



CogInsight

Chatbot SDK for iOS

COPYRIGHT MindwareWorks.

121-839 서울특별시 월드컵북로 12길 19 대성빌딩 2층

전화 (070) 4046-6752 • 팩스 (02) 3406-6752

담당자 문서 제·개정 이력			
버전	날짜	내용	담당자
1.4.0	2022-08-22	최초 작성	양철웅



목차

목차	2
개요	3
요구 사항	4
지원 환경	4
인증 정보	4
폴더 구조	4
Chatbot SDK	5
1. 프로젝트 설정	5
프로젝트 설정	5
info.plist 수정	5
2. 초기화	6
3. STT (Speech to Text)	7
4. Dialog Message	8
5. TTS (Text to Speech)	9
Reference	10
COGIMessageKitConfig	10
COGIMessageKitDelegate	11
COGIMessageKit	13
COGISttResult	14
COGIMessageOutput	14
Appendix	15



개요

본 문서는 Mindwareworks 의 CogInsight 솔루션의 Chatbot SDK 사용 방법을 설명하고 있습니다.

본 문서는 기술적인 오류나 구문의 오류를 포함하고 있을 수 있습니다. 당사는 본 문서에 포함된 정보의 정확성을 유지하기 위해 최선을 다할 것이나, 본 문서의 기술적 오류, 잘못된 정보가 포함되어 있 지 않다는 것을 보증하지 않습니다.

본 문서는 특별한 언급 없이 지속적으로 수정과 보완할 것이나 본 문서에 기술된 정보로 인하여 발생할 수 있는 직접적 혹은 간접적 손해, 데이터, 프로그램 기타 무형의 재산에 관한 손실, 사용 이익의 손실 등에 관하여 비록 이와 같은 손해 가능성에 대해 사전에 알고 있었다고 해도 손해 배상 등 기타 책임을 지지 않습니다.

사용자는 본 문서를 구입하거나, 전자문서로 다운로드 받거나, 사용을 시작함으로써, 본 사항에 명시된 내용을 이해하며, 이에 동의하는 것으로 간주합니다. 또한 본 내용이 이전의 문구나 기타 고지에 우선 하는 것임을 인정합니다.



요구 사항

CogInsight 솔루션의 Console 사이트에서 SDK 사용에 필요한 정보를 확인할 수 있습니다.

지원 환경

1. 지원 OS : iOS Deployment Target : 9.0 이상
2. 개발환경 : XCode 10.1 이상

인증 정보

CogInsight 솔루션 > Console 접속 > Domain > 인증키 관리 메뉴에서 확인

name	type	required	description
projectId	string	Y	챗봇 ID
apiKey	string	Y	Domain API Key

폴더 구조

name	description
COGIChatBotKit	framework 폴더
TestClient	테스트 클라이언트 소스 swift 프로젝트에 MessageKit 사용예 포함

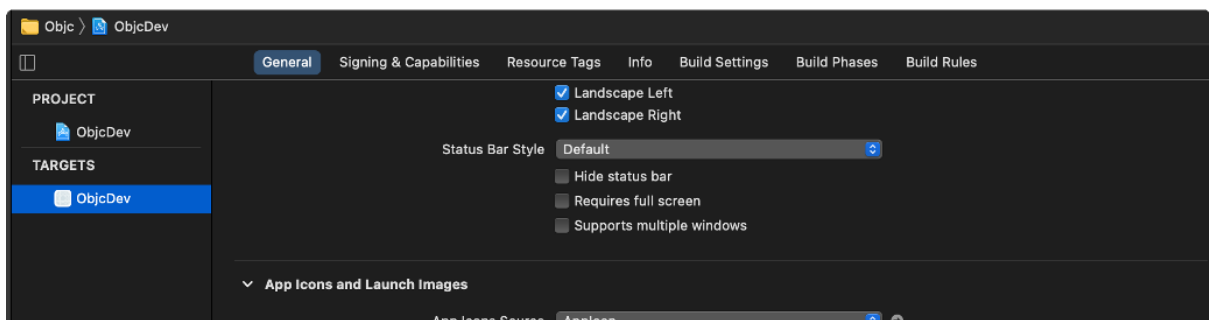
※ TestClient 는 CogInsight 개발 서버 (v2.aiware.io) 의 테스트 챗봇에 연결되어 동작합니다.
네트워크 환경에 따라, 방화벽 오픈 (포트:443) 이 필요할 수 있습니다.

