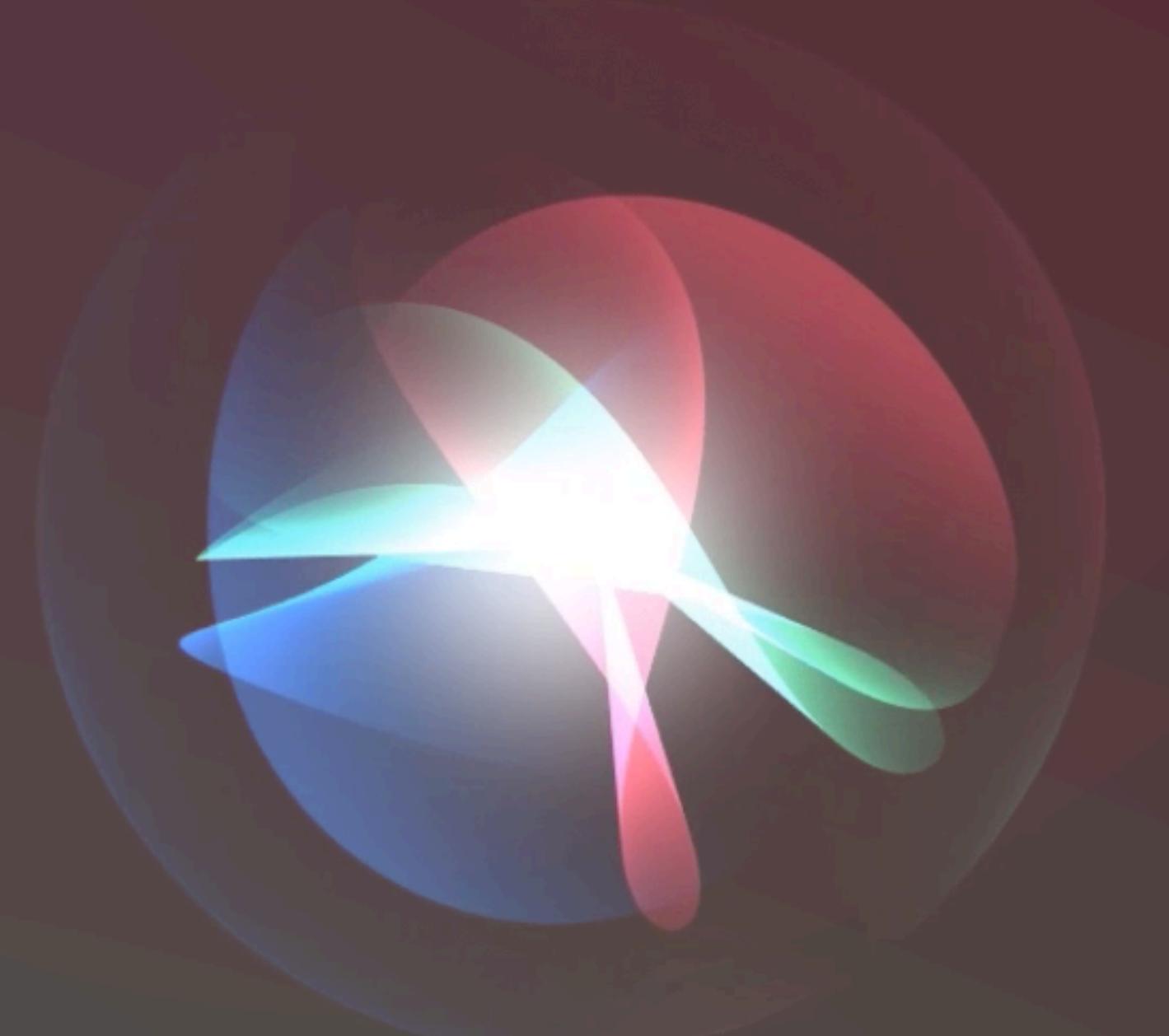


Driving Engagement with Siri Shortcuts and NSUserActivity

Making Siri work for you and your users
Siriをあなたとあなたのユーザーのために機能させる

**Siri does more than ever.
Even before you ask.**



There are some awesome things that Siri can do!

- Automated flows
自動フロー
- Tailored to the user
ユーザーへの最適化
- Ease of engagement
エンゲージのしやすさ

NSUserActivity

- A representation of the state of your app at a moment in time
ある時点でのアプリの状態の表現。
- Used for many purposes, including Siri actions
Siriのアクションを含む多くの目的に使用されます
- Used for “taking the user back”
「ユーザーを連れ戻す」ために使用されます。
- INIntent: Christina Moulton @ try! Swift NYC 2018: “Let Them Say What They Want”

Hey Siri...

As easy as 1, 2, 3

1. Donate an NSUserActivity instance
NSUserActivityインスタンスを受け渡す
2. Assign a command with Siri
Siriのコマンドを割り当てる
3. Handle the action
内部的にアクションを処理する

