

- 1 프로젝트 만들기
- 이 사용자 인터페이스 설계
- 3 음성 인식 권한 추가
- 4 음성 인식 권한 부여 요청
- 5 음성 및 오디오 객체를 선언하고 초기화
- 6 전사 시작
- 7 stopTranscribing 방법을 구현
- 8 테스트 앱
- 9 요약

프로젝트 만들기

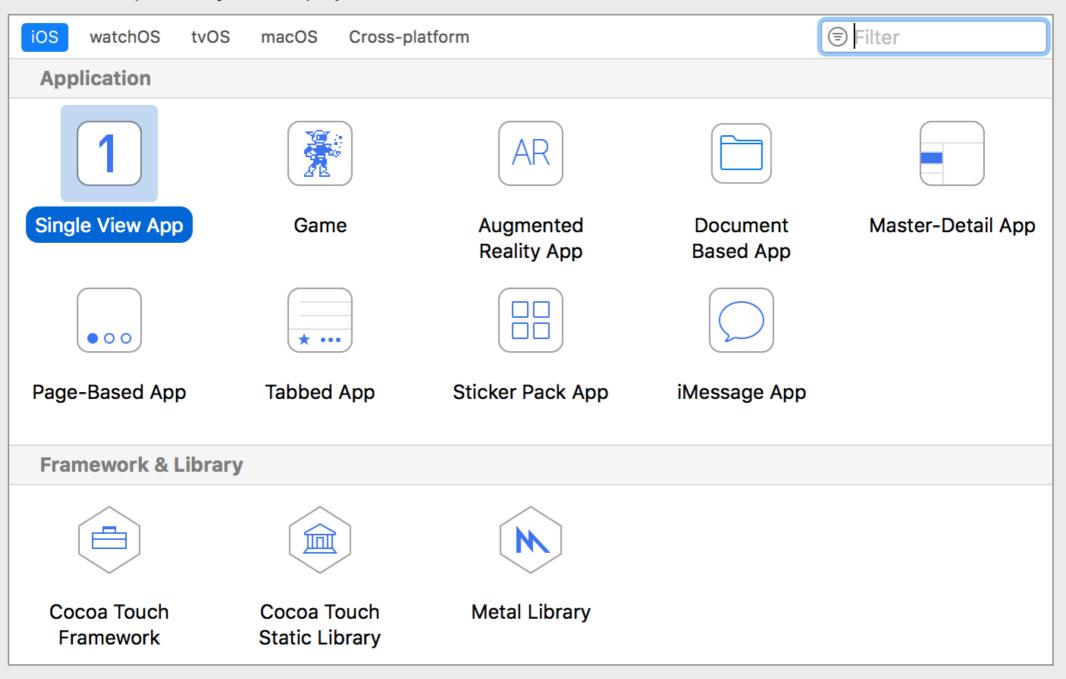
Xcode를 시작하고 Swift 프로그래밍 언어를 사용하여 LiveSpeech라는 새로운 유니버설 단일 뷰 기반 응용 프로그램을 만드는 것으로 시작하십시오.



Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.

Choose a template for your new project:



Cancel

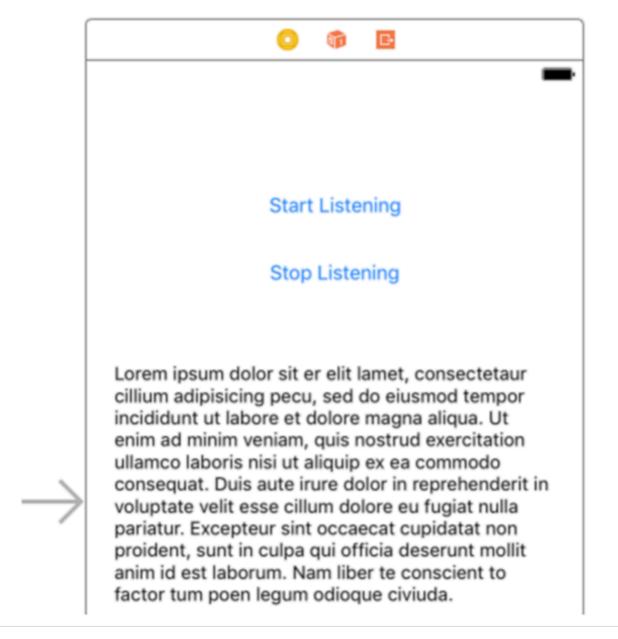
Previous

Next

Product Name:	LiveSpeech		
Team:	Korea Polytechnic University (Game and	\$	
Organization Name:	KIMYOUNG SIK		
Organization Identifier:	kpu.2018class		
Bundle Identifier:	kpu.2018class.LiveSpeech		
Language:	Swift	\$	
	Use Core Data		
	Include Unit Tests		
	Include UI Tests		