Chatbot SDK

1. 프로젝트 설정

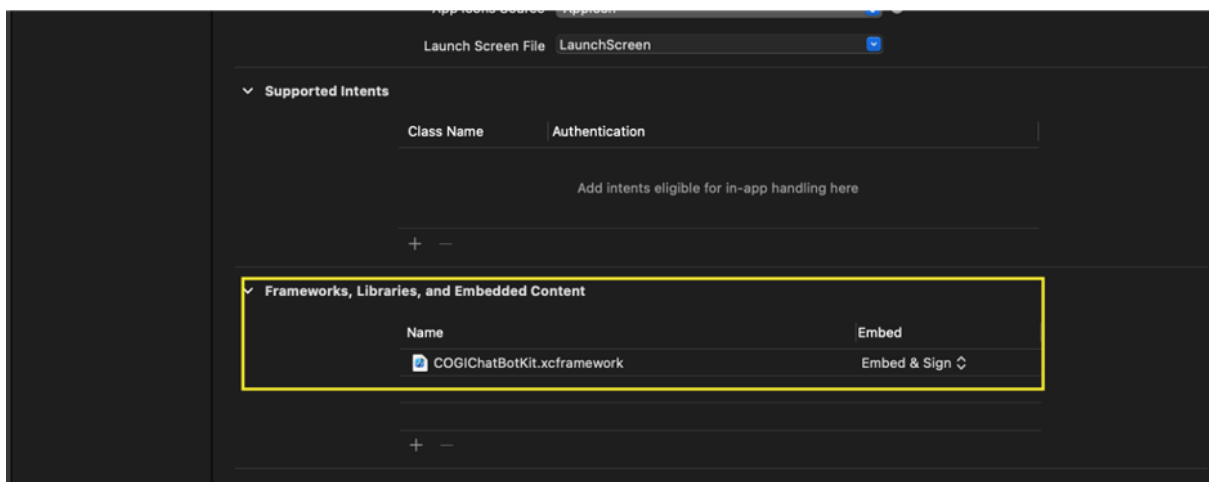
프로젝트 설정

COGIChatBotKit.xcframework 를 Project > General > Frameworks, Libraries, and Embedded Content에 Embed 옵션을 Embedded & Sign로 설정합니다.



info.plist 수정

HTTP통신을 위해 App Transport Security Settings를 설정합니다.
Location, Camera, Microphone을 사용할경우 관련 Privacy를 설정합니다.



2. 초기화

1. COGIChatBotKit 을 import 합니다.
2. COGIMessageKitConfig 를 설정 후 COGIMessageKit를 생성합니다.
3. COGIMessageKitDelegate의 CHAT_SETTING_UPDATED 에서 stt, tts 설정을 합니다.

예시)

```
import COGIChatBotKit
...
var cogiMessageKit: COGIMessageKit?
...
let config = COGIMessageKitConfig()
config.hostUrl = self.hostUrl
config.projectId = self.projectId
config.delegate = self

self.cogiMessageKit = COGIMessageKit(config: config)
...
func messageKit(_ messageKit: COGIMessageKit, event: String, data: [AnyHashable : Any]?) {
    switch event {
    case "CHAT_SETTING_UPDATED":
        if let data = data, let speech = data["speech"] as? NSDictionary, let stt =
speech["stt"] as? NSDictionary, let audioFormat = stt["audioFormat"] as? NSDictionary {

            if let sampleRate = audioFormat["sampleRate"] as? NSNumber {
                self.sampleRate = sampleRate.doubleValue
            }
            if let channels = audioFormat["channels"] as? NSNumber {
                self.channels = channels.uint32Value
            }
            if let ptime = audioFormat["ptime"] as? NSNumber {
                self.ptime = ptime.doubleValue
            }
            // 해당 정보로 audio 관련 세팅
        }
    }
```

3. STT (Speech to Text)

1. messageKitDidLoad가 호출된 이후 messageKit의 startStt() 호출합니다.
2. CHAT_STT_SOCKET_OPEN 이벤트가 오면 오디오 녹음을 시작합니다.
3. audio 데이터를 계속 messageKit의 sendSttData(data:) 함수로 stt로 전달합니다.
4. stt로 해석된 text 는 CHAT_STT_SOCKET_DATA 이벤트로 COGSttResult 형식으로 전달됩니다.
5. result의 text에는 현재까지 해석된 text, sttId에는 처리된 sttId가 전달됩니다.
 - a. 목음이 와서 해석할 음성이 더이상 없을 경우 result의 isFinal 이 true가 세팅되어 넘어갑니다. 이때 audio 녹음을 중단하면 됩니다.
6. stt와의 연결 종료되면 CHAT_STT_SOCKET_CLOSE 이벤트가 호출됩니다.
7. stt도중 취소하고 싶으면 messageKit의 stopStt()를 호출합니다.