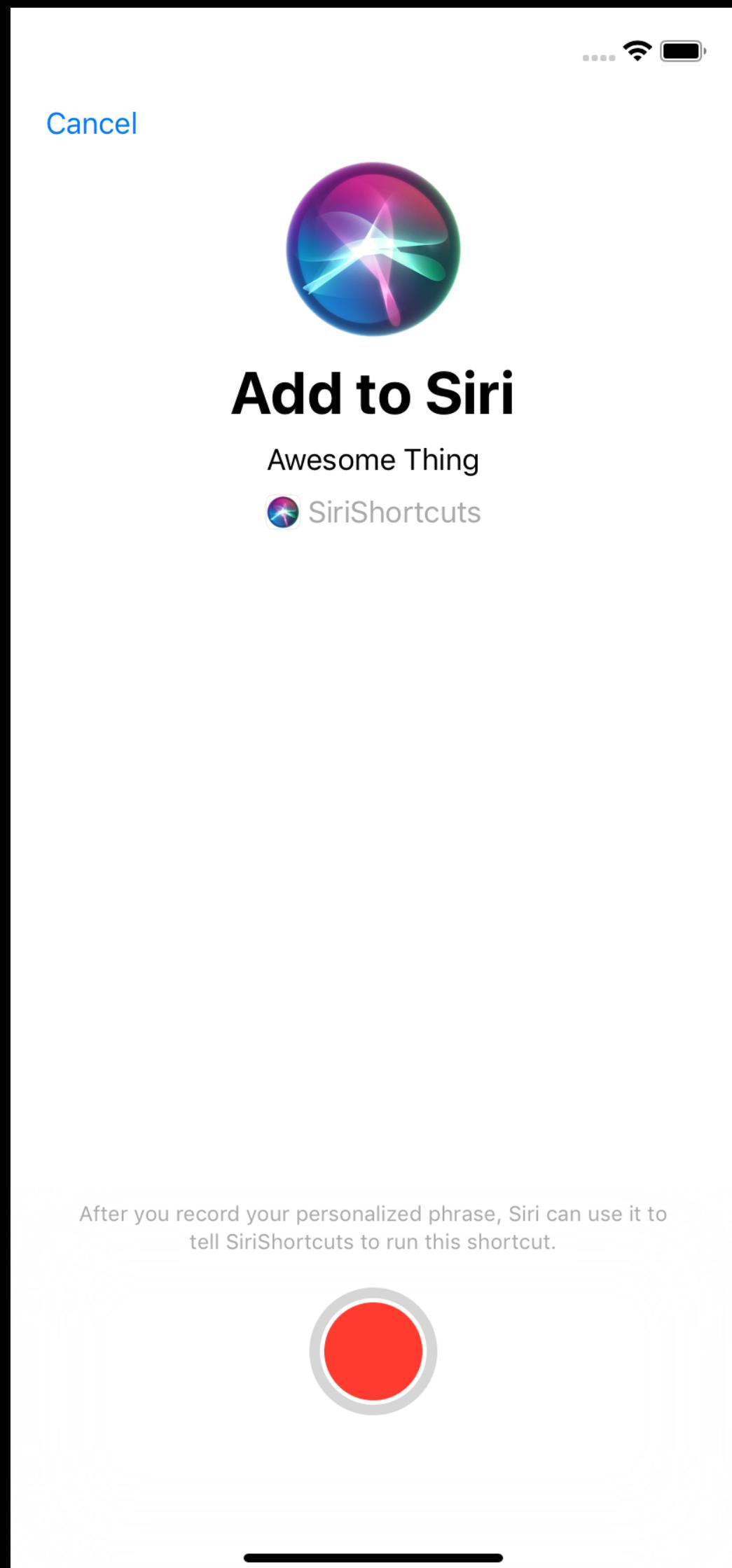
Step 1

Donate an NSUserActivity instance
NSUserActivityインスタンスを受け渡す

Step 1

```
let userActivity = NSUserActivity(activityType: "domain.company.app-name.awesome-thing")
userActivity.title = "Awesome Thing" // Be sure to localize user-facing strings!
userActivity.isEligibleForPrediction = true
self.userActivity = userActivity
```

