



广告

汽车之家疯狂购车节

买车就来汽车之家,享专属红包,还有豪华大礼赠送! 转到 1.autohome.com.cn



iOS 10中如何搭建一个语音转文字框架

2016-10-24 09:38 编辑: suiling 分类: iOS开发 来源: CocoaChina翻译组

2 2427

Speech framework

招聘信息: cocos2d-x游戏主程



- 原文: [Building a Speech-to-Text App Using Speech Framework in iOS 10](#)
- 作者: [Sahand Edrisian](#)
- 译者: 王跃

在2016WWDC大会上, Apple公司介绍了一个很好的语音识别的API,那就是Speech framework。事实上, 这个Speech Kit就是Siri用来做语音识别的框架。如今已经有一些可用的语音识别框架, 但是它们要么太贵要么不好。在今天的教程里面, 我会教你怎样创建一个使用Speech Kit来进行语音转文字类似Siri的app。

设计App UI

前提: 你需要Xcode 8 beta版本和一个运行iOS 10 beta系统版本的iOS 设备。

先从创建一个新的命名为SpeechToTextDemo的单视图工程开始。接下来, 到 Main.storyboard 中添加一个 UILabel, 一个 UITextView, 和一个 UIButton, 你的storyboard应该看起来如下图:

热门资讯



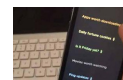
Xcode 8带来的新特性和坑
点击量 13331



21个高质量的Swift开源iOS App
点击量 8720



iOS开发--适配iOS 10以及Xcode 8
点击量 6173



iOS10推送必看 UNNotificationServiceExtension
点击量 6106



谈UIView Animation编程艺术
点击量 5548



不一样的角度 解读微信小程序
点击量 4679



程序员如何保持健康
点击量 4093



我的三次iOS面试经验
点击量 3964



Swift 3 迁移工作总结
点击量 3742



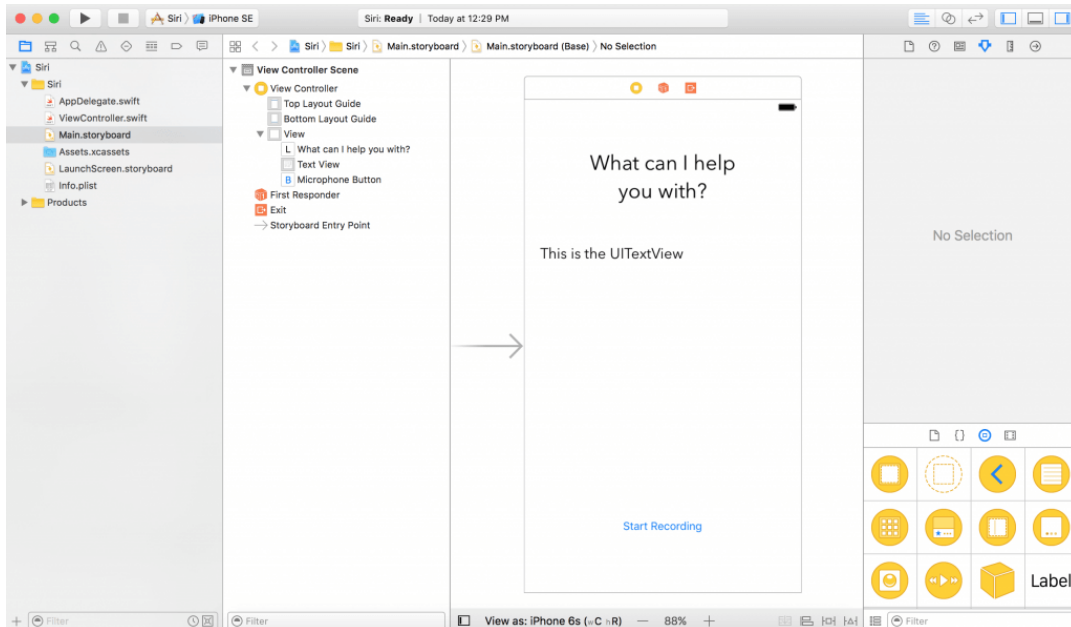
苹果确认Macbook Pro 2016发布时间
点击量 3454

综合评论

不错, 例子很生动, 容易理解
shr871217 评论了 协议和 NSProxy 实现多继承...

mvvm 是基于mvc, 主要功能是将控制器在网络请求返回数据转换成我们需要
zij_zij 评论了 做一个 App 前需要考虑的几件事...

好思路
lx125z 评论了 用CollectionView封装一个广告视图...