예시)

```
...
self.cogiMessageKit?.startStt()
...
func messageKit(_ messageKit: COGIMessageKit, event: String, data: [AnyHashable : Any]?) {
    switch event {
        case "CHAT_STT_SOCKET_OPEN":
            // 녹음 시작
            ...
            // recording한 data전송
            self.cogiMessageKit?.sendSttData(data)
            ...
    }
}

func messageKit(_ messageKit: COGIMessageKit, event: String, data: [AnyHashable : Any]?) {
    switch event {
        ...
        case "CHAT_STT_SOCKET_DATA":
            if let data = data {
                if let result = data["result"] as? COGSttResult {
                    if result.isFinal {
                        // 녹음 중단
                        self.sttResult = result
                        ...
                    }
                }
            }
            ...
    }
}
```


4. Dialog Message

`messageKit.inputSttMessage(sttResult: COGISTtResult)` 함수로 STT로 부터 전달받은 text메시지를 챗봇으로 보냅니다.

stt결과가 아닌 일반 text를 챗봇으로 보내야 할 경우 `messageKit.inputMessage(text:NSString)`를 사용합니다.

입력받은 메시지를 다이얼로그에서 처리후 출력값을 delegate `messageKitMessageReceived`로 전달받습니다.

5. TTS (Text to Speech)

1. messageKitMessageReceived 로 전달받은 messageOutput으로 messageKit.requestTtsData(messageOutput: COGIMessageOutput) 를 호출합니다.
2. 변환된 tts 음성 데이터는 CHAT_TTS_DATA 이벤트로 전달됩니다.

예시)

```
...
self.cogiMessageKit?.requestTtsData(messageOutput)
...
func messageKit(_ messageKit: COGIMessageKit, event: String, data: [AnyHashable : Any]?) {
    switch event {
    ...
    case "CHAT_TTS_DATA":
        if let data = data, let ttsData = data["data"] as? NSData {
            // ttsData play
        }
    ...
}
```

Reference

COGIMessageKitConfig

```
// 필수
NSString *hostUrl
    서비스 호스트 url
NSString *projectId
    챗봇 project ID
id<COGIMessageKitDelegate> delegate
    레퍼런스 COGIMessageKitDelegate 항목 참조
// optional
NSString *fullUrl
    서비스 full url. 외부에서 auth후 생성된 token으로 url호출시 사용
NSString *input
    시작시 input ex) @"{\"text\":\"시작\"}"
NSString *authServer
    auth시 사용할 server 주소. 미지정시 서비스 호스트 url사용
NSString *apiKey
    auth시 사용할 domain apiKey
NSString *userKey
    auth시 사용할 unique한 userKey

NSDictionary *secure
    auth시 사용할 secure data - dialog에서 secure로 접근

BOOL allowSelfSignedCert
    ssl 사설 인증서 허용 여부 - 서버에 사설 ssl 인증서문제로 접속 안될때 사용

NSString *extra
    서비스 호출 url뒤에 추가로 들어갈 문자열.
BOOL encrypt
    userkey, secure로 전달하는 값을 암호화할때 사용
```

