Cole Hemp

COP 2654

4/29/19

Siri Shortcuts

The year is 2008, and you're watching Robert Downy Jr. star in his first appearance of his notable role as Tony Stark. He comes into his workshop and starts talking to his A.I. assistant, J.A.R.V.I.S., and you think how cool it would be to have a piece of technology function as your assistant. Jump forward two years to 2010, when Siri was first released onto the IOS ecosystem. This was it. The J.A.R.V.I.S. that everyone saw and wanted from a few years' prior was here, except it wasn't. At most, Siri was a novelty to most of its users. It was used for random facts and users testing its knowledge, but not much outside of that. Shortly after its release, Apple acquired Siri and continued its development, but it never really got to the level of effectiveness that J.A.R.V.I.S. showed in the first Iron Man movie. That was until this previous year at WDCC where Apple announced a feature that could potentially expand the usefulness of Siri tenfold.

At WDCC 2018, Apple announced the functionality of Siri Shortcuts for Swift and Xcode. What makes this functionality so different over previous years is that it allows developers to integrate their apps into Siri's functionality. The way they do this is by creating app shortcuts for Siri to utilize. App shortcuts function like an API, a way for Siri to access a specific function inside of an app. The example the presenter gave was allowing Siri to ask the end user if they wanted to order their preferred soup from a soup ordering app. The shortcut creation works in three steps; defining the shortcut, donating the shortcut, and handling the shortcut. Defining the shortcut is where the developer creates the parameters and the functions revolving a specific shortcut. Donating a shortcut is where the user creates an instance of the shortcut by naturally using the app, and Siri monitors the user's access to the app in order to determine the best time to use said shortcut. Handling a shortcut is the final step where the shortcut is executed when Siri's determines the requirements to run the shortcut is met and the user accepts. To work with shortcuts, Apple has released two APIs in order to integrate your apps with Siri and Siri Shortcuts.

The first API is called NSUserActivity. NSUserActivity creates a shortcut that opens something in your app. It runs every time the user is in the associated app, and it creates the handler for NSUserActivity to monitor user activity to determine the best content and time to show the shortcut. An example use case of this would be to show a shortcut to a potentially interesting news article on the user's news app based on articles they've read in the past. The second API is called Intents. Intents offers better support and a more in-depth way of integrating your app with Siri. The way it works is by creating a new SiriKit Intent Definition File in your project, filling out the meta data, the parameters, and the shortcut types. The parameters and shortcut types are used to display the shortcut and information to the end user. It is generally the most efficient to create shortcut types of as many variations that are useful and to allow as many shortcuts as reasonably possible background execution. This will allow Siri to run the best possible shortcut given the parameters it currently has access to. Also, Siri is also more likely to choose shortcuts that allow background execution if possible. Overall, Intents and NSUserActivity allow the developer to display app information and app functions to the end user through Siri and Siri Shortcuts.

To Conclude, Apple introduced Siri Shortcuts this previous WDCC. Siri Shortcuts is a way for developers to integrate their apps with Siri using either the NSUserActivity API or the Intents API. Even though there's not much for app integration yet with Siri Shortcuts, I can see it in three to five years as a way for Siri to become as practical and advance as J.A.R.V.I.S was in the first Iron Man movie.

References

Apple Inc. (n.d.). Retrieved April 29, 2019, from https://developer.apple.com/videos/play/wwdc2018/211/