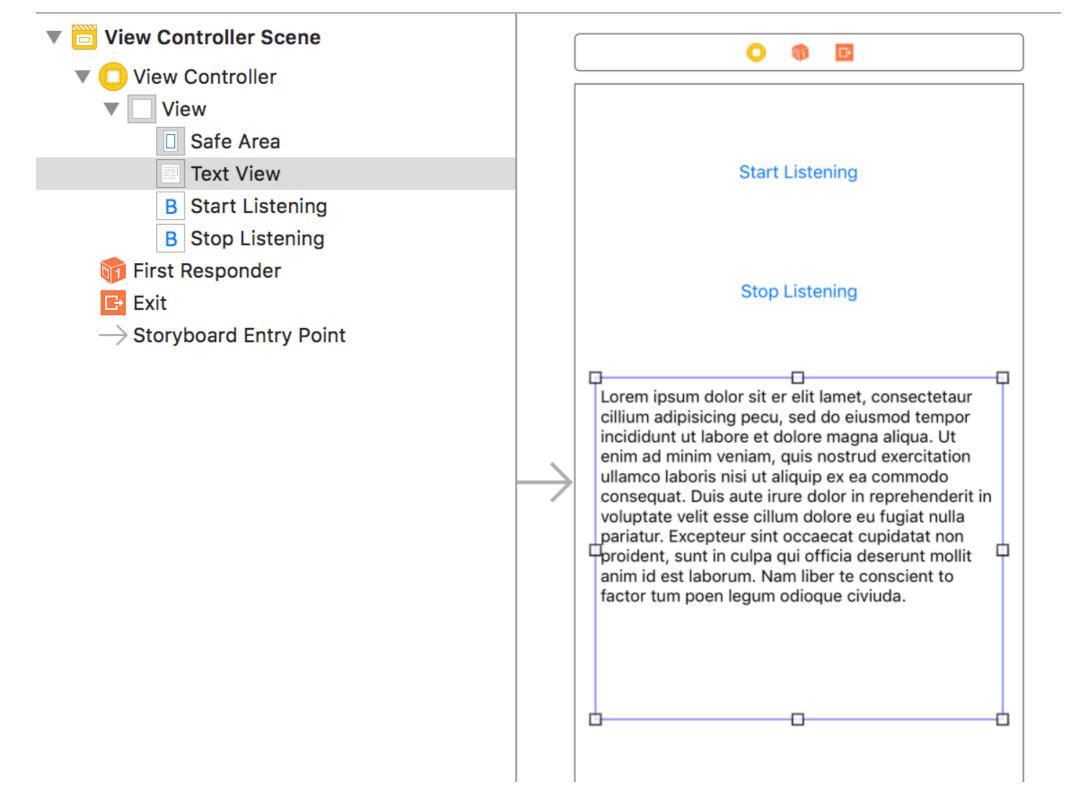
사용자 인터페이스 디자인

Main.storyboard 파일을 선택하고 <mark>단추와 텍스트 뷰</mark> 구성 요소 두 개를 장면에 추가하고 아래 그림 99-1과 같이 레이아웃이 나타나도록보기를 구성하고 배치합니다.



Button - Intercepts touch events and Button sends an action message to a target object when it's tapped.

Lorem ips conseuati aunc vital Nunc venj dui fringia **Text View** - Displays multiple lines of editable text and sends an action message to a target object when Ret...





🔻 📋 View Controller

▼ View

- □ Safe Area
- Text View
- **B** Start Listening
- **B** Stop Listening
- 👘 First Responder
- Exit
- → Storyboard Entry Point







Start Listening

Stop Listening

Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nam liber te conscient to factor tum poen legum odioque civiuda.

Selected Views
Update Constraint Constants
Add Missing Constraints
Poset to Suggested Constrain

