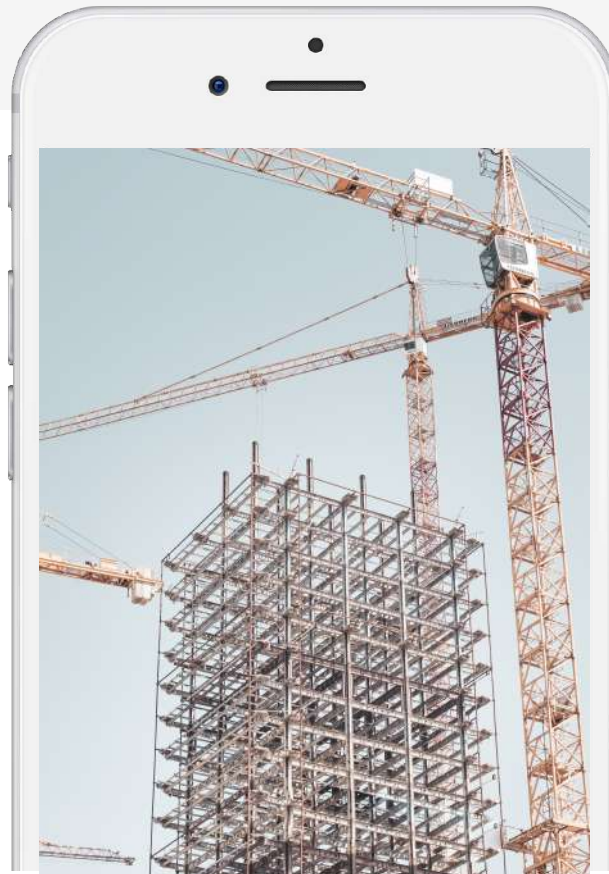


LET THEM SAY WHAT THEY WANT: ADDING SIRI SHORTCUTS

CHRISTINA MOULTON
TEAK MOBILE INC.
TEAKMOBILE.COM
GROKSWIFT.COM

SIRI SHORTCUTS ARE

Access to App
Features
via Voice
Command

