

- Description

cocolib 은 KnowledgeTalk COCO 시리즈 서비스의 특화된 기능을 더 쉽고 편하게 제공하기 위해 제작된 웹 브라우저 용 자바스크립트 라이브러리입니다.

본 문서 및 본 문서에 관련된 라이브러리, 샘플은 날리지포인트의 허가 없이 수정 및 재배포를 금합니다. 문의는 ivypark@knowledgepoint.co.kr로 메일 부탁드립니다.

- How to import? (in HTML)

```
<script type="module" src="https://knowledgepoint.co.kr/lib/cocolib-0.9.1.min.js"> </script>
```

- Sample

https://knowledgepoint.co.kr/sample/KnowledgeTalk-4.0_sample.zip

- Functions

1) 사생활 보호 모드

- Private Mode 란?

사생활 보호 기능은 비디오에서 사람 이미지만을 추출하여 배경을 흐리게 처리하는 기능입니다. 배경이 가려짐으로써 사용자에게 집중할 수 있다는 장점이 있습니다.

- To prepare (refer to sample)

1. WebRTC getUserMedia() 호출로 받은 stream 을 담은 video 객체 (<video>, HTMLVideoElement)
2. 사생활 보호 기능을 출력할 canvas 객체 (<canvas>, HTMLCanvasElement)

- Event flow (refer to sample)

1. 사생활 보호 기능을 시작하고 싶다면, start()를 호출하세요.
2. 사생활 보호 기능을 종료하고 싶다면, stop()을 호출하세요.

- Methods & Properties

(다음 장에 계속)

1. *privateMode.start()* : Method

cocolib.privateMode.start(canvas, video [, constraint, bgBlur, edgeBlur, horizontal]);
@parameter canvas : HTMLCanvasElement -> 기능을 출력할 Canvas 태그(객체)를 넣습니다. video : HTMLVideoElement -> Stream 이 있는 Video 태그(객체)를 넣습니다. constraint : Object (Optional) (default: {width: 960, height: 540}) -> 출력물의 width 와 height 를 설정합니다. ({width: xxx, height: xxx} 형식) bgBlur : Number (Optional) (default: 10) -> 배경 흐림 계수를 정합니다. 높을수록 흐려집니다. (최소 1, 최대 20) edgeBlur : Number (Optional) (default: 3) -> 배경과 사람 사이의 흐림 계수를 정합니다. 높을수록 흐려집니다. (최소 1, 최대 5) horizontal : Boolean (Optional) (default: false) -> 출력물의 좌우 반전 여부를 정합니다. false: 좌우 반전하지 않음, true: 좌우 반전
@return undefined
Description canvas 에 사생활 보호가 적용된 출력물이 나오게 됩니다. 구현 시 video 와 canvas 를 각각 감추고 보이게 하는 화면처리 등이 필요합니다. (샘플 참고)

2. *privateMode.stop()* : Method

cocolib.privateMode.stop();
@parameter none
@return undefined
Description canvas 에 사생활 보호가 적용된 출력물이 나오는 것이 멈추게 됩니다. 다시 video 와 canvas 를 각각 보이고 감추게 하는 화면처리 등이 필요합니다. (샘플 참고)

(다음 장에 계속)

2) 화자 감지

- Detect Speaker란?

화자 감지 기능은 일정 시간 이상 소리가 인식되는 발화자를 감지하여 화면에 표시하는 기능입니다.

- To prepare (refer to sample)

1. KnowledgeTalk 플랫폼을 이용하여 3자 이상의 통화 진행 필요, Presence 확인 필요 (연동규격서 참고)

- Event flow (refer to sample)

1. 화자 감지 기능을 시작하실 때, `start()`를 호출하세요.

2. 화자 감지 기능을 종료하실 때, `stop()`을 호출하세요.

3. 현재 화자 감지 중인지를 확인하시려면, `status`를 확인하세요.

(본 라이브러리에서는 화자 감지 기능을 직접 제공하는 것이 아니라, 화자 감지를 하기 위해 필요한 상태 관리 등의 보조 기능을 제공합니다. 화자 감지 기능은 본 라이브러리를 사용하지 않으시더라도 구현이 가능합니다.)

- Methods & Properties

1. `detectSpeaker.status` : Property

`cocolib.detectSpeaker.status`

@return

status: Boolean

2. `detectSpeaker.start()` : Method

`cocolib.detectSpeaker.stop()`

@parameter

none

@return

undefined

3. `detectSpeaker.stop()` : Method

`cocolib.detectSpeaker.stop()`

@parameter

none

@return

undefined

(다음 장에 계속)

3) 오디오 세이프

- Audio Safe 란?

