Abstract

The purpose of this document to introduce about the application **EZ Time Tracker** integrated with **Siri,** which is Apple's virtual personal assistant and can be used for a long list of tasks, such as making phone calls, sending messages, identifying songs, downloading apps, changing device settings, searching the web, finding movie and restaurant reviews, making dinner reservations, creating reminders and calendar events, calculating tips, and more. **EZ Time tracker App** is made basically for time logging according to project and task within any organizational Task premise. This Article is a discussion about integration of both platforms in a technical way.

Integration of SiriKit

Main task was to integrate Siri in TimeTracker app and the purpose was to make it easy for users to add, start or stop tasks through interacting Siri i.e. user doesn't have to open the app but he can manage/interact with these tasks directly from Siri, which is a computer program that works as an intelligent personal assistant and information navigator, part of Apple Inc.'s iOS, watch OS, macOS, and tvOS operating systems. The feature uses a natural language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of Web services.

SiriKit support is divided into specific domains i.e.

- VoIP
- Messaging
- Payments
- Photos
- Workouts
- Ride booking
- Car commands
- CarPlay
- Restaurants

And unfortunately Time Tracker app's domain is "productivity" which doesn't come under SiriKit supported domains, and this was the biggest challenge for us.

So another solution was to enable speech to text functionality within app with custom command based scenarios. There are many libraries are available for SpeechToText but most of them are paid. By searching more on this found that Siri natively uses **SpeechKit** framework for speech recognition. And for this we've successfully created a PoC for this which converts SpeechToText. The **SpeechKit** framework provides the classes necessary to perform network-based speech recognition

and text-to-speech synthesis. This framework provides a simple, high-level speech service API that automatically performs all the tasks necessary for speech recognition or synthesis, including audio recording, audio playback, and network connection management.

Similarly, we can convert Text to Speech using **AVSpeechSynthesizer.** The **AVSpeechSynthesizer** class produces synthesized speech from text on an iOS device, and provides methods for controlling or monitoring the progress of ongoing speech. Whereas, Introduced in iOS 7, **AVSpeechSynthesizer** produces synthesized speech from a given **AVSpeechUtterance.** Each utterance can adjust its rate of speech and pitch, and be configured to use any one of the available **AVSpeechSynthesisVoices.**

Speechkit Features

- Will be used for SpeechToText feature

- Supported in iOS 10+
- Native component, embedded in iOS SDK
- Works only when connected with internet
- Works online and used for SPEECH to TEXT and third party libs can be reviewed for offline mode.

AVSpeechSynthesizer Features

- Will be used for TextToSpeech feature

- Supported in IOS 7+
- Native Component, embedded in IOS SDK
- Works in offline mode
- Works offline and used for SPEECH to TEXT and third party libs can be reviewed for offline mode

Speechkit Requirements

SpeechKit requires iOS 10+ and requires internet in order to recognize voice and convert it to text. The main benefit of using this SDK is that it will give 95% success result, so user may not have to speak again and again. For offline mode, There is some open source SPEECH to TEXT APIs available but there result of recognize the speech is not good.

Running Environments

SpeechKit framework uses internet for processing because the recognition doesn't happen just locally on the iOS device but Apple's servers. All the voice data are transmitted to Apple's backend for processing. However **AVSpeechSynthesizer** can be used locally or in offline mode or we can say **AVSpeechSynthesizer** is a built-in API in iOS SDK that works offline. We can use it for TEXT to SPEECH feature. Where we can provide 'text' and use **AVSpeechSynthesizer** to utter.