

#WWDC18

Introduction to Siri Shortcuts

Session 211

Ari Weinstein, Siri
Willem Mattelaer, Siri



Shortcuts let you expose the capabilities of your apps to Siri.

9:41



Search  Cancel

SIRI SUGGESTIONS

Show More



Philz Coffee



Settings



Shortcuts



Calendar



Order Mint Mojito

Order a large, medium cream, medium sugar Mint Mojito at Philz Cupertino.

Philz Coffee





MON 4



10:09



Philz Coffee



Order Your
Usual




464 CAL
15 MIN
11 HR

AT&T



9:41

Monday, June 4

 PHILZ COFFEE | SIRI SUGGESTION

Order Mint Mojito

Order a large, medium cream, medium sugar
Mint Mojito at Philz Cupertino.




AT&T



9:41

Monday, June 4

 PHILZ COFFEE | SIRI SUGGESTION

Order Mint Mojito

Order a large, medium cream, medium sugar
Mint Mojito at Philz Cupertino.



9:41



Coffee time

Tap to Edit >

Ready to order with Philz Coffee?

 PHILZ COFFEE

Pick-up at Cupertino - 10:00am

**Is it Mint Mojito
Time?**

Large, Medium Cream, Medium Sugar



No

Yes



9:41



 PHILZ COFFEE



Thanks Kim!

See you at Cupertino in 11 mins

Mint Mojito **\$5.50**
Large, Medium Cream, Medium Sugar



OK. Ordering. Philz Coffee says:
"Great! Kim, we'll see you at Philz
Cupertino in 11 minutes for pick-
up."



9:41



Cancel



Add to Siri

Order Mint Mojito

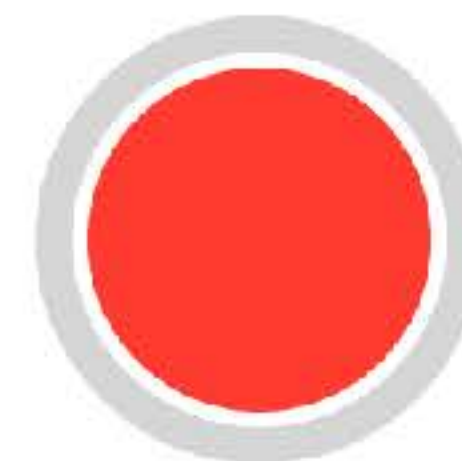
Order a large, medium cream, medium sugar Mint Mojito at Philz Cupertino.



You can say something like...

"Coffee time"

After you record your personalized phrase, Siri can use it to tell "Philz Coffee" to run this shortcut.





9:41



Start My Day

Done



Philz Coffee



Order Mint Mojito

Order a large, medium cream, medium sugar Mint Mojito at Philz Cupertino.