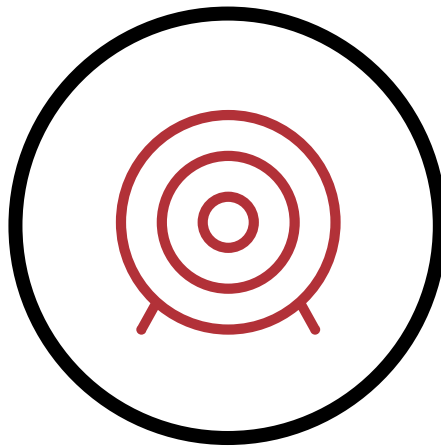


Available on
lock screen,
watch face,
search suggestions,
CarPlay,
Shortcuts app

SIRI SHORTCUTS SHOULD



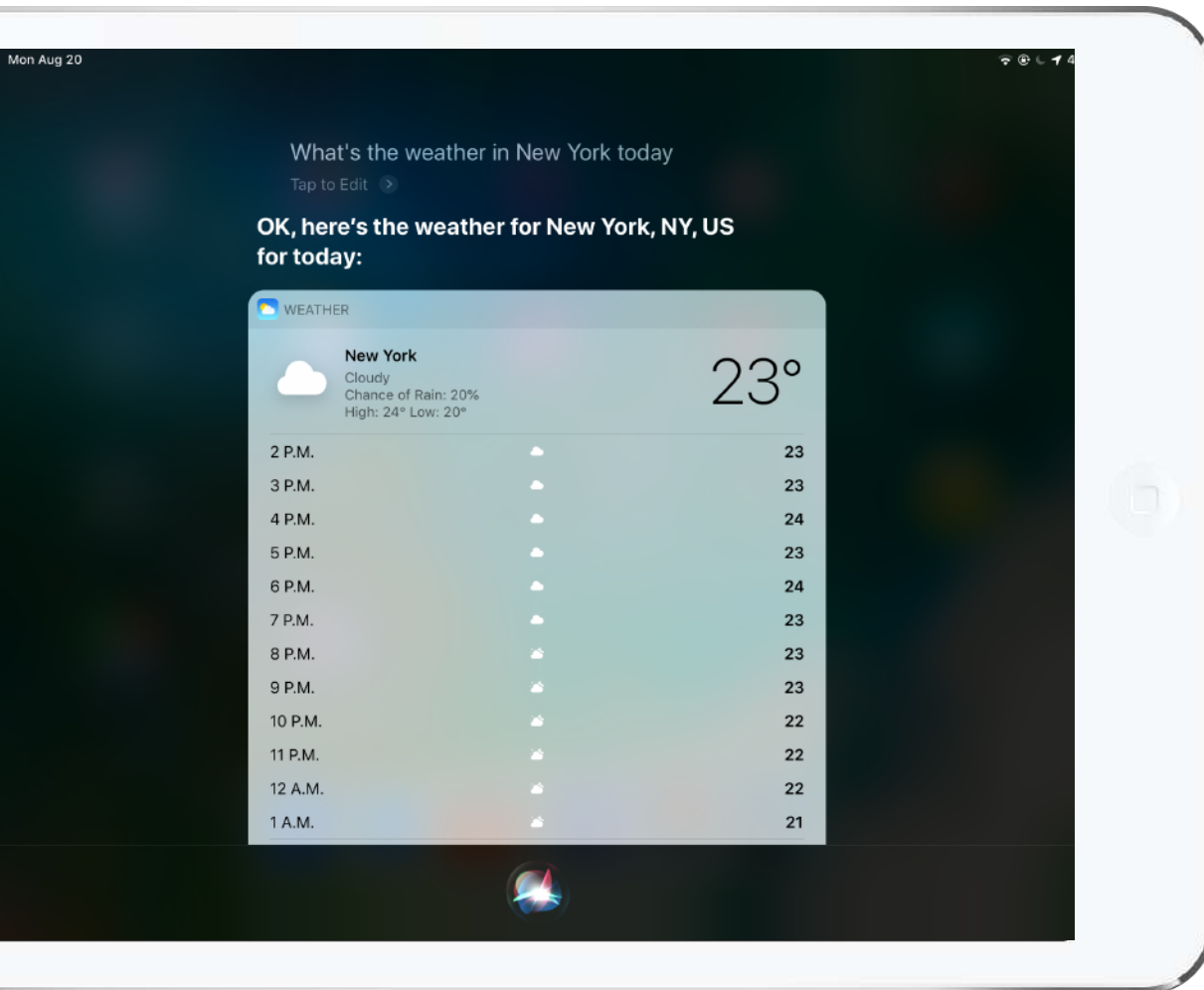
ALWAYS BE
AVAILABLE



ACCELERATE!