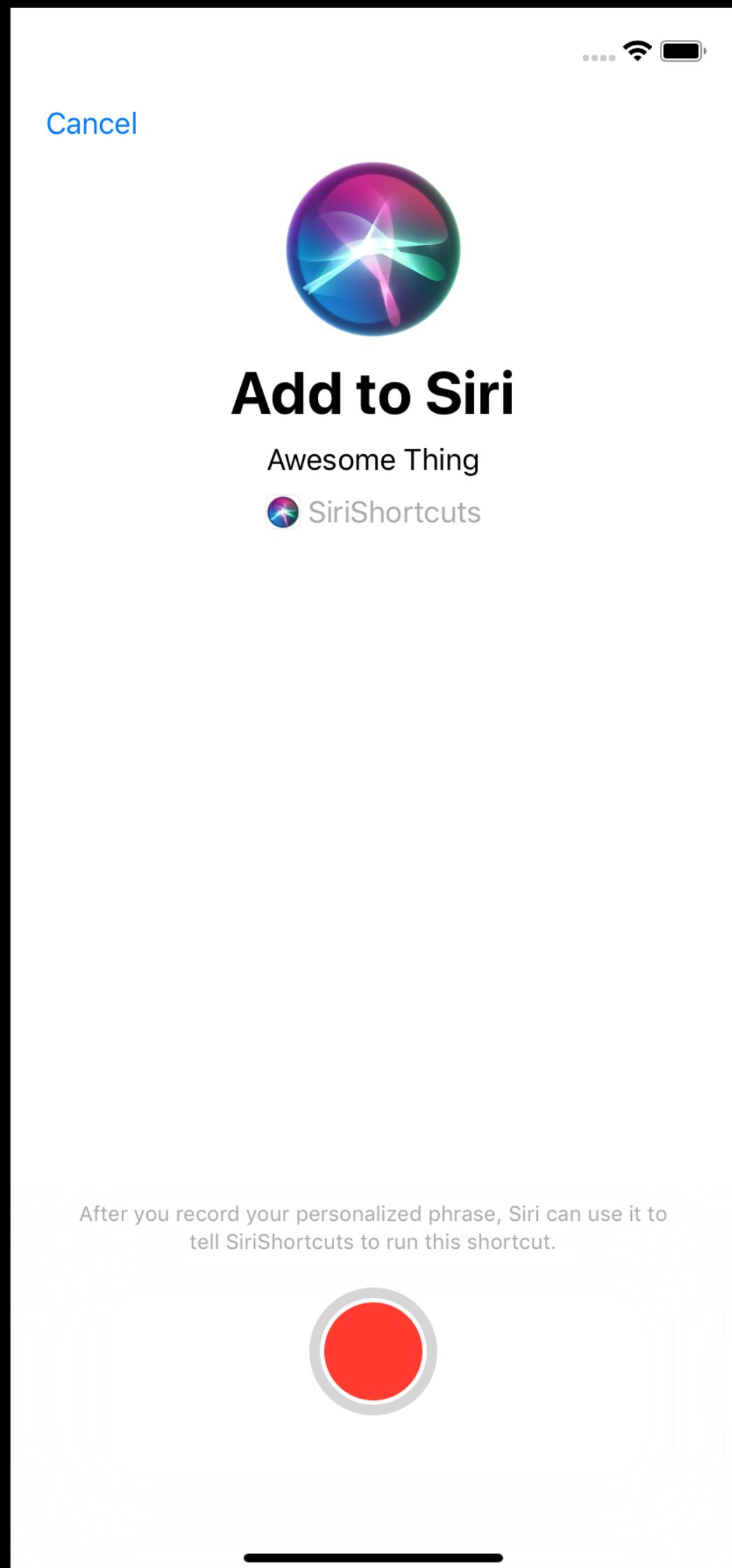


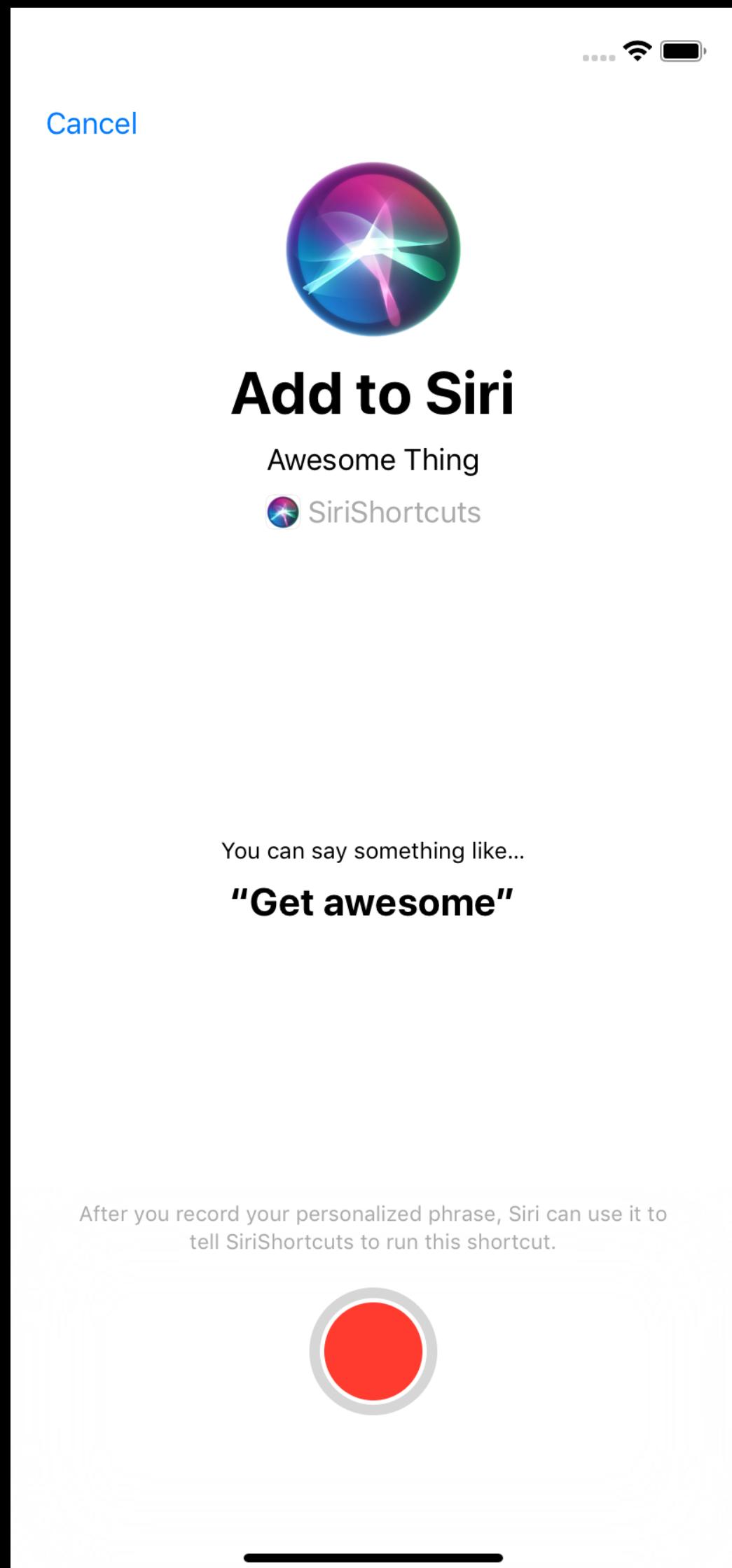


Step 1

NSUserActivity Magic

```
import Intents  
...  
  
userActivity.suggestedInvocationPhrase = "Get awesome"
```