Show When Run



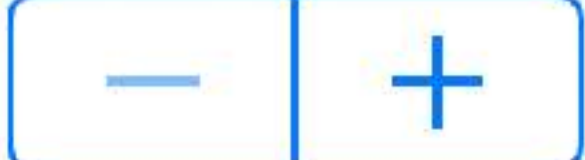
Get Upcoming Events



Calendar

All Calendars

Get 1 Event



Day

Any Day



Show Directions



Mode

Driving

Walking

Transit

Search

Calendar

Adopting shortcuts

Optimizing for suggestions

Privacy

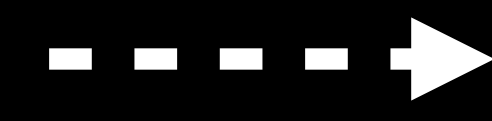
Media shortcuts

Adopting Shortcuts

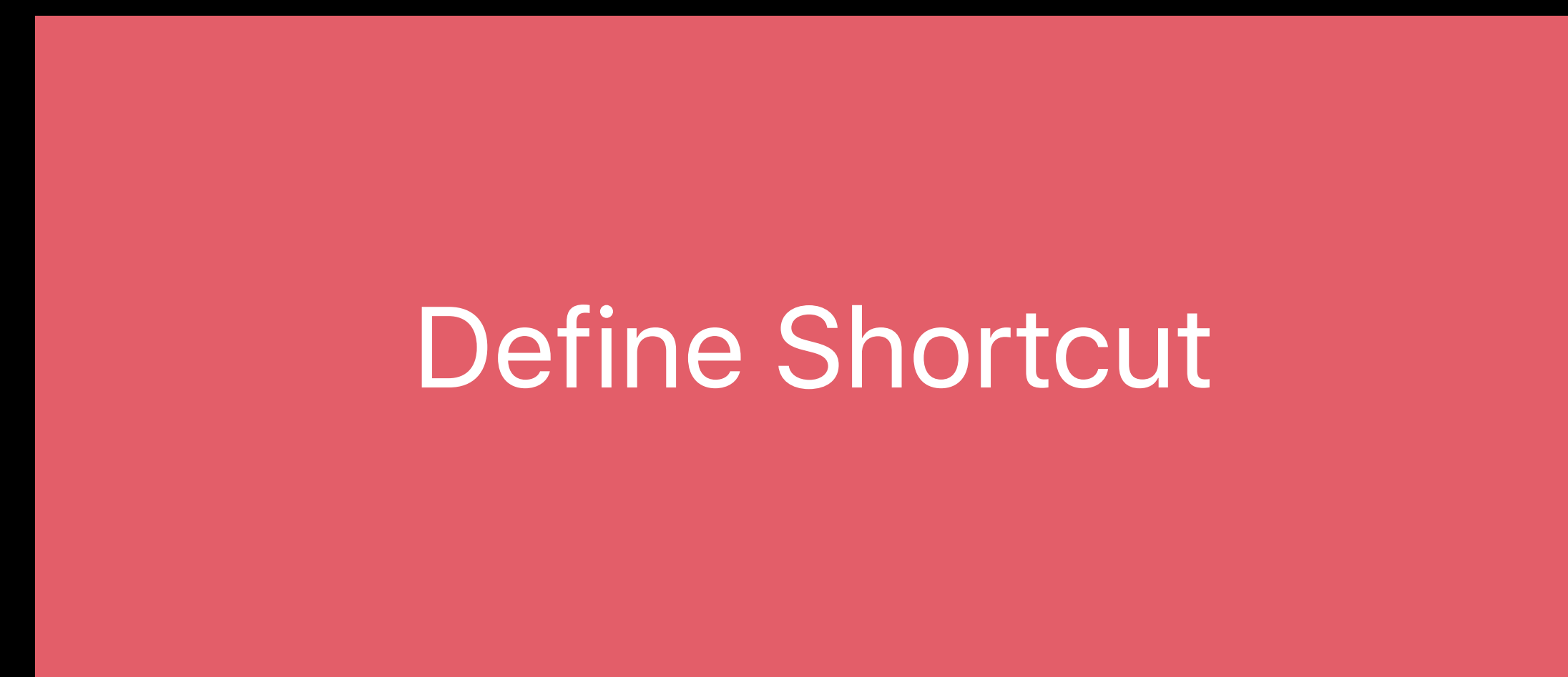
Creating Shortcuts

Creating Shortcuts

Define Shortcut



Creating Shortcuts



Creating Shortcuts



What Makes a Great Shortcut?

Every shortcut should:

- Accelerate the user to perform a key function of your app
- Be of persistent interest to the user
- Be executable at any time

Shortcuts APIs



NSUserActivity



Intents

Choosing an Adoption Strategy



Use `NSUserActivity` if your shortcut:

- Opens something in your app
- Represents showing items that you index in Spotlight or offer for Handoff

Choosing an Adoption Strategy



For the best experience, adopt Intents. Intents-based shortcuts can:

- Run inline, without launching your app
- Include custom voice response or a custom UI
- Include granular predictions