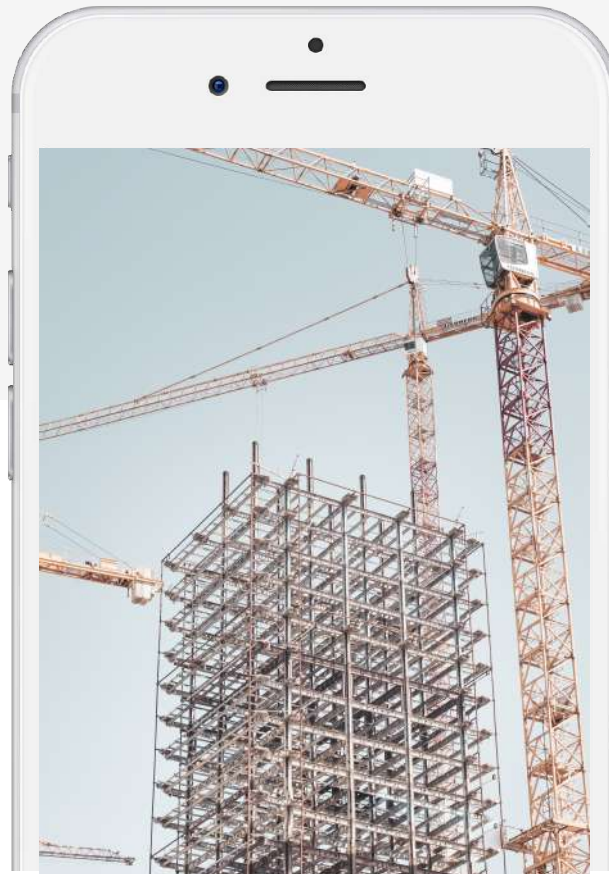
ALWAYS BE
USEFUL



SIRI SHORTCUTS USER EXPERIENCE

- PERFORM TASK IN APP
- ADD SHORTCUT TO SIRI
- ACTIVATE BY VOICE, ETC
- CONFIRMATION
- CUSTOM UI

IMPLEMENTING SIRI SHORTCUTS: `NSUserActivity` & `INIntent`





Shortcuts with NSUserActivity

```
let activity = NSUserActivity(activityType:
    NSUserActivity.myType)
activity.title = "My Title"
activity.contentAttributeSet = ...
```

```
activity.isEligibleForPrediction = true
activity.suggestedInvocationPhrase = "Do it"
```

```
currentVC.userActivity = activity
// OR activity.becomeCurrent()
```

```
func application(_ application: UIApplication,
    continue userActivity: NSUserActivity,
    restorationHandler: @escaping
        ([UIUserActivityRestoring]?) -> Void) -> Bool {
    if userActivity.activityType == NSUserActivity.myType {
        handleUserActivity()
        return true
    }
    return false
}
```




Basic INIntent Implementation

- Add Siri + Intents & Intents UI Extensions
- Intents.intentDefinition:
 - Title, Description, Category
 - Confirmation?
 - Parameters
 - Acceptable Parameter Combos
 - Background Execution?
 - Responses
 - Properties
 - Response Templates
 - Failure, Success, Custom

Basic INIntent Implementation: Extension

```
public class MyCustomIntentHandler: NSObject, MyCustomIntentHandling {
    public func confirm(intent: MyCustomIntent,
        completion: @escaping (MyCustomIntentResponse) -> Void) {
        if isOk(intent) {
            completion(MyCustomIntentResponse(code: .ready, userActivity: nil)):
            completion(MyCustomIntentResponse(code: .failure, userActivity: nil))
        }

        public func handle(intent: MyCustomIntent,
            completion: @escaping (MyCustomIntentResponse) -> Void) {
                guard isOk(intent) else {
                    completion(MyCustomIntentResponse(code: .failure, userActivity: nil))
                    return
                }

                // TODO: handle intent
                completion(MyCustomIntentResponse.success(customProperty: propertyValue))
            }
        }
    }
```


Basic INIntent Implementation: Main App

Handle:

```
func application(_ application: UIApplication,
    continue userActivity: NSUserActivity,
    restorationHandler: @escaping ([UIUserActivityRestoring]?) -> Void) -> Bool {
    if let intent = userActivity.interaction?.intent as? MyCustomIntent {
        handle(intent)
        return true
    }
    return false
}
```

Donate Intent:

```
let interaction = INInteraction(intent: intentForAction(action), response: nil)
interaction.donate { (error) in
    // ...
}
```

Test by setting up voice command, turning on Developer Settings



OPTIONAL ENHANCEMENTS

- `INUIAddVoiceShortcutViewController`
(even for `NSUserActivity`)
- Custom UI
- Show on Watch: `INRelevantShortcut`
- Intent images & titles

RESOURCES

WWDC Videos
2018 – 202, 211,
214, 217
2017 – 214

SiriKit Docs
[developer.apple.com/
sirikit](https://developer.apple.com/sirikit)



Apple Demo App
[developer.apple.com/
documentation/sirikit/
accelerating_app
interactions with shortcuts](https://developer.apple.com/documentation/sirikit/accelerating_app_interactions_with_shortcuts)

Walkthrough
[medium.com/@pietropizzi/a-
beginners-guide-to-developing-
custom-intent-siri-shortcuts-
for-ios-12-a3627b7011af](https://medium.com/@pietropizzi/a-beginners-guide-to-developing-custom-intent-siri-shortcuts-for-ios-12-a3627b7011af)