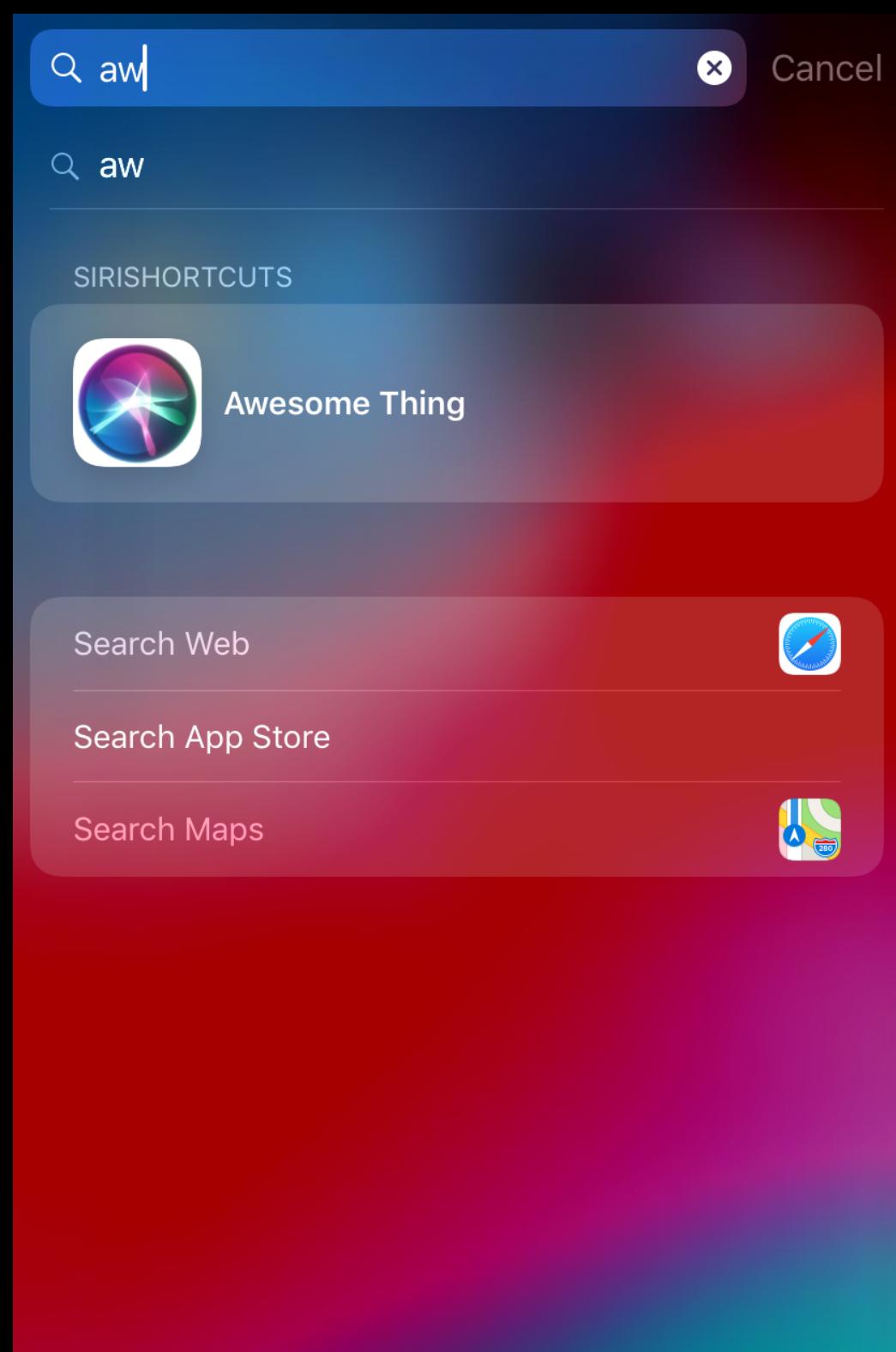




Step 1

NSUserActivity Magic

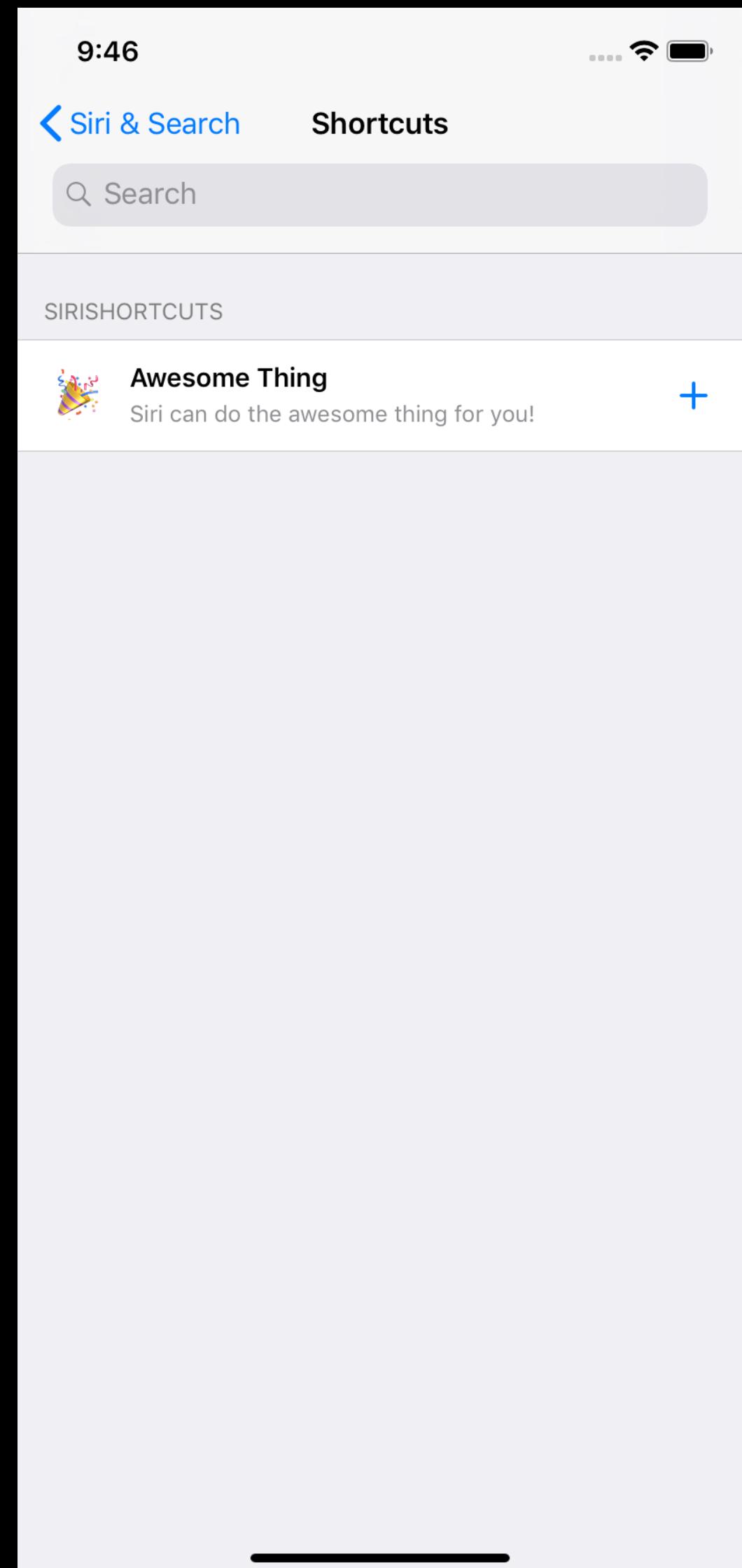
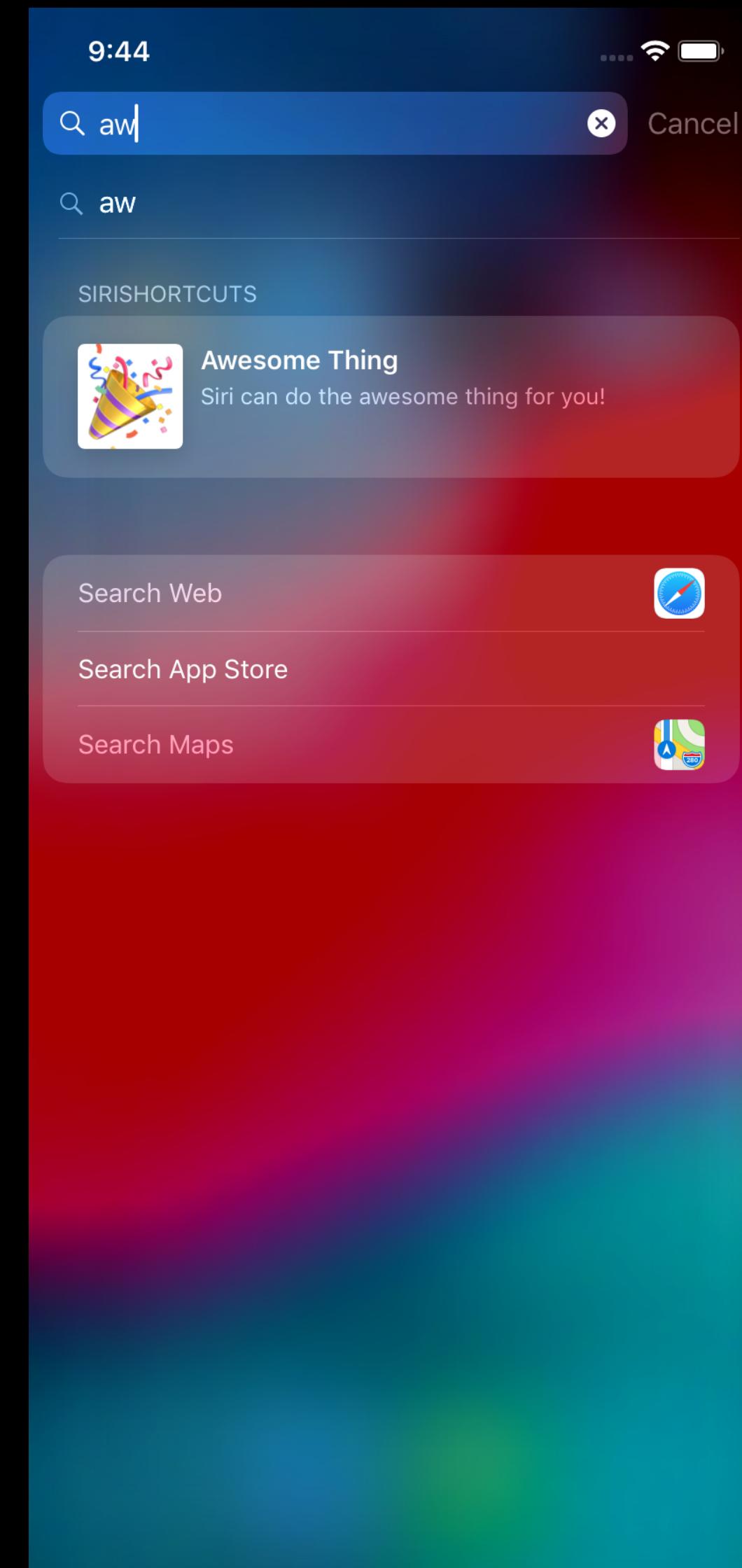
```
userActivity.isEligibleForSearch = true
```

