTLS Client/Server
- Authenticating a Key Agreement
- Authenticating a Fingerprint
- Identity Proxy in WebRTC Triangle
- Single Identity Proxy with Triangle
- Identity Proxy in WebRTC Trapezoid
- Identity Services
- Communication Consent
- ICE Communication Consent
- Privacy in WebRTC
- Identity Privacy
- IP Address Privacy
- Browser Fingerprinting
- Media Privacy
- WebRTC and the Enterprise
- WebRTC Security Summary

Status of WebRTC and ‘What’s next’

Topics to study:

- Status of WebRTC APIs
- Status of WebRTC Protocols
- Status of Browser support of WebRTC
- Support of Signaling Channel
- Support in Mobile
- Opportunities
- Obstacles
- What’s next?

Use Cases

Topics to study:

- WebRTC Use Cases and Requirements
- Web Conferencing (Multiparty Conferencing)
- Communications Client (UC and Consumer)
- Contact Centers (B2C and Agent)
- Distributed Communication (Freemium Services)
- Mobile (Voice for Smartphones)
- Single Line of Code WebRTC (Libraries)
- Control (Microphone and Cam Access)
- Gaming (In-Game Media, Chat and Data Channel)
- Overlay Network (Data Channel)

Interoperability and Portability

Topics to study:

- Definitions
- Relative importance
- Things to watch
- Implemented APIs still prefixed
- Connecting to output is different
- No default STUN servers
- Multiple audio/video not supported
- Opus not default
- DTLS-SRTP
- Staying current
- Other resources