接下来在 ViewController.swift文件中为UITextView 和UIButton 定义outlet变量。在这个demo当中，我设置UITextView 的名称为“textView”，UIButton的名称为“microphoneButton”。然后创建一个当microphone按钮被点击时会触发的空的按钮执行方法。

```
1  @IBAction func microphoneTapped(_ sender: AnyObject) {
2
3  }
```

如果你不想从创建最原始工程开始，你可以在 [在这里下载原始工程](#) 然后继续下面的教学指导。

使用Speech Framework

为了使用Speech framework, 你必须首先导入它然后遵循 SFSpeechRecognizerDelegate 协议。因此让我们导入这个框架，然后在 ViewController 文件中加上它的协议。现在你的 ViewController.swift 文件应该如下图所示：

```
1  import UIKit
2  import Speech
3
4  class ViewController: UIViewController, SFSpeechRecognizerDelegate {
5
6
7      @IBOutlet weak var textView: UITextView!
8      @IBOutlet weak var microphoneButton: UIButton!
9
10     override func viewDidLoad() {
11         super.viewDidLoad()
12
13     }
14
15     @IBAction func microphoneTapped(_ sender: AnyObject) {
16
17     }
18
19 }
```

用户授权

在使用speech framework做语音识别之前，你必须首先得到用户的允许，因为不仅仅只有本地的ios设备会进行识别，苹果的服务器也会识别。所有的语音数据都会被传递到苹果的后台进行处理。因此，获取用户授权是强制必须的。

让我们在 viewDidLoad 方法里授权语音识别。用户必须允许app使用话筒和语音识别。首先，声明一个 speechRecognizer变量：

好好好

lx125z 评论了 Xcode8调试黑科技：Memory Graph实...

cc 质量回升？？

JYW129 评论了 iOS项目iCloud及 CloudKit Dash...

讲的是对的，但是例子起的不是很好，因为child跟person也有一定的继承关系
kabda 评论了 协议和 NSProxy 实现多继承...

属性copy的block是不是就是“定义全局变量的地方有block语法时”这种情况
liaoxiaofan 评论了 Block内存管理实例分析...

swift3.0更新了

njlszs 评论了 Swift—高仿单糖App【附源码】...

太装逼了

Lindon1228 评论了 协议和 NSProxy 实现多继承...

这篇文章去年就看过的

conse 评论了 轻量级的运营，口碑爆棚的产品，5000万用户的ke...

相关帖子

关于swift3中的String

cocos2dx 实现mac 上的 magic effect (神奇效果)

身份证ocr识别 sdk

关于NSXML解析,如果有不同的URL, 格式不一样怎么解析呢

xcode在release下真机调试报错问题

ios10 ReplayKit录屏功能问题

求教MVC模式下的model数据处理

《江苏省南京市五星宏辉符号单草花机》13728222140

《江苏省南京市五星宏辉符号单对单》

```
1 private let speechRecognizer = SFSpeechRecognizer(locale: Locale.init(identifier: "en-US"))
```

然后如下图更新 viewDidLoad 方法：

```
1 override func viewDidLoad() {
2     super.viewDidLoad()
3
4     microphoneButton.isEnabled = false //2
5
6     speechRecognizer.delegate = self //3
7
8     SFSpeechRecognizer.requestAuthorization { (authStatus) in //4
9
10        var isEnabled = false
11
12        switch authStatus { //5
13        case .authorized:
14            isEnabled = true
15
16        case .denied:
17            isEnabled = false
18            print("User denied access to speech recognition")
19
20        case .restricted:
21            isEnabled = false
22            print("Speech recognition restricted on this device")
23
24        case .notDetermined:
25            isEnabled = false
26            print("Speech recognition not yet authorized")
27        }
28
29        OperationQueue.main.addOperation() {
30            self.microphoneButton.isEnabled = isEnabled
31        }
32    }
33 }
```

1. 首先，我们创建一个带有标识符en-US 的 SFSpeechRecognizer实例，这样语音识别API就能知道用户说的是哪一种语言。这个实例就是处理语音识别的对象。
2. 我们默认让microphone按钮失效直到语音识别功能被激活。
3. 接下来，把语音识别的代理设置为 self 也就是我们的ViewController。
4. 之后，我们必须通过调用SFSpeechRecognizer.requestAuthorization方法来请求语音识别的授权。
5. 最后，检查验证的状态。如果被授权了，让microphone按钮有效。如果没有，打印错误信息然后让microphone按钮失效。