NSUserActivity

NSUserActivity

Define your shortcut



Declare a type in `NSUserActivityTypes` in your `Info.plist`

```
<key>NSUserActivityTypes</key>
<array>
  <string>com.myapp.name.my-activity-type</string>
</array>
```




```
// How to donate a shortcut
```

```
let userActivity = NSUserActivity(activityType: "com.myapp.name.my-activity-type")
userActivity.isEligibleForSearch = true
userActivity.isEligibleForPrediction = true
userActivity.title = "Activity"
userActivity.userInfo = ["key": "value"]
userActivity.suggestedInvocationPhrase = "Let's do it"
```

```
let attributes = CSSearchableItemAttributeSet(itemContentType: kUTTypeItem as String)
let image = UIImage(named: "myImage")!
attributes.thumbnailData = image.pngData()
attributes.contentDescription = "Subtitle"
userActivity.contentAttributeSet = attributes
```

```
viewController.userActivity = userActivity
```



```
// How to donate a shortcut
```

```
let userActivity = NSUserActivity(activityType: "com.myapp.name.my-activity-type")
```

```
userActivity.isEligibleForSearch = true
```

```
userActivity.isEligibleForPrediction = true
```

```
userActivity.title = "Activity"
```

```
userActivity.userInfo = ["key": "value"]
```

```
userActivity.suggestedInvocationPhrase = "Let's do it"
```

```
let attributes = CSSearchableItemAttributeSet(itemContentType: kUTTypeItem as String)
```

```
let image = UIImage(named: "myImage")!
```

```
attributes.thumbnailData = image.pngData()
```

```
attributes.contentDescription = "Subtitle"
```

```
userActivity.contentAttributeSet = attributes
```

```
viewController.userActivity = userActivity
```



```
// How to donate a shortcut
```

```
let userActivity = NSUserActivity(activityType: "com.myapp.name.my-activity-type")
```

```
userActivity.isEligibleForSearch = true
```

```
userActivity.isEligibleForPrediction = true
```

```
userActivity.title = "Activity"
```

```
userActivity.userInfo = ["key": "value"]
```

```
userActivity.suggestedInvocationPhrase = "Let's do it"
```

```
let attributes = CSSearchableItemAttributeSet(itemContentType: kUTTypeItem as String)
```

```
let image = UIImage(named: "myImage")!
```

```
attributes.thumbnailData = image.pngData()
```

```
attributes.contentDescription = "Subtitle"
```

```
userActivity.contentAttributeSet = attributes
```

```
viewController.userActivity = userActivity
```



```
// How to donate a shortcut
```

```
let userActivity = NSUserActivity(activityType: "com.myapp.name.my-activity-type")
userActivity.isEligibleForSearch = true
userActivity.isEligibleForPrediction = true
userActivity.title = "Activity"
userActivity.userInfo = ["key": "value"]
userActivity.suggestedInvocationPhrase = "Let's do it"
```

```
let attributes = CSSearchableItemAttributeSet(itemContentType: kUTTypeItem as String)
let image = UIImage(named: "myImage")!
attributes.thumbnailData = image.pngData()
attributes.contentDescription = "Subtitle"
userActivity.contentAttributeSet = attributes

viewController.userActivity = userActivity
```



```
// How to donate a shortcut
```

```
let userActivity = NSUserActivity(activityType: "com.myapp.name.my-activity-type")
userActivity.isEligibleForSearch = true
userActivity.isEligibleForPrediction = true
userActivity.title = "Activity"
userActivity.userInfo = ["key": "value"]
userActivity.suggestedInvocationPhrase = "Let's do it"
```

```
let attributes = CSSearchableItemAttributeSet(itemContentType: kUTTypeItem as String)
let image = UIImage(named: "myImage")!
attributes.thumbnailData = image.pngData()
attributes.contentDescription = "Subtitle"
userActivity.contentAttributeSet = attributes
```

```
viewController.userActivity = userActivity
```



```
// How to handle a shortcut
```

```
func application(_ application: UIApplication, continue userActivity: NSUserActivity,  
restorationHandler: @escaping ([Any]?) -> Void) -> Bool {  
    if userActivity.activityType == "com.myapp.name.my-activity-type" {  
        // Restore state for userActivity and userInfo  
    }  
}
```