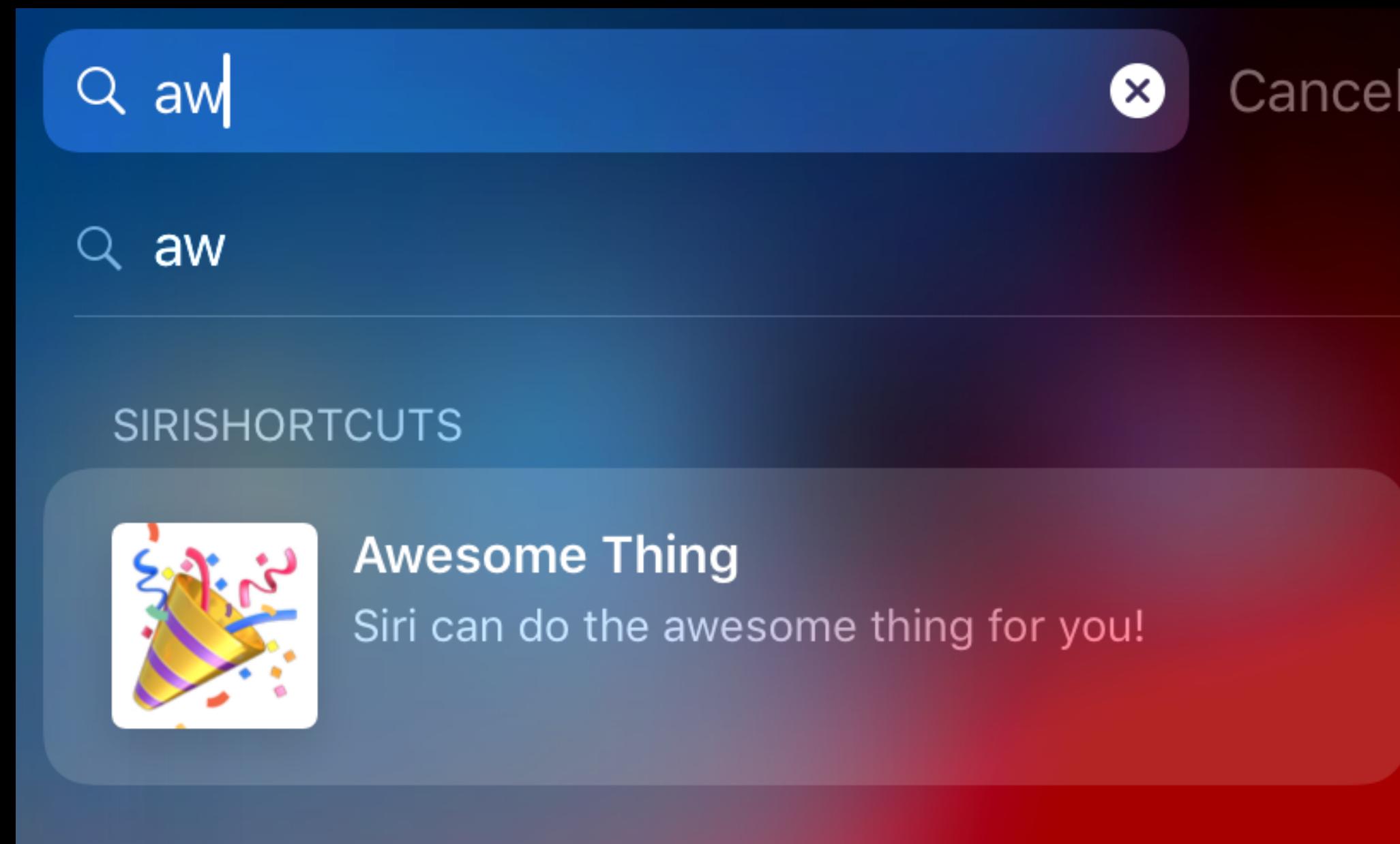


Step 1

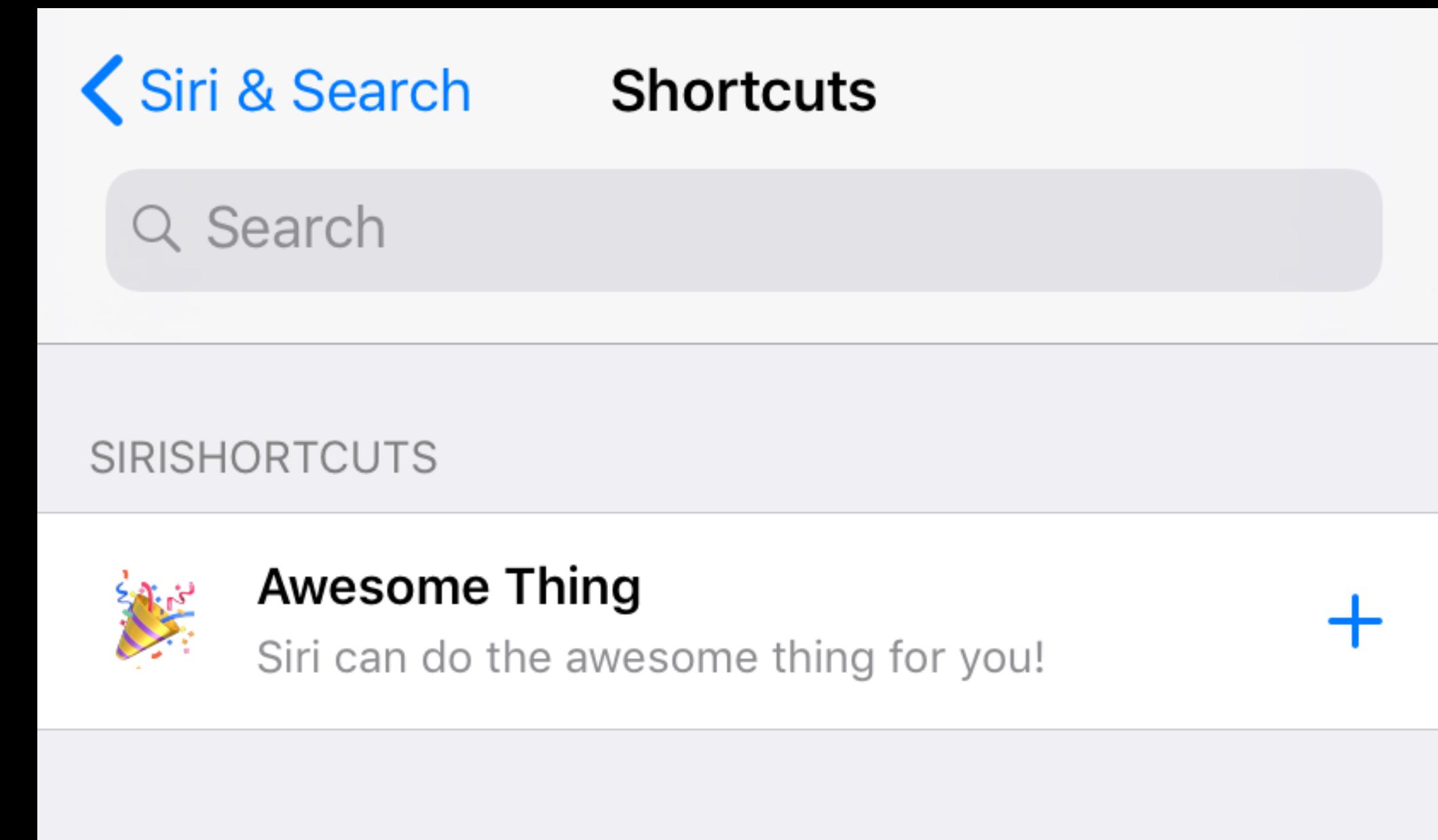
NSUserActivity Magic

```
import CoreSpotlight
Import CoreServices
...
let attributes = CSSearchableItemAttributeSet(itemContentType: kUTTypeItem as String)
attributes.contentDescription = "Shows you how awesome Siri is!"
attributes.thumbnailData = UIImage(named: "awesome-thing")?.pngData()
userActivity.contentAttributeSet = attributes
```