Reset to Suggested Constraints
Clear Constraints

All Views in View Controller
Update Constraint Constants

Add Missing Constraints

Reset to Suggested Constraints

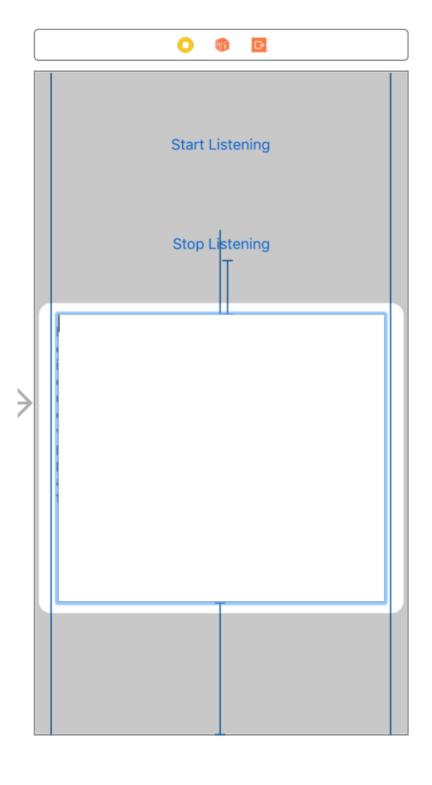
Clear Constraints

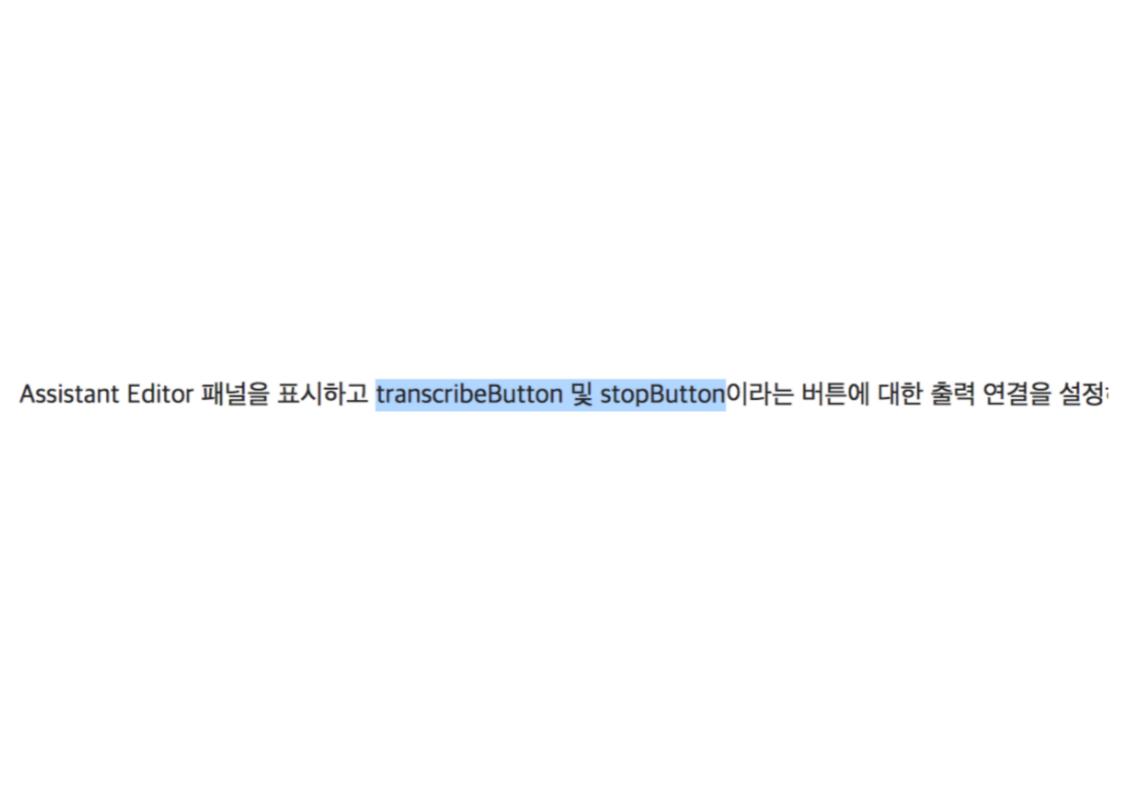




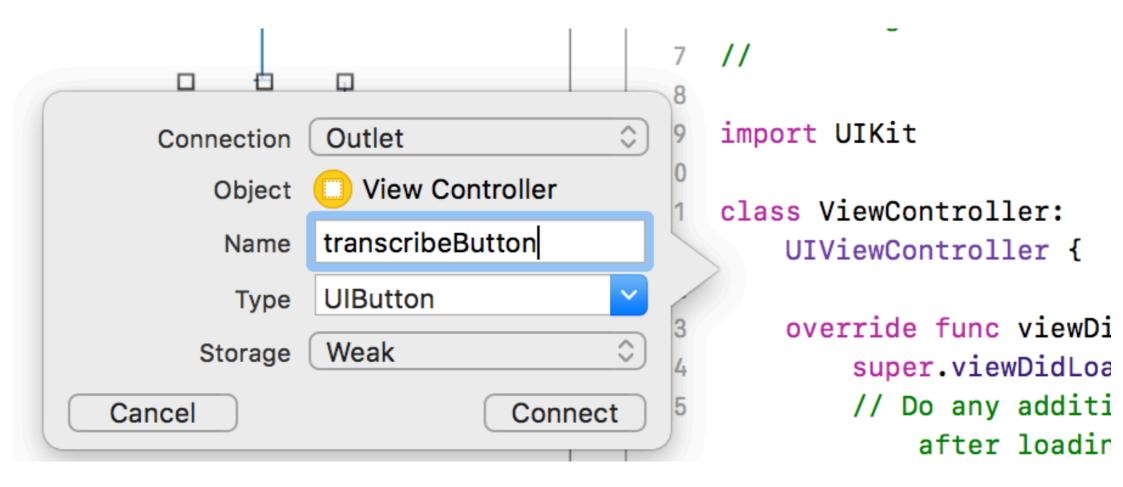


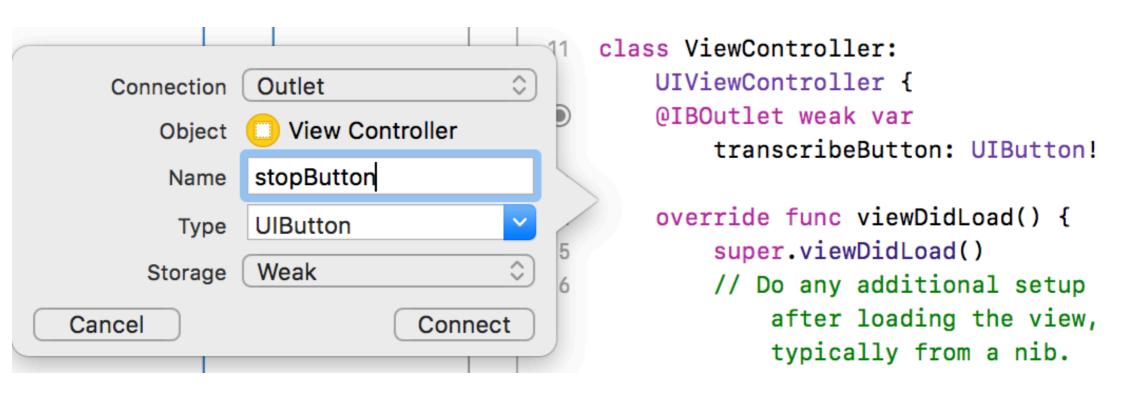
샘플 라틴 텍스트를 제거하십시오.