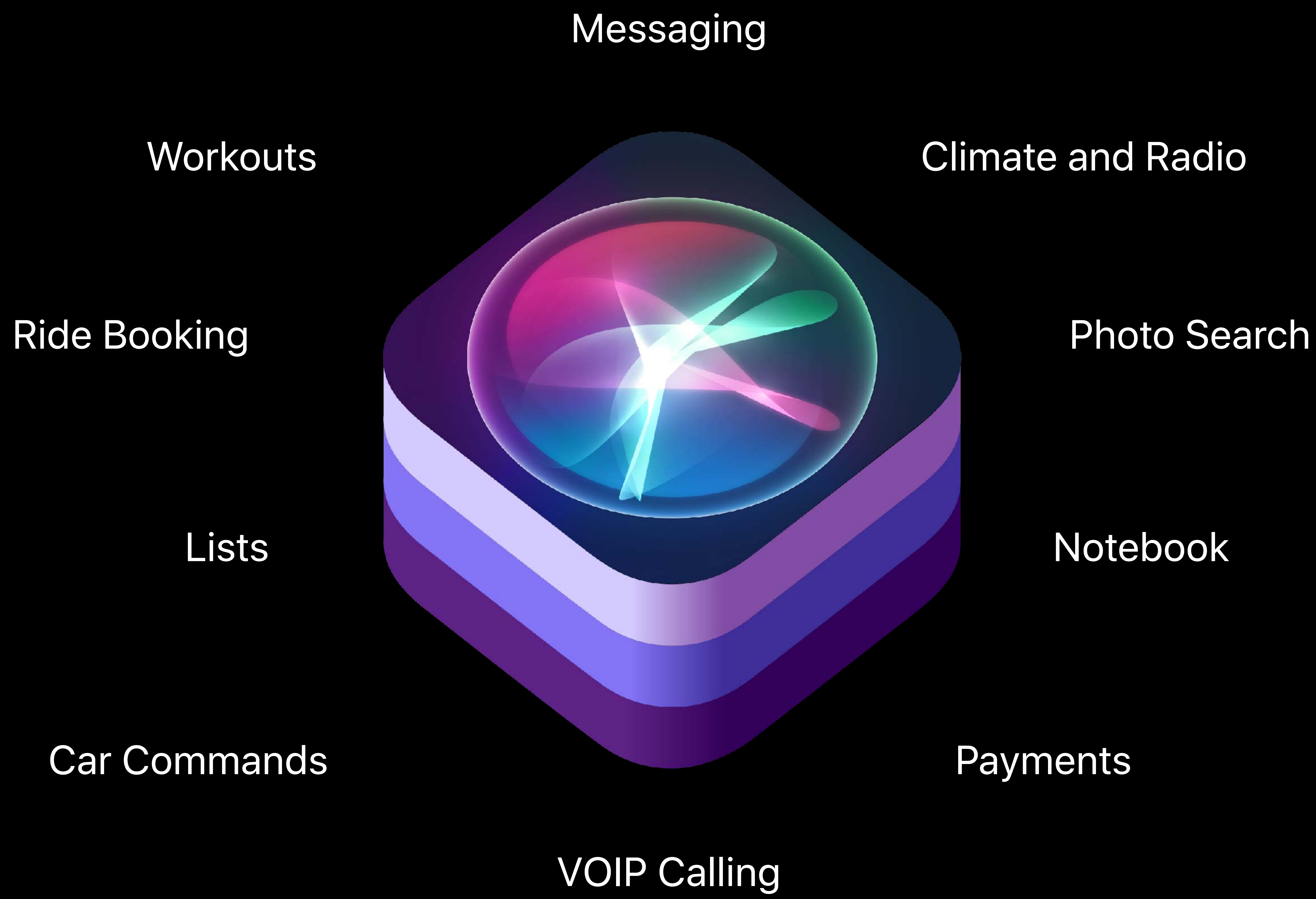


NSUserActivity



Intents





Custom



Built-In



Intents

Define your shortcut



Create an intent definition file in Xcode to customize a built-in SiriKit intent or define a new custom intent.