isEligibleForSearch



isEligibleForPrediction

Step 1

Donate an NSUserActivity instance
NSUserActivityインスタンスを受け渡す

Step 2

Assign a command with Siri
Siriのコマンドを割り当てる

Step 2

- `IntentsUI`
- `INUIAddVoiceShortcutButton`
- `INUIAddVoiceShortcutViewController`

Step 2

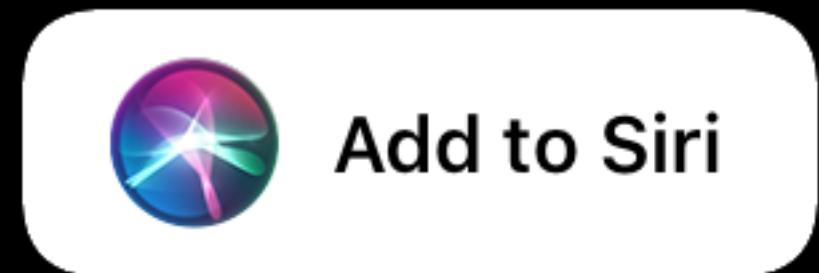
```
let addToSiriButton = INUIAddVoiceShortcutButton(style: .white)
```

Step 2

```
let addToSiriButton = INUIAddVoiceShortcutButton(style: .white)
```

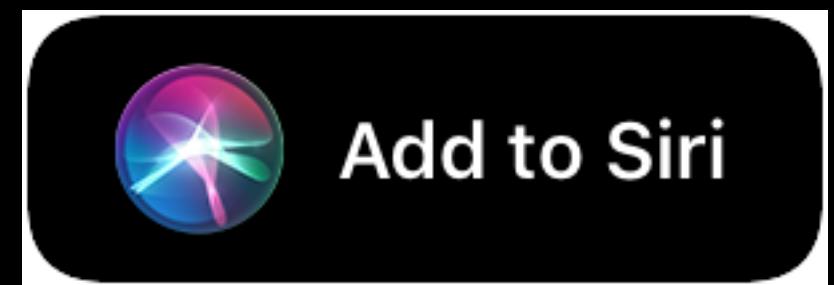
Step 2

```
let addToSiriButton = INUIAddVoiceShortcutButton(style: .white)
```



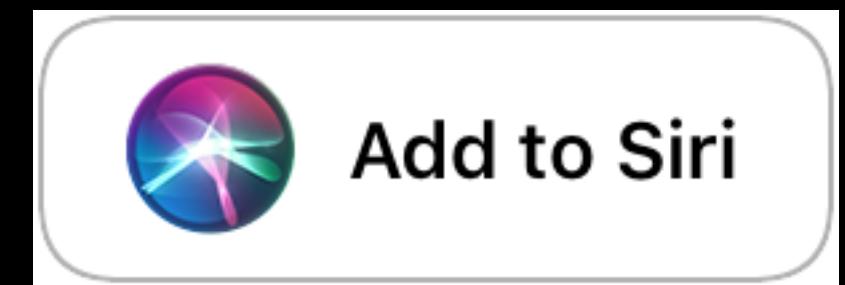
Step 2

```
let addToSiriButton = INUIAddVoiceShortcutButton(style: .black)
```



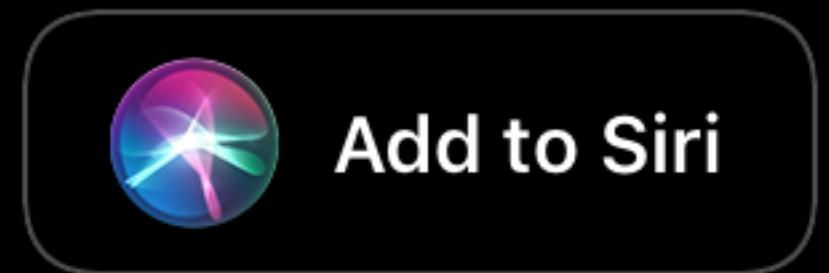
Step 2

```
let addToSiriButton = INUIAddVoiceShortcutButton(style: .whiteOutline)
```



Step 2

```
let addToSiriButton = INUIAddVoiceShortcutButton(style: .blackOutline)
```



Step 2

```
// Set up like any other UIButton  
addToSiriButton.translatesAutoresizingMaskIntoConstraints = false  
addToSiriButton.addTarget(...)  
self.view.addSubview(addToSiriButton)
```

Step 2

```
let viewController = INUIAddVoiceShortcutViewController(shortcut:  
    INShortcut(userActivity: userActivity))  
viewController.delegate = self  
present(viewController...)
```

Step 2

```
let viewController = INUIAddVoiceShortcutViewController(shortcut:  
    INShortcut(userActivity: userActivity))  
viewController.delegate = self  
present(viewController...)
```