现在如果你认为app跑起来之后你会看到一个授权弹出窗口，那你就错了。如果运行，app会崩溃。好吧，既然知道结果为什么还要问呢？（别打我），看看下面解决方法。

提供授权消息

苹果要求app里所有的授权都要一个自定义的信息。例如语音授权，我们必须请求2个授权：

1. 麦克风使用权。
2. 语音识别。

为了自定义信息，你必须在info.plist 配置文件里提供这些自定义消息。

让我们打开 info.plist配置文件的源代码。首先，右键点击 info.plist。然后选择Open As > Source Code。最后，拷贝

下面的XML代码然后在 `</dict>` 标记前插入这段代码。

```
<key>NSMicrophoneUsageDescription</key> <string>Your microphone will be used to record your speech when you press the "Start Recording" button.</string>
<key>NSSpeechRecognitionUsageDescription</key> <string>Speech recognition will be used to determine which words you speak into this device's microphone.</string>
```

微博



CocoaChina

加关注

【做一个 App 前需要考虑的几件事】
随着工具链的完善，语言的升级以及各种优质教程的涌现，做一个 App 的成本也越来越低了。尽管如此，有些事情最好前期就做起来，避免当 App 有了一定规模后，再感慨当初为什么没有多留点心。（via: limboy 的博客）
<http://t.cn/RVjcYwE>



10月25日 14:40 转发(8) | 评论(8)

【Block内存管理实例分析】（作者：tree_one）说道block大家都不陌生，内存管理问题也是开发者最头疼的问题，网上很多讲block的博客，但大都是理论性多点，本文结合一些实例来

100offer
高端人才招聘网站

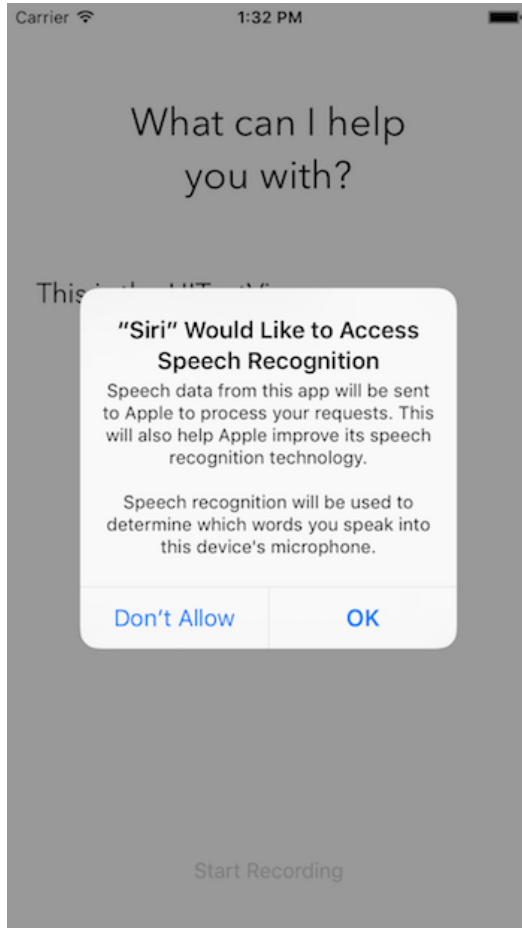
告别盲目投简历
让海量好机会主动来找你

www.100offer.com

现在你已经在info.plist文件里添加了两个键值:

- NSMicrophoneUsageDescription - 为获取麦克风语音输入授权的自定义消息。注意这个语音输入授权仅仅只会在用户点击microphone按钮时发生。
- NSSpeechRecognitionUsageDescription – 语音识别授权的自定义信息

可以自行更改这些消息的内容。现在点击Run按钮, 你应该可以编译和成功运行app了, 不会报任何错误。



注意: 如果稍后在工程运行完成时还没有看到语音输入授权框, 那是因为你是模拟器上运行的程序。iOS模拟器没有权限进入你Mac电脑的麦克风。

处理语音识别

现在我们已经实现了用户授权, 我们现在去实现语音识别功能。先从在 ViewController里定义下面的对象开始:

```
1 private var recognitionRequest: SFSpeechAudioBufferRecognitionRequest?
2 private var recognitionTask: SFSpeechRecognitionTask?
3 private let audioEngine = AVAudioEngine()
```

1. recognitionRequest对象处理了语音识别请求。它给语音识别提供了语音输入。
2. recognition task对象告诉你语音识别对象的结果。拥有这个对象很方便因为你可以用它删除或者中断任务。
3. audioEngine是你的语音引擎。它负责提供你的语音输入。