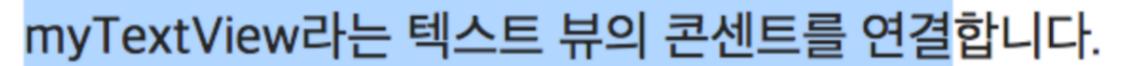






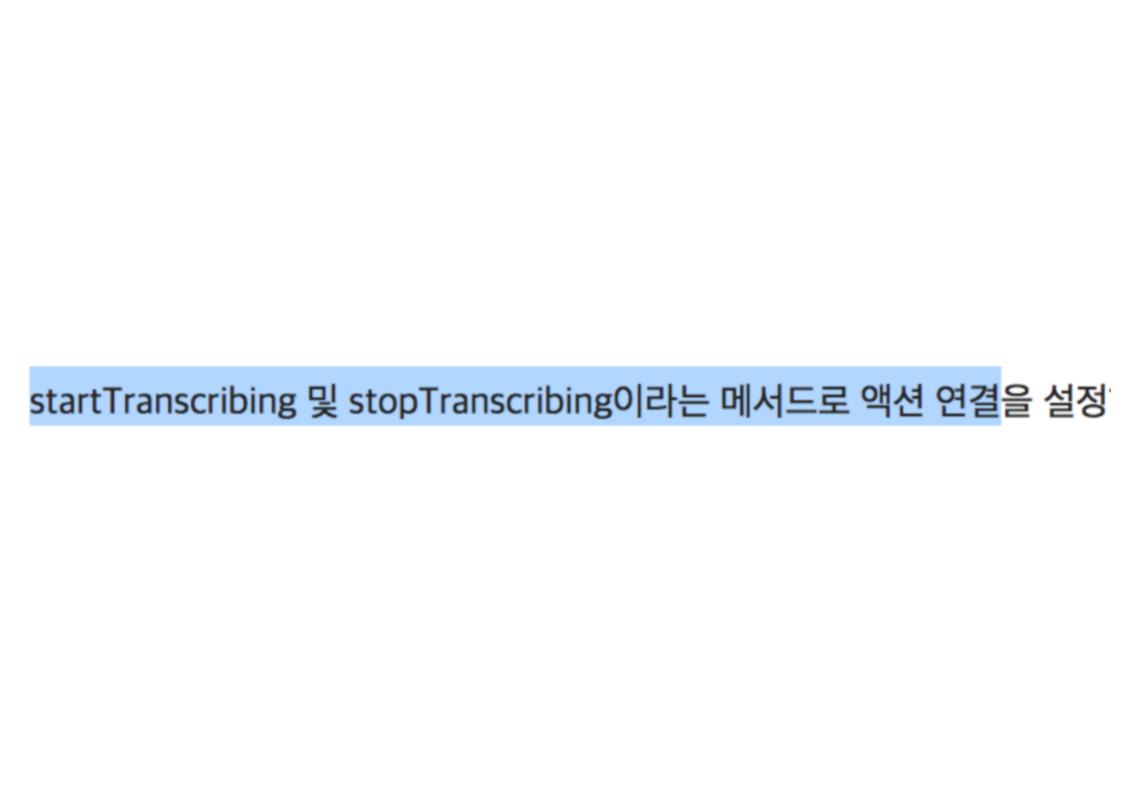
import UIKit

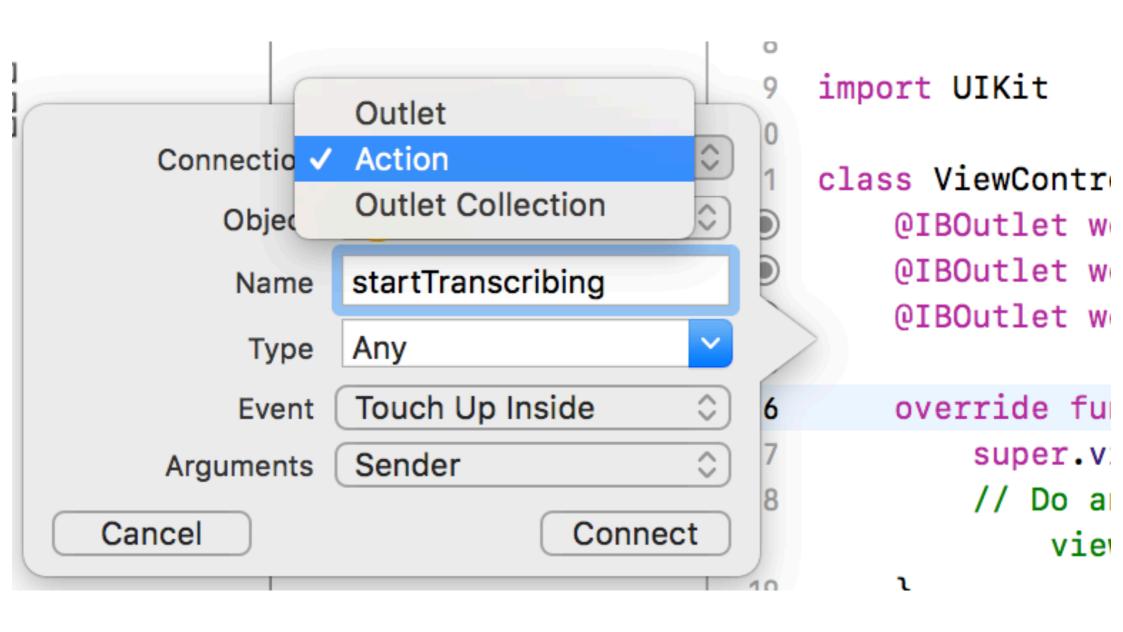
```
class ViewController: UIViewController {
    @IBOutlet weak var transcribeButton: UIButton!
    @IBOutlet weak var stopButton: UIButton!
```

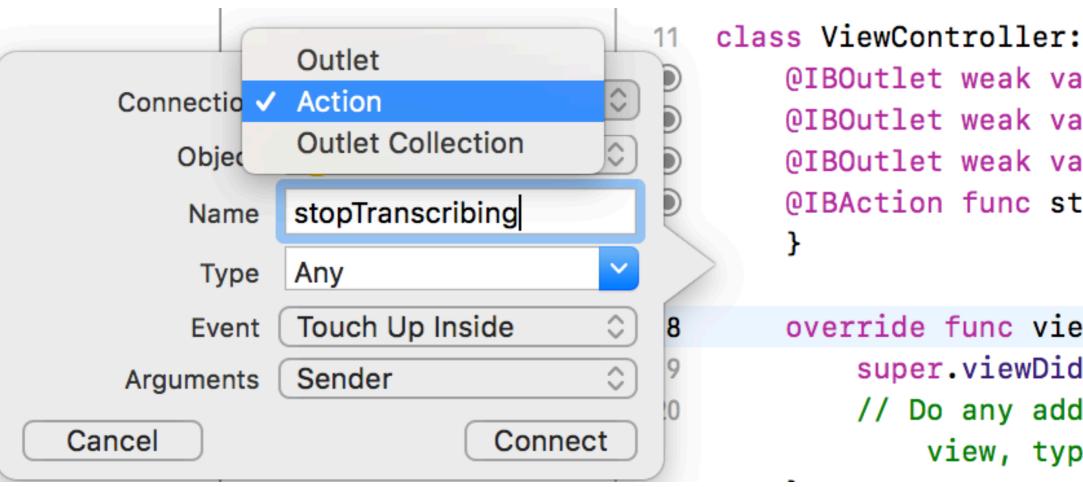




```
class ViewController: UIViewController {
    @IBOutlet weak var transcribeButton: UIButton!
    @IBOutlet weak var stopButton: UIButton!
    @IBOutlet weak var myTextView: UITextView!
```







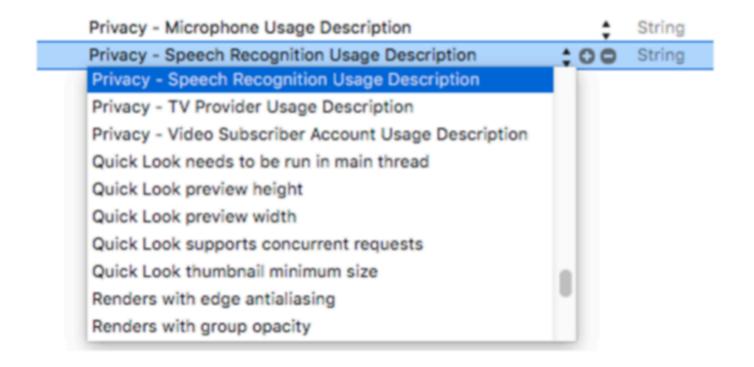
@IBOutlet weak var @IBOutlet weak var @IBOutlet weak var @IBAction func sta

