



Web RTC.

Web Real-Time Communication



같이 밥 먹어 조

WebRTC(Web Real-Time Communication)

- 드라이버나 플러그인 설치 없이 Web-browser 간 데이터 교환 가능

웹 애플리케이션과 사이트가 중간자 없이
브라우저 간에 오디오나 영상 미디어를 포착하고
마음대로 Stream 할 수 있도록 하는 기술
임의의 데이터도 교환도 가능!(peer to peer방식)

호환성

- 크롬에서는 완벽 호환(SSL인증시)
- But 다양한 플랫폼에서 표준화가 되어있지 않음

환경구축

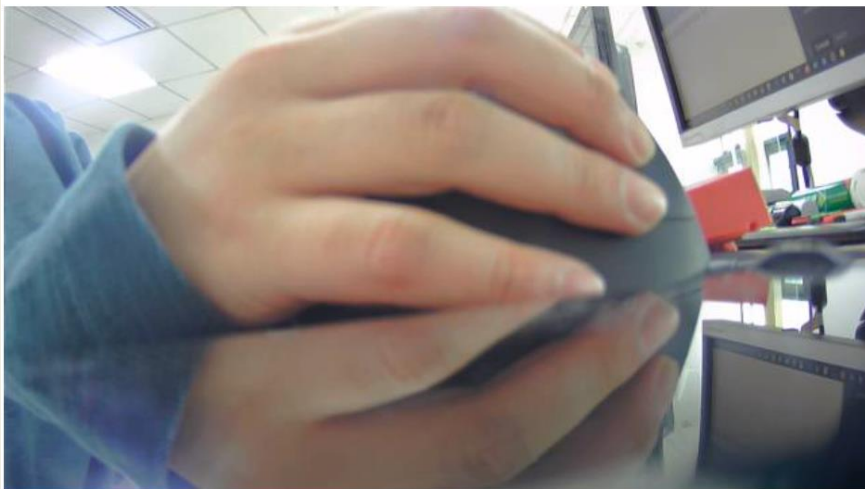
1. 서버구축 Node JS	Windows PowerShell 접속 Chocolatey 다운 Node Js 설치	SSL 탑재 서버 운용 Peer to peer 방식
2. SSL 준비 Open ssl	Https로 사용	getUserMedia() 사용 가능
3. sampleCode 준비 Web-RTC	Index.html -> index.js Index.js Adapter.js Package.json	Npm install Web-Cam 사용 가능

실행 가능

시연 시작

WebRTC 가동해보기

내화면(나를 찍고있는 화면)



상대방 화면(상대방이 보고 있는 송출중인 나의 화면)

```

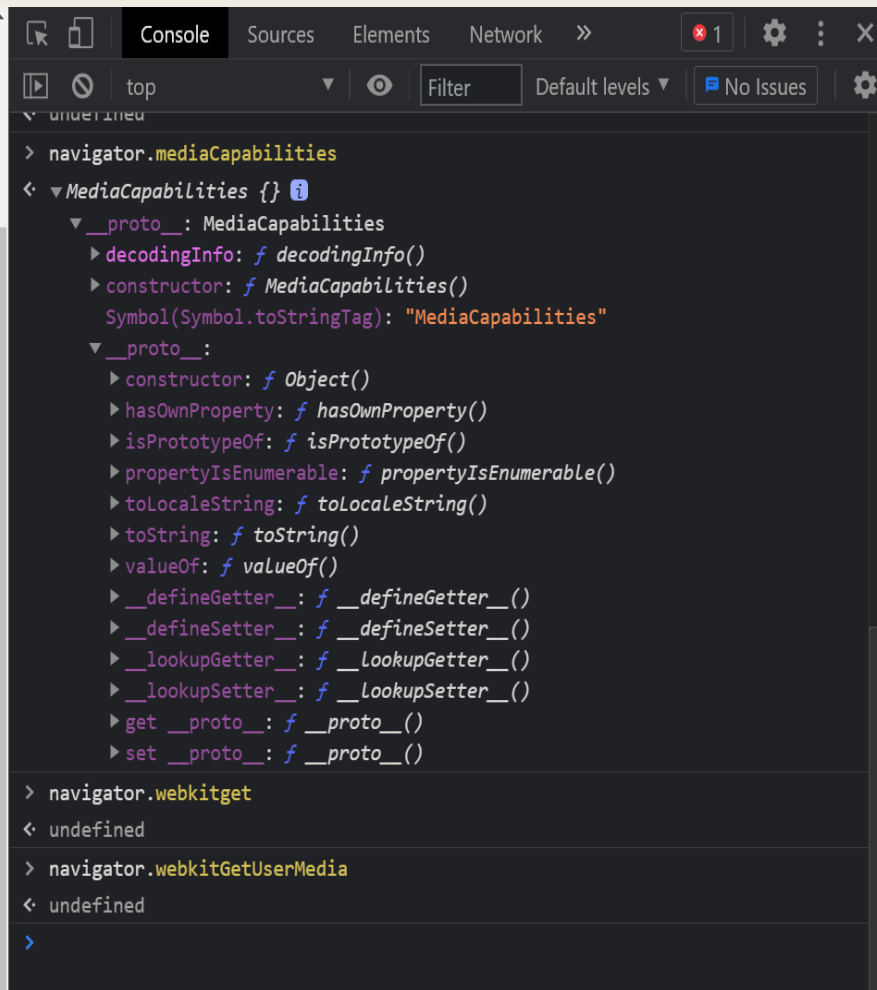
    Console Sources Elements Network >>
    top Filter Default levels No Issues

    > __proto__: Navigator
    > navigator.getUserMedia
    < f getUserMedia(constraints, onSuccess, onError) {
      shimConstraints(constraints, function(c) {
        navigator.webkitGetUserMedia(c, onSuccess, function(e) {
          if (onError) {
            onEr...
          }
        })
      })
    }

    > navigator.mediaCapabilities
    < MediaCapabilities {}
    > __proto__: MediaCapabilities
    > decodingInfo: f decodingInfo()
    > constructor: f MediaCapabilities()
    > Symbol(Symbol.toStringTag): "MediaCapabilities"
    > __proto__:
    > constructor: f Object()
    > hasOwnProperty: f hasOwnProperty()
    > isPrototypeOf: f isPrototypeOf()
    > propertyIsEnumerable: f propertyIsEnumerable()
    > toLocaleString: f toLocaleString()
    > toString: f toString()
    > valueOf: f valueOf()
    > __defineGetter__: f __defineGetter__()
    > __defineSetter__: f __defineSetter__()
    > __lookupGetter__: f __lookupGetter__()
    > __lookupSetter__: f __lookupSetter__()
    > get __proto__: f __proto__()
    > set __proto__: f __proto__()

    > navigator.webkitGetUserMedia
    < f webkitGetUserMedia() { [native code] }
    >
  