接下来, 创建一个新的方法名叫 startRecording()。

```
1 func startRecording() {
2
3     if recognitionTask != nil {
4         recognitionTask?.cancel()
```

```

5         recognitionTask = nil
6     }
7
8     let audioSession = AVAudioSession.sharedInstance()
9     do {
10         try audioSession.setCategory(AVAudioSessionCategoryRecord)
11         try audioSession.setMode(AVAudioSessionModeMeasurement)
12         try audioSession.setActive(true, with: .notifyOthersOnDeactivation)
13     } catch {
14         print("audioSession properties weren't set because of an error.")
15     }
16
17     recognitionRequest = SFSpeechAudioBufferRecognitionRequest()
18
19     guard let inputNode = audioEngine.inputNode else {
20         fatalError("Audio engine has no input node")
21     }
22
23     guard let recognitionRequest = recognitionRequest else {
24         fatalError("Unable to create an SFSpeechAudioBufferRecognitionRequest object")
25     }
26
27     recognitionRequest.shouldReportPartialResults = true
28
29     recognitionTask = speechRecognizer.recognitionTask(with: recognitionRequest, result
30
31         var isFinal = false
32
33         if result != nil {
34
35             self.textView.text = result?.bestTranscription.formattedString
36             isFinal = (result?.isFinal)!
37         }
38
39         if error != nil || isFinal {
40             self.audioEngine.stop()
41             inputNode.removeTap(onBus: 0)
42
43             self.recognitionRequest = nil
44             self.recognitionTask = nil
45
46             self.microphoneButton.isEnabled = true
47         }
48     })
49
50     let recordingFormat = inputNode.outputFormat(forBus: 0)
51     inputNode.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer
52         self.recognitionRequest?.append(buffer)
53     )
54 }
55
56 audioEngine.prepare()
57
58 do {
59     try audioEngine.start()
60 } catch {
61     print("audioEngine couldn't start because of an error.")
62 }
63
64 textView.text = "Say something, I'm listening!"
65 }

```

这个方法会在Start Recording按钮被点击时调用。它主要功能是开启语音识别然后聆听你的麦克风。我们一行行分析上面的代码：

- 3-6行 – 检查 recognitionTask 是否在运行。如果在就取消任务和识别。
- 8-15行 – 创建一个 AVAudioSession来为记录语音做准备。在这里我们设置session的类别为recording，模式为 measurement，然后激活它。注意设置这些属性有可能会抛出异常，因此你必须把他们放入try catch语句里面。
- 17行 – 实例化recognitionRequest。在这里我们创建了SFSpeechAudioBufferRecognitionRequest对象。稍后我们利用它把语音数据传到苹果后台。
- 19-21行 – 检查 audioEngine（你的设备）是否有做录音功能作为语音输入。如果没有，我们就报告一个错误。
- 23-25行 – 检查recognitionRequest对象是否被实例化和不是nil。
- 27行– 当用户说话的时候让recognitionRequest报告语音识别的部分结果。
- 29行 – 调用 speechRecognizer的recognitionTask 方法来开启语音识别。这个方法有一个completion handler回调。这个回调每次都会在识别引擎收到输入的时候，完善了当前识别的信息时候，或者被删除或者停止的时候被

调用，最后会返回一个最终的文本。

- 31行 – 定义一个布尔值决定识别是否已经结束。
- 35行 – 如果结果 result 不是nil, 把 textView.text 的值设置为我们的最优文本。如果结果是最终结果，设置 isFinal 为true。
- 39-47行 – 如果没有错误或者结果是最终结果，停止 audioEngine(语音输入)并且停止 recognitionRequest 和 recognitionTask.同时，使Start Recording按钮有效。
- 50-53行 – 向 recognitionRequest增加一个语音输入。注意在开始了recognitionTask之后增加语音输入是OK的。Speech Framework 会在语音输入被加入的同时就开始进行解析识别。
- 55行 – 准备并且开始audioEngine。

触发语音识别

我们需要保证当创建一个语音识别任务的时候语音识别功能是可用的，因此我们必须给ViewController添加一个代理方法。如果语音输入不可用或者改变了它的状态，那么 microphoneButton.enable属性就要被设置。针对这种情况，我们实现了SFSpeechRecognizerDelegate 协议的 availabilityDidChange 方法。实现内容看下面：

```
1 func speechRecognizer(_ speechRecognizer: SFSpeechRecognizer, availabilityDidChange available: Bool) {
2     if available {
3         microphoneButton.isEnabled = true
4     } else {
5         microphoneButton.isEnabled = false
6     }
7 }
```