> override func view super.viewDidL // Do any addi view, typi

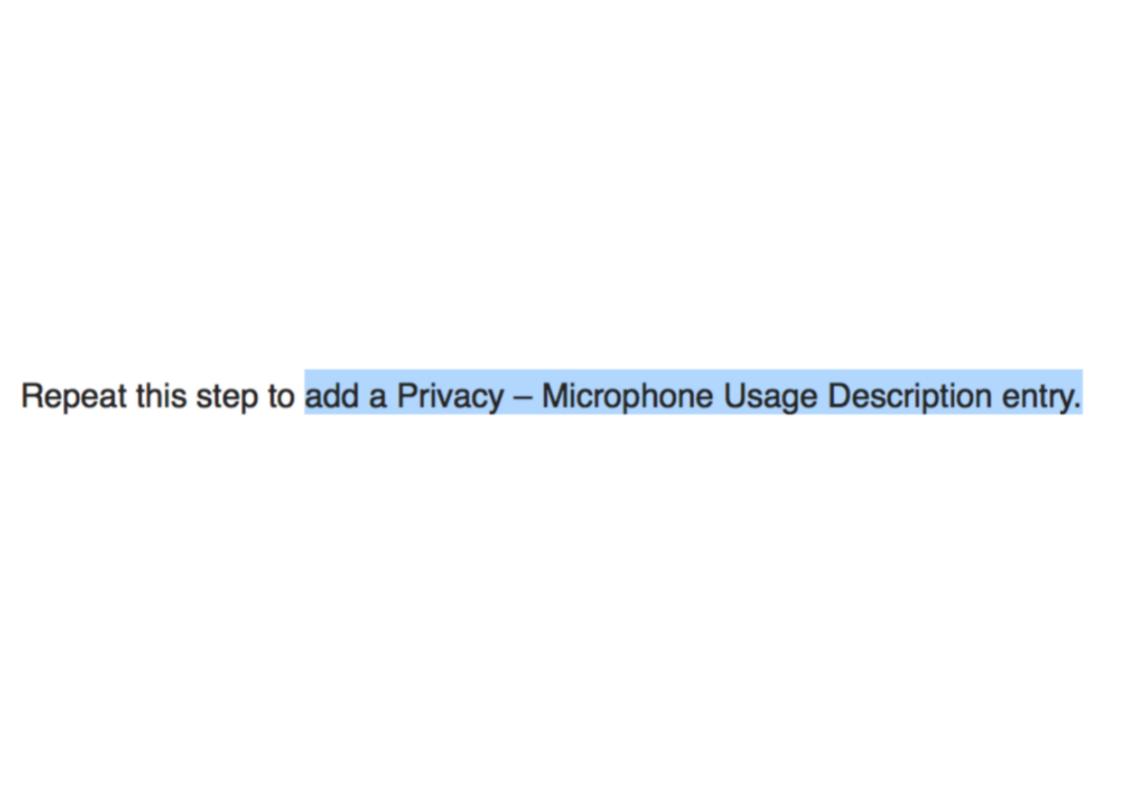
```
class ViewController: UIViewController {
    @IBOutlet weak var transcribeButton: UIButton!
    @IBOutlet weak var stopButton: UIButton!
    @IBOutlet weak var myTextView: UITextView!
    @IBAction func startTranscribing(_ sender: Any) {
    }
    @IBAction func stopTranscribing(_ sender: Any) {
}
```

음성 인식 권한 추가

Info.plist 파일을 선택하고 속성 목록에서 맨 아래 항목을 찾은 다음 마우스 포인터를 항목 위로 가져갑니다. 더하기 단추가 나타나면 해당 단추를 클릭하여 목록에 새 항목을 추가하십시오. 사용 가능한 키의 드롭 다운 목록에서 그림 99-2와 같이 개인 정보 - 음성 인식 사용 설명 옵션을 찾아 선택합니다.



속성의 값 필드 내에 음성 인식 사용 권한을 요청할 때 사용자에게 표시 할 메시지를 입력하십시오. 예
음성 인식 서비스는이 앱에서 음성을 텍스트로 변환하는 데 사용됩니다.



Supported interface orientations (iPad)	\$	Array	(4 items)
Privacy - Speech Recognition Usage Description	\$	String	음성 인식 서비스는 앱에서 음성을 텍스트로 변환하는데 사용됩니다.
Privacy - Microphone Usage Description	\$ 0 0	String	마이크를 사용하여 음성을 텍스트로 변환