```

상대방 화면(상대방이 보고 있는 송출중인 나의 화면)



NotAllowedError

사용자가 브라우저 설정을 통해 장치에 대한 접근권한을 차단하였거나 장치 사용 권한 요청에 거부한 경우 이 에러가 발생합니다. 이 외에도 어떤 식으로든 장치에 대한 접근을 차단하였다면 이 에러가 발생합니다.

이전 버전의 규격에서는 이 에러와 `SecurityError` 를 동일한 의미로 사용하였지만, 현재 버전에서는 다른 의미로 사용하므로 혼용하여선 안 됩니다.

NotFoundError

`constraints` 매개변수 조건에 맞는 미디어 트랙이 없는 경우 이 에러가 발생합니다.

MediaDevices.getUserMedia()

`MediaDevices` 인터페이스의 `getUserMedia()` 메서드는 사용자에게 미디어 입력 장치 사용 권한을 요청하며, 사용자가 수락하면 요청한 미디어 종류의 트랙을 포함한 `MediaStream` (en-US)을 반환합니다. 스트림은 카메라, 비디오 녹화 장치, 스크린 공유 장치 등 하드웨어와 가장 비디오 소스가 생성하는 비디오 트랙과, 마이크, A/D 변환기 등 물리적과 가상 오디오 장치가 생성하는 오디오 스트림, 그리고 그 외의 다른 종류의 스트림을 포함할 수 있습니다.

반환하는 값은 `MediaStream` (en-US) 객체로 이행하는 `Promise`입니다. 사용자가 권한 요청을 거부했거나 일치하는 유형의 미디어를 사용할 수 없는 경우, 프로미스는 각각 `NotAllowedError`와 `NotFoundError`로 거부합니다.

참고: 사용자가 권한 요청에 대한 선택을 하지 않고 완전히 무시할 수도 있기 때문에, 프로미스 또한 이행도 거부도 하지 않을 수 있습니다.

보통, `MediaDevices` 싱글톤 객체는 다음과 같이 `navigator.mediaDevices`를 사용해 접근합니다.

OpenSSL

위키백과, 우리 모두의 백과사전.

OpenSSL은 네트워크를 통한 데이터 통신에 쓰이는 프로토콜인 TLS와 SSL의 오픈 소스 구현판이다. C 언어로 작성되어 있는 중심 라이브러리 안에는, 기본적인 암호화 기능 및 여러 유틸리티 함수들이 구현되어 있다.

OpenSSL은 Eric A. Young과 Tim Hudson이 만든 SSLeay에 그 근거를 두고 있다!!!. SSLeay의 개발은 Young과 Hudson이 RSA Security로 적을 옮긴 1998년 12월 이래 비공식적으로 중단되어 있다.[\[출처 필요\]](#)

거의 모든 버전의 유닉스 계열 운영 체제(솔라리스, 맥 OS X, 리눅스, BSD 포함) 및 OpenVMS, 윈도우에서 OpenSSL을 이용할 수 있다.

목차 [숨기기]

- 1 주요 버전 배포 현황
- 2 알고리즘
- 3 취약점
 - 3.1 하트블리드
- 4 같이 보기
- 5 각주
- 6 외부 링크