Step 2

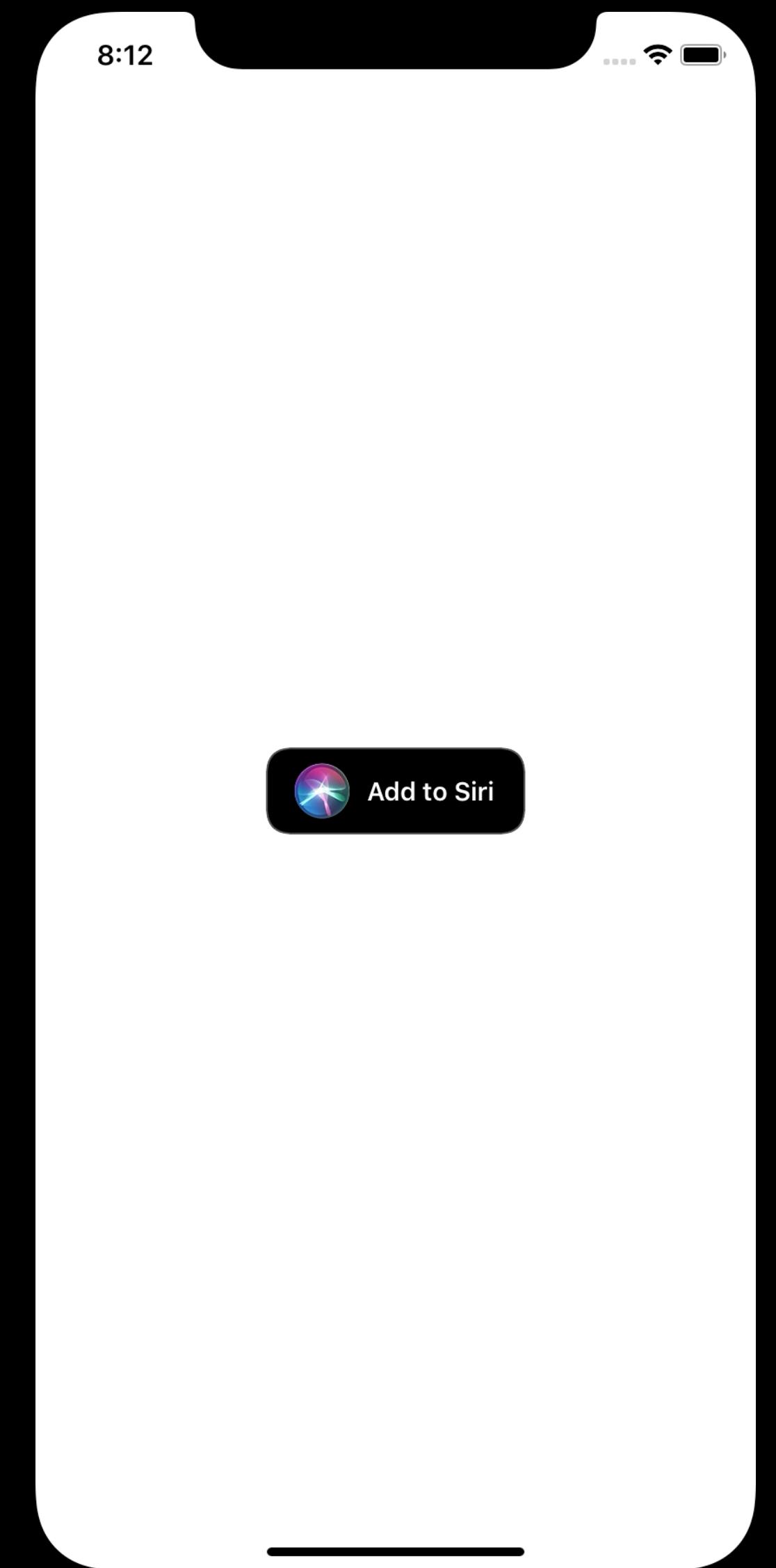
```
let viewController = INUIAddVoiceShortcutViewController(shortcut:  
    INShortcut(userActivity: userActivity))  
viewController.delegate = self  
present(viewController...)
```

Step 2

```
let viewController = INUIAddVoiceShortcutViewController(shortcut:  
    INShortcut(userActivity: userActivity))  
viewController.delegate = self  
present(viewController...)
```

Step 2

```
let viewController = INUIAddVoiceShortcutViewController(shortcut:  
    INShortcut(userActivity: userActivity))  
viewController.delegate = self  
present(viewController...)
```



Step 2

Assign a command with Siri
Siriのコマンドを割り当てる

Step 3

Handle the action

内部的にアクションを処理する

Step 3

1. App needs to be launched

AppDelegate → Store intent → Check in ViewController

2. App running in background

AppDelegate → Send notification → Trigger selector

Step 3

In **AppDelegate.application(_:didFinishLaunchingWithOptions:)**

```
let userActivity = launchOptions?[.userActivityDictionary] as? NSUserActivity  
let intent = userActivity?.activityType  
... // Handle the action here
```

Step 3

In **AppDelegate.application(_:didFinishLaunchingWithOptions:)**

```
let userActivity = launchOptions?[.userActivityDictionary] as? NSUserActivity  
let intent = userActivity?.activityType  
... // Handle the action here
```

Step 3

In **AppDelegate.application(_:didFinishLaunchingWithOptions:)**

```
let userActivity = launchOptions?[.userActivityDictionary] as? NSUserActivity  
let intent = userActivity?.activityType  
... // Handle the action here
```

Step 3

In AppDelegate.application(_: didFinishLaunchingWithOptions:)

```
let userActivity = launchOptions?[.userActivityDictionary] as? NSUserActivity  
let intent = userActivity?.activityType  
... // Handle the action here
```

Then just handle the action!
あとはアクションを処理するだけです!

Step 3

In AppDelegate.application(_:continue:restorationHandler:)

```
let intent = userActivity.activityType  
... // Handle the action here
```

Step 3

In AppDelegate.application(_:continue:restorationHandler:)

```
let intent = userActivity.activityType  
... // Handle the action here
```

Step 3

In AppDelegate.application(_:continue:restorationHandler:)

```
let intent = userActivity.activityType  
... // Handle the action here
```

Then just handle the action!
あとはアクションを処理するだけです!

Step 3

Storing the Action

```
class SiriShortcutManager {  
  
    static let shared = SiriShortcutManager()  
  
    var intent: String? {  
        didSet {  
            guard self.intent != nil else { return }  
            NotificationCenter.default.post(Notification(name: .SiriShortcutReceived))  
        }  
    }  
}
```

Step 3

Storing the Action

```
class SiriShortcutManager {  
  
    static let shared = SiriShortcutManager()  
  
    var intent: String? {  
        didSet {  
            guard self.intent != nil else { return }  
            NotificationCenter.default.post(Notification(name: .SiriShortcutReceived))  
        }  
    }  
}
```

Step 3

Storing the Action

```
class SiriShortcutManager {  
  
    static let shared = SiriShortcutManager()  
  
    var intent: String? {  
        didSet {  
            guard self.intent != nil else { return }  
            NotificationCenter.default.post(Notification(name: .SiriShortcutReceived))  
        }  
    }  
}
```

Step 3

In the AppDelegate

```
AppDelegate.application(_:didFinishLaunchingWithOptions:)...
```

```
let userActivity = launchOptions?[.userActivityDictionary] as? NSUserActivity  
let intent = userActivity?.activityType  
... // Handle the action here
```

```
AppDelegate.application(_:continue:restorationHandler:)...
```

```
let intent = userActivity.activityType  
... // Handle the action here
```

Step 3

In the AppDelegate

```
AppDelegate.application(_:didFinishLaunchingWithOptions:)...
```

```
let userActivity = launchOptions?[.userActivityDictionary] as? NSUserActivity  
let intent = userActivity?.activityType  
SiriShortcutManager.shared.intent = intent
```

```
AppDelegate.application(_:continue:restorationHandler:)...
```

```
let intent = userActivity.activityType  
SiriShortcutManager.shared.intent = intent
```