Go to File -> New File, and choose "SiriKit Intent Definition File."

No Intents

No Selection

No Intents

No Selection



Filter

CUSTOM INTENTS

I OrderSoup

R Response

▼ Custom Intent

Category

Title

Description

Confirmation User confirmation required

▼ Parameters

Parameter	Type	Array

+ -

▼ Shortcut Types

Parameter Combination	Title	Subtitle	Background
(No Parameters)	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/> Supports background execution

+ -

Intent

Name

Authentication

Custom Class

Class Name

Watch Shortcuts

Primary Type

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

Parameter	Type	Array
items	Custom	<input checked="" type="checkbox"/>
deliveryLocation	Location	<input type="checkbox"/>

+ -

Shortcut Types

Parameter Combination
items
items, deliveryLocation
deliveryLocation

Title Order items

Subtitle To deliveryLocation

Background Supports background execution

+ -

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

Parameter

items

deliveryLocation

+ -

Shortcut Types

Parameter Combination

items

items, deliveryLocation

deliveryLocation

+ -

Title Order items

Subtitle To deliveryLocation

Background Supports background execution

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

Parameter

items

deliveryLocation

+ -

Shortcut Types

Parameter Combination

items

items, deliveryLocation

deliveryLocation

+ -

Title Order items

Subtitle To deliveryLocation

Background Supports background execution

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

Parameter	Type	Array
items	Custom	<input checked="" type="checkbox"/>
deliveryLocation	Location	<input type="checkbox"/>

+ -

Shortcut Types

Parameter Combination
items
items, deliveryLocation
deliveryLocation

Title Order items

Subtitle To deliveryLocation

Background Supports background execution

+ -

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

items

deliveryLocation

+

▼ **Parameters**

Parameter	Type	Array
items	Custom	<input checked="" type="checkbox"/>
deliveryLocation	Location	<input type="checkbox"/>

+

+

Subtitle To deliveryLocation

Background Supports background execution

+

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

Parameter	Type	Array
items	Custom	<input checked="" type="checkbox"/>
deliveryLocation	Location	<input type="checkbox"/>

+ -

Shortcut Types

Parameter Combination
items
items, deliveryLocation
deliveryLocation

Title Order items

Subtitle To deliveryLocation

Background Supports background execution

+ -

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

Parameter	Type	Array
items	Custom	<input checked="" type="checkbox"/>

Shortcut Types

- Parameter Combination
- items
- items, deliveryLocation**
- deliveryLocation

Title: Order items

Subtitle: To deliveryLocation

Background Supports background execution

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation

CUSTOM INTENTS

OrderSoup

Response

Custom Intent

Category Order

Title Order Soup

Description Order a soup to a specific location

Confirmation User confirmation required

Parameters

Parameter	Type	Array
items	Custom	<input checked="" type="checkbox"/>

Shortcut Types

- Parameter Combination
- items
- items, deliveryLocation**
- deliveryLocation

Title

Subtitle

Background Supports background execution

Intent

Name OrderSoup

Authentication Restricted While Locked

Custom Class

Class Name OrderSoupIntent

Watch Shortcuts

Primary Type items, deliveryLocation



```
public class OrderSoupIntent: INIntent {  
    public var items: [INObject]?  
    public var deliveryLocation: CLPlacemark?  
}
```

```
public protocol OrderSoupIntentHandling: NSObjectProtocol {  
    public func handle(intent: OrderSoupIntent, completion: @escaping  
(OrderSoupIntentResponse) -> Void)  
    optional func confirm(intent: OrderSoupIntent, completion: @escaping  
(OrderSoupIntentResponse) -> Void)  
}
```





Intents

Define your shortcut





Target selection

- Your intent definition should be added to any target where the intents are used
- If you're using a framework, only generate classes for the framework

Target Membership		
<input checked="" type="checkbox"/>	 SoupChef	No Generated Classes ⇅
<input checked="" type="checkbox"/>	 SoupChefIntents	No Generated Classes ⇅
<input checked="" type="checkbox"/>	 SoupKit	Intent Classes ⇅