음성 인식 권한 요청

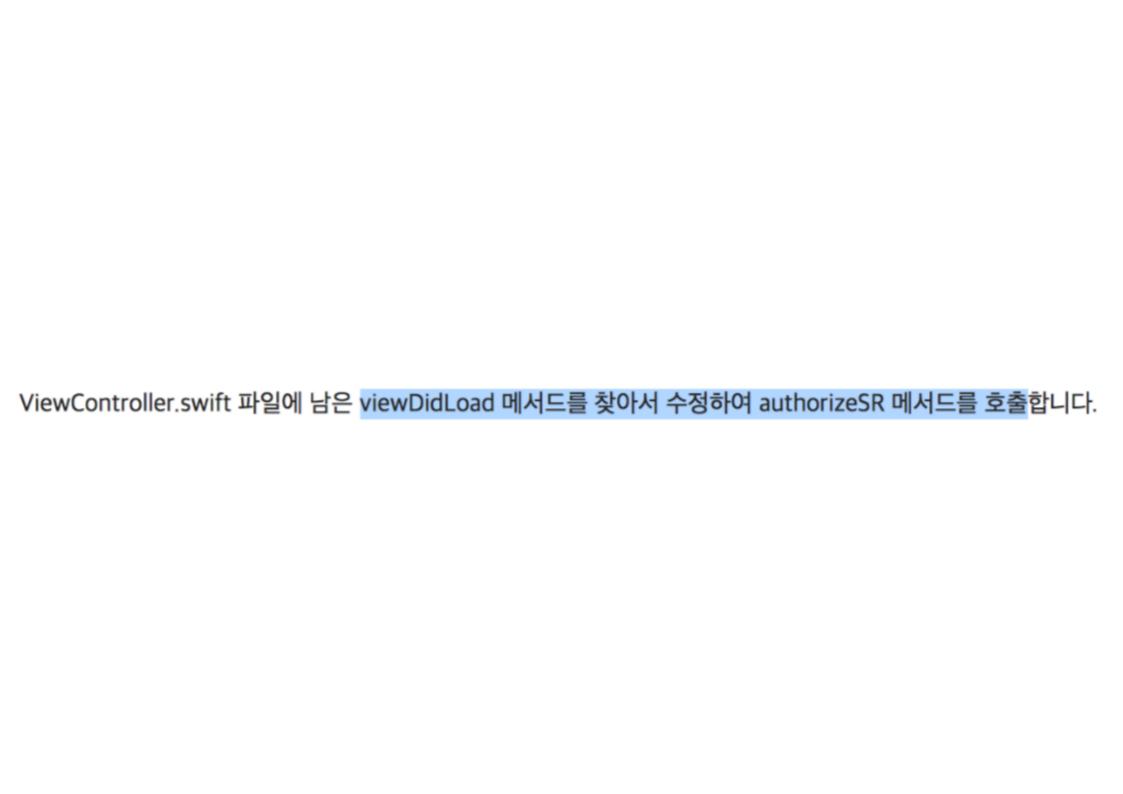
음성 인식 권한을 요청하는 코드는 이전 장의 코드와 동일합니다. 이

ViewController.swift 파일 내에 authorizeSR이라는 메서드로 추가

import UIKit
import Speech

class ViewController: UIViewController {

```
//이전 장의 코드와 동일
func authorizeSR() {
   //완료 핸들러로 지정된 클로저를 사용하여 SFSpeechRecognizer 클래스의 requestAuthorization 메소드를 호출합니다.
   SFSpeechRecognizer.requestAuthorization { authStatus in
       //이 핸들러에는 4 개의 값 (권한 부여, 거부, 제한 또는 결정되지 않음) 중 하나 일 수있는 상태 값이 전달됩니다.
       //그런 다음 switch 문을 사용하여 상태를 평가하고 기록 버튼을 활성화하거나 해당 버튼에 실패 원인을 표시합니다.
       OperationQueue.main.addOperation {
           switch authStatus {
           case .authorized:
               self.transcribeButton.isEnabled = true
           case .denied:
               self.transcribeButton.isEnabled = false
               self.transcribeButton.setTitle("Speech recognition access denied by user", for: .disabled)
           case .restricted:
               self.transcribeButton.isEnabled = false
               self.transcribeButton.setTitle("Speech recognition restricted on device", for: .disabled)
           case .notDetermined:
               self.transcribeButton.isEnabled = false
               self.transcribeButton.setTitle("Speech recognition not authorized", for: .disabled)
```



```
override func viewDidLoad() {
    super.viewDidLoad()
    authorizeSR()
}
```

음성 및 오디오 객체 선언 및 초기화

```
@IBOutlet weak var transcribeButton: UIButton!
@IBOutlet weak var stopButton: UIButton!
@IBOutlet weak var myTextView: UITextView!
//음성인식 객체
//실시간으로 말하기 기능을 구현하려면 앱에서 SFSpeechRecognizer, SFSpeechAudioBufferRecognitionRequest
//및 SFSpeechRecognitionTask 클래스의 인스턴스가 필요합니다.
//영어한글 모두 인식하고 싶다면
private let speechRecognizer = SFSpeechRecognizer()!
//영어만 인식하고 싶다면
//private let speechRecognizer = SFSpeechRecognizer(locale: Locale(identifier: "en-US"))!
private var speechRecognitionRequest: SFSpeechAudioBufferRecognitionRequest?
private var speechRecognitionTask: SFSpeechRecognitionTask?
//AVAudioEngine 인스턴스를 사용하여 오디오를 오디오 버퍼로 스트리밍
private let audioEngine = AVAudioEngine()
```