오디오 세이프 기능은 자신의 네트워크 상태를 체크하여 영상 수신을 제어하고, 최소한 음성 통화는 보장하게 하는 기능입니다. 네트워크가 좋지 않다면 영상 수신을 차단하고, 네트워크가 좋아지면 영상을 다시 수신하게 됩니다.

- To prepare (refer to sample)

1. KnowledgeTalk 플랫폼을 이용하여 3자 이상의 통화 진행 필요

- Event flow (refer to sample)

1. 오디오 세이프 기능을 시작하고 싶다면, `start()`를 호출하세요.
2. 오디오 세이프 기능 시작 전에 3개의 네트워크 상태 체크 `EventListener(onaudiosafegd, onaudiosafebd, onaudiosafenbd)`를 등록하세요. (상태 체크 계수는 인원 수에 따라 차등으로 지정되며, 해당 `EventListener`는 `window` 객체에 자동 등록됩니다.)
3. 방의 인원 수가 변경되었다면, `update()`를 호출 하세요. (Presence 메시지 참고)
4. 오디오 세이프 기능을 종료해야 할 때(통화 종료 등)는 `stop()`을 호출하세요.
5. 현재 오디오 세이프 기능이 동작 중인지 확인할 때는 `status`를 확인하세요.

- Methods & Properties & EventListeners

(다음 장에 계속)

1. *audioSafe.status* : Property

cocolib.audioSafe.status
@return status: Boolean

2. *audioSafe.start()* : Method

cocolib.audioSafe.start(peer, count);
@parameter peer : RTCPeerConnection -> 자신의 RTCPeerConnection 객체를 넣습니다. count : Number -> 현재 화면에 표시된 인원 수를 넣습니다. (최소 2, 최대 25)
@return undefined
Description Local Peerconnection 객체의 네트워크 속도를 체크하여 eventlistener 에 dispatch 합니다(start() 호출 전 eventlistener 등록 필요). 3 자 이상 통화 시 한 번만 호출하셔야 합니다(여러 번 호출 시 event 가 여러 번 발생). 화면에 표시된 인원 수가 변경되었다면, update()를 호출해 주시면 됩니다. 인원 수가 맞지 않는다면 원치 않는 결과가 발생할 수 있으니 주의하세요.

3. *audioSafe.stop()* : Method

cocolib.audioSafe.stop();
@parameter none
@return undefined
Description audioSafe 기능을 종료합니다. 통화 종료 혹은 시나리오에 맞게 사용하시면 됩니다.

4. *audioSafe.update()* : Method

cocolib.audioSafe.update(count);
@parameter count : Number -> 현재 화면에 표시된 인원 수를 넣습니다. (최소 2, 최대 25)
@return undefined
Description start() 이후 사용하세요. 다른 사용자가 들어오거나 나갔을 때 인원을 업데이트 하기 위해 사용됩니다.

5. *onaudiosafegd* : *EventListener*

window.addEventListener("onaudiosafegd", callback);
@parameter callback : Function -> 네트워크 상태가 좋아졌을 때의 이벤트 혹은 화면 처리 등을 작성하세요.
@return undefined
Description 네트워크 상태가 좋아졌을 때 trigger 됩니다.

6. *onaudiosafebd* : *EventListener*

window.addEventListener("onaudiosafebd", callback);
@parameter callback : Function -> 네트워크 상태가 나빠졌을 때의 이벤트 혹은 화면 처리 등을 작성하세요.
@return undefined
Description 네트워크 상태가 나빠졌을 때 trigger 됩니다.

7. *onaudiosafenbd* : *EventListener*

window.addEventListener("onaudiosafenbd", callback);
@parameter callback : Function -> 네트워크 상태가 좋지도, 나쁘지도 않을 때의 이벤트 혹은 화면 처리 등을 작성하세요.
@return undefined
Description 네트워크 상태가 좋지도, 나쁘지도 않을 때 trigger 됩니다.

(다음 장에 계속)

- Release Note

v0.9.1 (201119)

- audioSafe.update() method 추가

v0.9.0 (initialize)

- 3 개 기능(오디오 세이프, 화자 감지, 사생활 보호 모드) 구현, 문서 초안 작성