Framework

Target Membership		
<input checked="" type="checkbox"/>	 SoupChef	Intent Classes ⇅
<input checked="" type="checkbox"/>	 SoupChefIntents	Intent Classes ⇅

No Framework



```
// Donate your shortcut

let intent = PlaceOrderIntent()
intent.items = order.items.map({ INObject(identifier: $0.id.uuidString, display:
$0.menuItem.displayName) })
intent.deliveryLocation = order.destinationLocation

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



```
// Handle your shortcut

func application(_ application: UIApplication, continue userActivity: NSUserActivity,
restorationHandler: @escaping ([Any]?) -> Void) -> Bool {
    if userActivity.activityType == "OrderSoupIntent",
        let intent = userActivity.interaction?.intent as? OrderSoupIntent {
        // Show ordering UI, pre-populated with the fields from the intent
    }
}
```



Choose a template for your new target:

iOS watchOS tvOS macOS Cross-platform

Filter

Application Extension



Action Extension



Audio Unit Extension



Broadcast Setup UI Extension



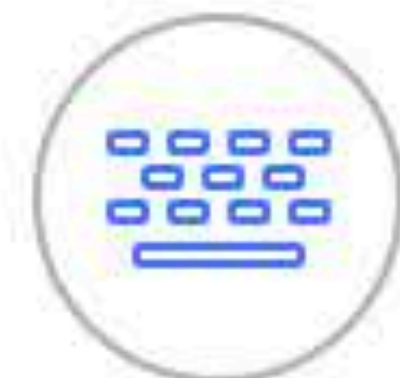
Broadcast Upload Extension



Call Directory Extension



Content Blocker Extension



Custom Keyboard Extension



File Provider Extension



File Provider UI Extension



iMessage Extension



Intents Extension



Intents UI Extension



Message Filter Extension



Network Extension



Notification Content Extension



Cancel

Previous

Next



```
// Handle your shortcut
```

```
class IntentHandler: INExtension, OrderSoupIntentHandling {  
  
    func confirm(intent: OrderSoupIntent, completion: (OrderSoupIntentResponse) -> Void) {  
        completion(OrderSoupIntentResponse(code: .ready, userActivity: nil))  
    }  
  