COGIMessageKitDelegate

```
messageKitDidStartLoad(_ messageKit: COGIMessageKit)
    챗봇 로딩 시작시 호출

messageKitDidFinishLoad(_ messageKit: COGIMessageKit)
    챗봇이 로딩을 마치고 정상동작을 시작했을때 호출됨

messageKitDidFailLoad(_ messageKit: COGIMessageKit)
    로딩 실패시 호출

messageKitMessageSent(_ messageKit: COGIMessageKit)
    챗봇으로 메시지 입력시 전송

messageKitMessageSendFailed(_ messageKit: COGIMessageKit, data: [AnyHashable : Any])
    챗봇으로 메시지 전송 실패시 data["reason"]에 실패 이유 전달

messageKitMessageReceived(_ messageKit: COGIMessageKit, data: [AnyHashable : Any])
    챗봇 응답 메시지 수신. data["msg"]에 COGIMessageOutput 전달.
    레퍼런스 COGIMessageOutput 항목 참조

messageKit(_ messageKit: COGIMessageKit, didReceiveError message: String)
    오류 메시지 수신시 호출
    auth 실패시 - "chatbot auth failed"
    오디오 권한 오류시 - "audio denied"
    카메라 권한 오류시 - "camera denied"

messageKit(_ messageKit: COGIMessageKit, event: String, data: [AnyHashable : Any]?)
    챗봇에서 보낸 이벤트 처리 - postFront, stt 등
    event의 종류
    CHAT_ACTION
        챗봇에서 PostFront 버튼 클릭시 전달받은 action 이벤트 처리
        챗봇이 부모 app에 요청을 전달할때 사용하며
        json으로 넘긴 경우 NSJSONSerialization으로 파싱하여 사용
        data["data"] 에 json string으로 전달

    CHAT_SETTING_UPDATED
        dictionary로 setting 전달, stt, tts 관련 data(sample rate, channel 수 등) 받아서 세팅

    CHAT_STT_SOCKET_OPEN
        STT로 음성 데이터 전송 준비 완료 이벤트

    CHAT_STT_SOCKET_DATA
        data["result"]에 COGISttResult 전달
        전송 받은 음성데이터를 전환한 text data
        무음 발생하여 음성 입력 완료시 COGISttResult.isFinal true로 전달

    CHAT_STT_SOCKET_CLOSE
```

최종 stt인식되어 text 전달후 stt와 연결 종료 이벤트

CHAT_TTS_DATA

tts요청시 변환된 음성 데이터 전달.

data["data"] 에 NSData로 전달

COGIMessageKit

```
init(config: COGIMessageKitConfig)
    config값을 참조하여 챗봇 생성, 초기화
아래 함수들은 챗봇이 로딩된 이후부터 가능 (delegate의 messageKitDidFinishLoad 호출 이후)

func startStt()
    stt 시작. stt가 준비가되면 CHAT_STT_SOCKET_OPEN 이벤트 전송
func sendSttData(data: NSData)
    입력된 audio data를 stt서버로 전송
func stopStt()
    stt 취소
func inputMessage(text: NSString)
    입력한 text 메시지를 챗봇으로 전송
func inputSttMessage(sttResult: COGISStResult)
    입력된 stt메세지를 챗봇으로 전송
func requestTtsData(msg: COGIMessageOutput)
    받은 message를 tts data로 변환요청
func sendMessage(action: String, data: NSObject?)
    챗봇에 action, data 전달
    가능한 action
    LOCATION_UPDATE
        위치정보 업데이트. 다음과 같은 json string을 인자로 받습니다.
        {
            "longitude": nnn.nnn,
            "latitude": nnn.nnn
        }
    MSG_STATE
        다이얼로그에 데이터를 전달하기 위한 액션입니다. data에 JSON 형식을 String으로 전달하면,
        다이얼로그에서 input.data로 접근가능합니다.
        주의: 한번 state로 설정된 데이터는 이후 모든 메시지에 input.data로 설정되어 전달됩니다.
    MSG_INPUT
        사용자의 입력을 임의로 구성해서 ChatView에 전달합니다. 이는 Chatbot Server로 요청이 발생하여,
        자연스럽게 대화의 흐름을 이어나갈 수 있습니다. 다음과 같은 형식의 JSON을 String으로
        변환하여 전달합니다.
        {
            "text": "...", // 입력 텍스트
            "data": {
                // 입력 데이터 ...
            }
        }
```

COGISTtResult

```
var text: NSString
    stt 결과 text
var sttId: NSString
    sttId
var isFinal: BOOL
    stt 종료 여부. 음성인식 도중이면 false,
    음성인식이 완료되어 최종 text가 나오면 true
```

COGIMessageOutput

```
var messageId: NSString
    해당 메시지 id
var at: NSString
    메시지 생성된 시간
var output: NSArray
    메시지의 output list
func getText -> NSString
    메시지의 output list에서 text타입과 ssm1 타입만 추출
```

Appendix