这个方法会在可用性状态改变时被调用。如果语音识别可用，那么记录按钮record会被设为可用状态。

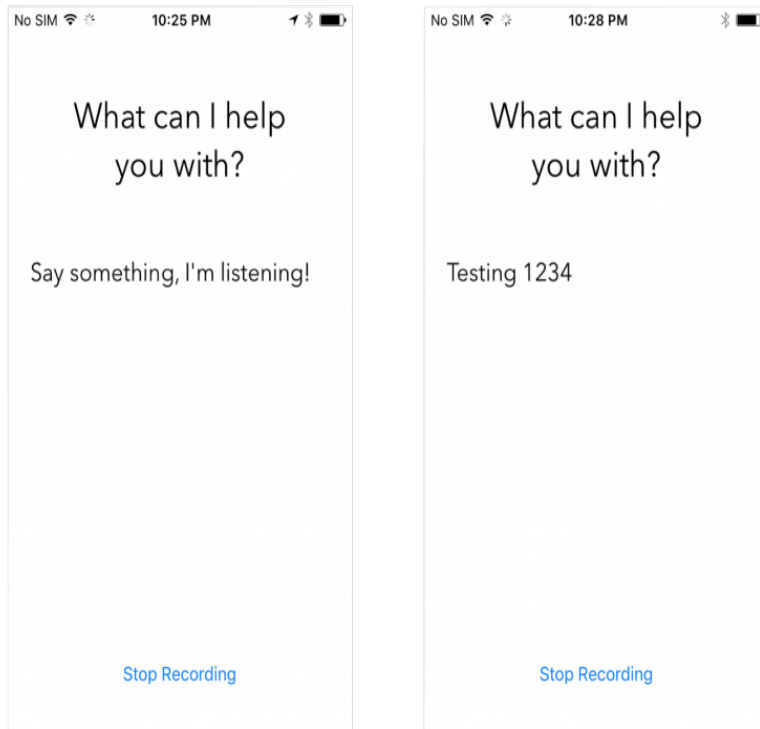
最后一件事就是我们必须更新响应方法microphoneTapped(sender):

```
1 @IBAction func microphoneTapped(_ sender: AnyObject) {
2     if audioEngine.isRunning {
3         audioEngine.stop()
4         recognitionRequest?.endAudio()
5         microphoneButton.isEnabled = false
6         microphoneButton.setTitle("Start Recording", for: .normal)
7     } else {
8         startRecording()
9         microphoneButton.setTitle("Stop Recording", for: .normal)
10    }
11 }
```

在这个方法中，我们必须检查 audioEngine是否正在工作。如果是，app应该停止 audioEngine, 中止向 recognitionRequest输入音频，让microphoneButton按钮不可用，并且设置按钮的标题为 “Start Recording”

如果 audioEngine 正在工作，app应该调用 startRecording() 并且设置按钮的标题为 “Stop Recording”。

非常好！现在可以准备测试app了。把app部署到一个iOS10的设备，然后点击“Start Recording”按钮。去说些什么吧！



注意:

1. 苹果公司对每个设备的识别功能都有限制。具体的限制并不知道，但是你可以联系苹果公司了解更多信息。
2. 苹果公司对每个app也有识别功能限制。
3. 如果你经常遇到限制，请一定联系苹果公司，他们应该可以解决问题。
4. 语音识别会很耗电以及会使用很多数据。
5. 语音识别一次只持续大概一分钟时间。

总结

在这个教程中，你学习到了怎样好好的利用苹果公司开放给开发者的惊人的新语言API，用于语音识别并且转换到文本。Speech framework 使用了跟Siri相同的语音识别框架。这是一个相对小的API。但是，它非常强大可以让开发者们开发非凡的应用比如转换一个语音文件到文本文字。

我推荐你看[WWDC 2016 session 509](#)去获取更多有用信息。希望你喜欢这篇文章并且在探索这个全新API中获得乐趣。

作为参考，你可以[在这里查看Github完整工程](#)



微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号：CocoaChina

我要投稿

收藏文章

分享到:

6

上一篇: iOS10适配: 被弃用的openURL

下一篇: Xcode 8带来的新特性和坑

相关资讯



我来说两句



你怎么看? 快来评论一下吧!

发表评论

所有评论 (2)



508037051@qq.com

10分钟前

IOS、Swift、APP开发学习交流.群.号 579501755 群.内提供代码库, 分享学习资料文档

0 0 回复



cocosoeasy

2016-10-24 10:41:55

科大讯飞.....

0 0 回复

[关于我们](#) [商务合作](#) [联系我们](#) [合作伙伴](#)

北京触控科技有限公司版权所有

©2016 Chukong Technologies, Inc.

京ICP备 11006519号 京ICP证 100954号 京公网安备11010502020289



京网文[2012]0426-138号