    func handle(intent: OrderSoupIntent, completion: (OrderSoupIntentResponse) -> Void) {  
        // Order the soup  
        completion(OrderSoupIntentResponse(code: .success, userActivity: nil))  
    }  
}
```

Intent Handling Notes



Implement an intents extension to run your shortcut in the background, without launching your app

Always implement `-continueUserActivity`, even if you have an intents extension

INRelevantShortcut

Expose Shortcuts to the Siri Watch Face by providing INRelevantShortcut objects

Optionally, include relevance information as a hint as to when your shortcuts should be shown on the Watch Face

Works even if you don't have a Watch app

Demo

Adopting Shortcuts

Willem Mattelaer, Siri

Soup Chef

Made `NSUserActivity` eligible for prediction

Soup Chef

Made `NSUserActivity` eligible for prediction

Defined a custom intent

Soup Chef

Made `NSUserActivity` eligible for prediction

Defined a custom intent

Donated the intent

Soup Chef

Made `NSUserActivity` eligible for prediction

Defined a custom intent

Donated the intent

Handled the intent

Optimizing for Suggestions

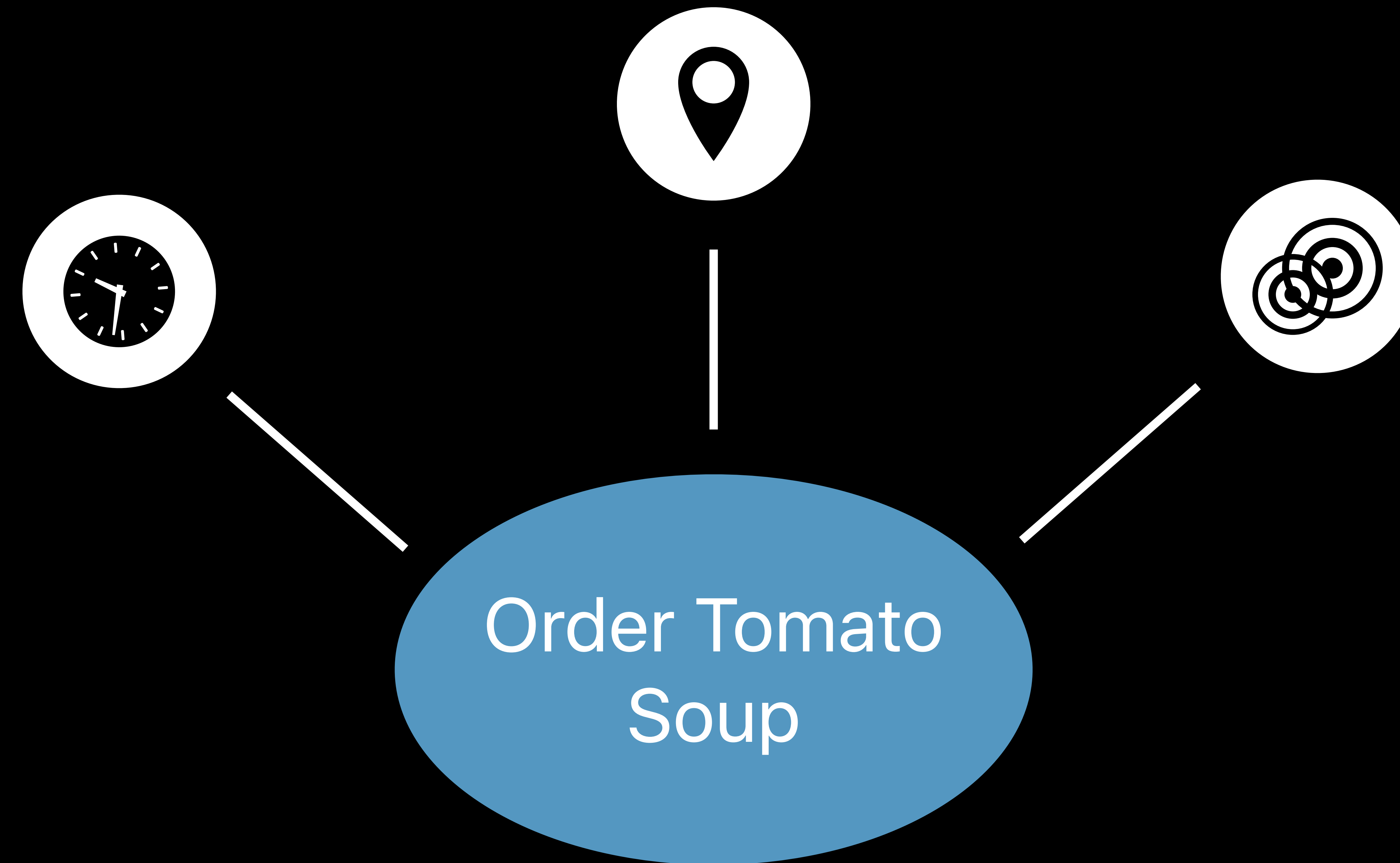
How is a Shortcut Suggested?

How is a Shortcut Suggested?



Order Tomato
Soup

How is a Shortcut Suggested?