전사 시작

음성 인식을 시작하는 첫 번째 작업은 startTranscribing 동작 메서드에 코드를 추가하는 것입니다.

```
//startSession이라는 새 메서드 호출 추가
@IBAction func startTranscribing(_ sender: Any) {
    transcribeButton.isEnabled = false
    stopButton.isEnabled = true
    try! startSession()
}
```

```
//음성 인식을 수행하는 메소드 호출은 예외를 throw 할 가능성 있음
func startSession() throws {
   //startSession 메서드 내에서 수행되는 첫 번째 작업은 이전 인식 작업이 실행 중인지 확인하고, 그렇지 않으면 작업을 취소하는 것입니다.
   if let recognitionTask = speechRecognitionTask {
       recognitionTask.cancel()
       self.speechRecognitionTask = nil
   //또한이 메서드는 오디오 녹음 세션을 구성
   let audioSession = AVAudioSession.sharedInstance()
   try audioSession.setCategory(AVAudioSessionCategoryRecord)
   //이전에 선언 된 speechRecognitionRequest 변수에 SFSpeechAudioBufferRecognitionRequest 개체를 할당해야합니다.
   speechRecognitionRequest = SFSpeechAudioBufferRecognitionRequest()
   //그런 다음 SFSpeechAudioBufferRecognitionRequest 개체가 성공적으로 만들어 졌는지 확인하기위한 테스트가 수행됩니다.
   //생성이 실패한 경우 예외가 throw됩니다.
   guard let recognitionRequest = speechRecognitionRequest else
       { fatalError("SFSpeechAudioBufferRecognitionRequest object creation failed") }
```

```
guard let recognitionRequest = speechRecognitionRequest else
   { fatalError("SFSpeechAudioBufferRecognitionRequest object creation failed") }
//오디오 엔진의 inputNode에 대한 참조를 가져 와서 상수에 할당해야합니다.
//입력 노드를 사용할 수없는 경우 치명적인 오류가 발생합니다.
let inputNode = audioEngine.inputNode //else { fatalError("Audio engine has no input node") }
//recognitionRequest 인스턴스는 부분 결과를 반환하도록 구성되므로 음성 오디오가 버퍼에 도착할 때 계속해서 녹음을 수행 할 수 있습니다.
//이 속성이 설정되지 않은 경우 앱은 녹음 과정을 시작하기 전에 오디오 세션이 끝날 때까지 대기합니다.
recognitionRequest.shouldReportPartialResults = true
//인식 작업이 초기화됩니다.
//인식 요청 객체로 초기화 된 인식 태스크를 생성합니다.
//클로저는 완성 된 텍스트의 각 블록이 완료 될 때 반복적으로 호출되는 완료 핸들러로 지정됩니다.
//처리기가 호출 될 때마다 오류 개체와 함께 최신 버전의 텍스트가 포함 된 결과 개체가 전달됩니다.
speechRecognitionTask = speechRecognizer.recognitionTask(with: recognitionRequest) { result, error in
   var finished = false
   //결과 객체의 isFinal 속성이 false (라이브 오디오가 여전히 버퍼로 스트리밍됨을 나타냄) 오류가 발생하지 않는 한 텍스트는 텍스트보기에
       표시됩니다.
   if let result = result {
       self.myTextView.text =
          result.bestTranscription.formattedString
       finished = result.isFinal
   }
   //그렇지 않으면 오디오 엔진이 중지되고 탭이 오디오 노드에서 제거되고 인식 요청 및 인식 작업 객체가 nil로 설정됩니다.
   //기록 단추는 다음 세션 준비를 위해 사용할 수도 있습니다.
   if error != nil || finished {
       self.audioEngine.stop()
       inputNode.removeTap(onBus: 0)
       self.speechRecognitionRequest = nil
       self.speechRecognitionTask = nil
       self.transcribeButton.isEnabled = true
   }
```

```
self.transcribeButton.isEnabled = true
}
//인식 작업을 구성하고 나면 이 단계에서 남은 것은 오디오 엔진의 입력 노드에 탭을 설치 한 다음 실행중인 엔진을 시작하는 것입니다.
//inputNode 객체의 installTap 메서드는 클로저를 완료 핸들러로 사용합니다.
//이 핸들러의 코드는 호출 될 때마다 최신 오디오 버퍼를 speechRecognitionRequest 객체에 추가하기 만하면 자동으로 녹음되고
//텍스트 인식 뷰에 표시되는 음성 인식 작업의 완료 핸들러로 전달됩니다.
let recordingFormat = inputNode.outputFormat(forBus: 0)
inputNode.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer: AVAudioPCMBuffer, when:
   AVAudioTime) in
   self.speechRecognitionRequest?.append(buffer)
}
audioEngine.prepare()
try audioEngine.start()
```

```
@IBAction func stopTranscribing(_ sender: Any) {
    //오디오 엔진을 중지하고 다음 세션에 사용할 준비가 된 버튼의 상태를 구성
    if audioEngine.isRunning {
        audioEngine.stop()
        speechRecognitionRequest?.endAudio()
        transcribeButton.isEnabled = true
        stopButton.isEnabled = false
    }
}
```

앱 테스트

실제 iOS 기기에서 앱을 컴파일하고 실행하고 마이크에 대한 액세스 권한을 부여하고 음성 인식을 사용할 수있는 권한을 부여한 다음 전송 시작 버튼을 탭합니다. 장치에 말을하고 오디오가 텍스트보기로 옮겨 지는지 관찰하십시오. 세션을 끝내려면 언제든지 전송 중지 버튼을 누릅니다.









► Li...ch 〉 ■ YOUNG SIK의 iPad