Step 3

1. App needs to be launched

AppDelegate → Store intent → Check in ViewController

2. App running in background

AppDelegate → Send notification → Trigger selector

Step 3

In the ViewController

```
NotificationCenter.default.addObserver(self,  
    selector: #selector(sayHello),  
    name: .SiriShortcutReceived,  
    object: nil)  
  
self.sayHello()
```

Step 3

In the ViewController

```
NotificationCenter.default.addObserver(self,  
    selector: #selector(sayHello),  
    name: .SiriShortcutReceived,  
    object: nil)  
  
self.sayHello()
```

Step 3

In the ViewController

```
NotificationCenter.default.addObserver(self,  
    selector: #selector(sayHello),  
    name: .SiriShortcutReceived,  
    object: nil)  
  
self.sayHello()
```

Step 3

In the ViewController

```
@objc func sayHello() {  
    guard SiriShortcutManager.shared.intent == self.awesomeThingIdentifier  
        else { return }  
    SiriShortcutManager.shared.intent = nil  
    // Do your thing here!  
    self.present(awesomeThingViewController...)  
}
```

Step 3

In the ViewController

```
@objc func sayHello() {  
    guard SiriShortcutManager.shared.intent == self.awesomeThingIdentifier  
        else { return }  
    SiriShortcutManager.shared.intent = nil  
    // Do your thing here!  
    self.present(awesomeThingViewController...)  
}
```

Step 3

In the ViewController

```
@objc func sayHello() {  
    guard SiriShortcutManager.shared.intent == self.awesomeThingIdentifier  
        else { return }  
    SiriShortcutManager.shared.intent = nil  
    // Do your thing here!  
    self.present(awesomeThingViewController...)  
}
```

Step 3

In the ViewController

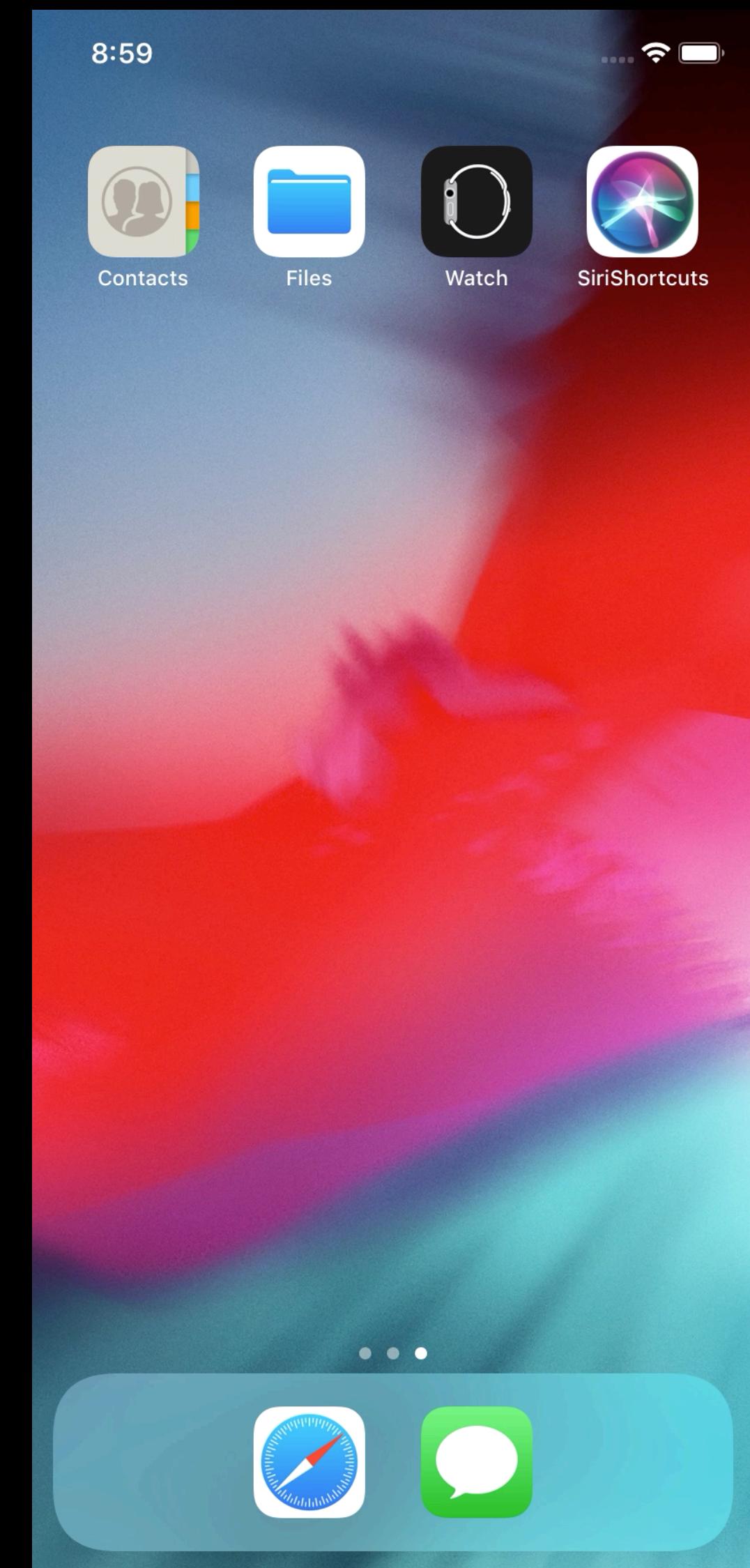
```
@objc func sayHello() {  
    guard SiriShortcutManager.shared.intent == self.awesomeThingIdentifier  
        else { return }  
    SiriShortcutManager.shared.intent = nil  
    // Do your thing here!  
    self.present(awesomeThingViewController...)  
}
```

Step 3

Handle the action

内部的にアクションを処理する

Hey Siri...



A Few Caveats

There are some limitations

- During development, use `NSUserActivity.deleteAllSavedUserActivities`
- 開発中は `NSUserActivity.deleteAllSavedUserActivities`を使う
- Siri does weird caching - Siriは変なキャッシングをする
- No access to all your shortcuts - すべてのショートカットにアクセスできない
- if #available(iOS 12.0, *)

Recap

1. Donate an NSUserActivity instance
NSUserActivityインスタンスを受け渡す
2. Assign a command with Siri
Siriのコマンドを割り当てる
3. Handle the action
内部的にアクションを処理する

Add Siri Shortcuts

Because it's easy!
簡単なので!

Nic Laughter

Software Engineer - ☺ JANE

- Come say hi during Office Hours
- オフィスアワーで会いましょう
- twitter.com/nictheawesome
- <https://github.com/niclaughter/SiriShortcuts>



References

- <https://www.go-gulf.com/blog/virtual-digital-assistants/>
- [https://www.youtube.com/watch?v= OPzPj2UVVw](https://www.youtube.com/watch?v=OPzPj2UVVw)
- <https://developer.apple.com/videos/play/wwdc2018/211/>
- <https://developer.apple.com/videos/play/wwdc2018/214/>
- <https://developer.apple.com/videos/play/wwdc2017/214/>