How is a Shortcut Suggested?

Time



How is a Shortcut Suggested?

Order tomato soup
with croutons

Monday, 11:57 AM

Time



How is a Shortcut Suggested?

Order tomato soup
with croutons

Order tomato soup
with red pepper

Monday, 11:57 AM

Monday, 6:06 PM

Time



How is a Shortcut Suggested?

Order tomato soup
with croutons

Order tomato soup
with red pepper

Order tomato soup
with croutons

Monday, 11:57 AM

Monday, 6:06 PM

Tuesday, 11:59 AM

Time



How is a Shortcut Suggested?

Order tomato soup
with croutons

Order tomato soup
with red pepper

Order tomato soup
with croutons

...

Shortcut Suggestion

Monday, 11:57 AM

Monday, 6:06 PM

Tuesday, 11:59 AM

Friday, 12:00 PM

Time



How is a Shortcut Suggested?

Order tomato soup
with croutons

Order tomato soup
with red pepper

Order tomato soup
with croutons

...

Order tomato soup
with croutons

Shortcut Suggestion

Monday, 11:57 AM

Monday, 6:06 PM

Tuesday, 11:59 AM

Friday, 12:00 PM

Time



How is a Shortcut Suggested?

NSUserActivity

NSUserActivity.userInfo

soup

quantity

scrollPosition

How is a Shortcut Suggested?

NSUserActivity

Time



How is a Shortcut Suggested?

NSUserActivity

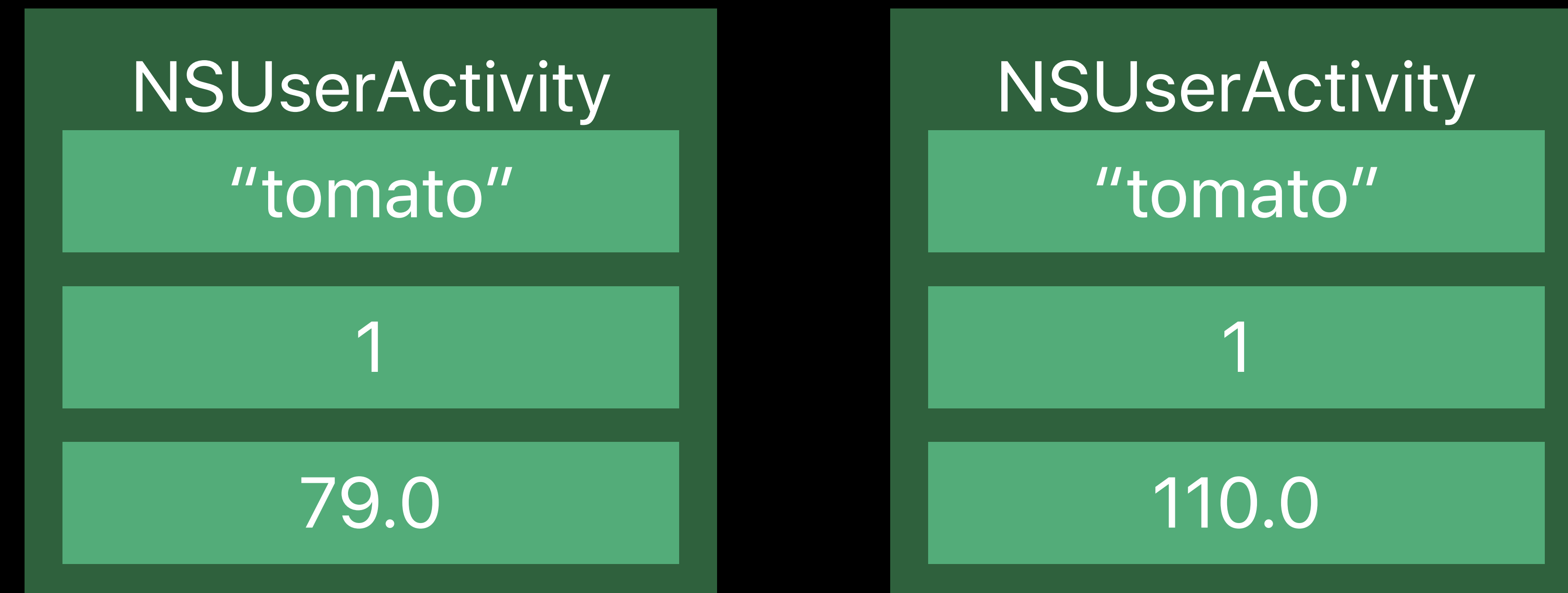
NSUserActivity
"tomato"
1
79.0

Time



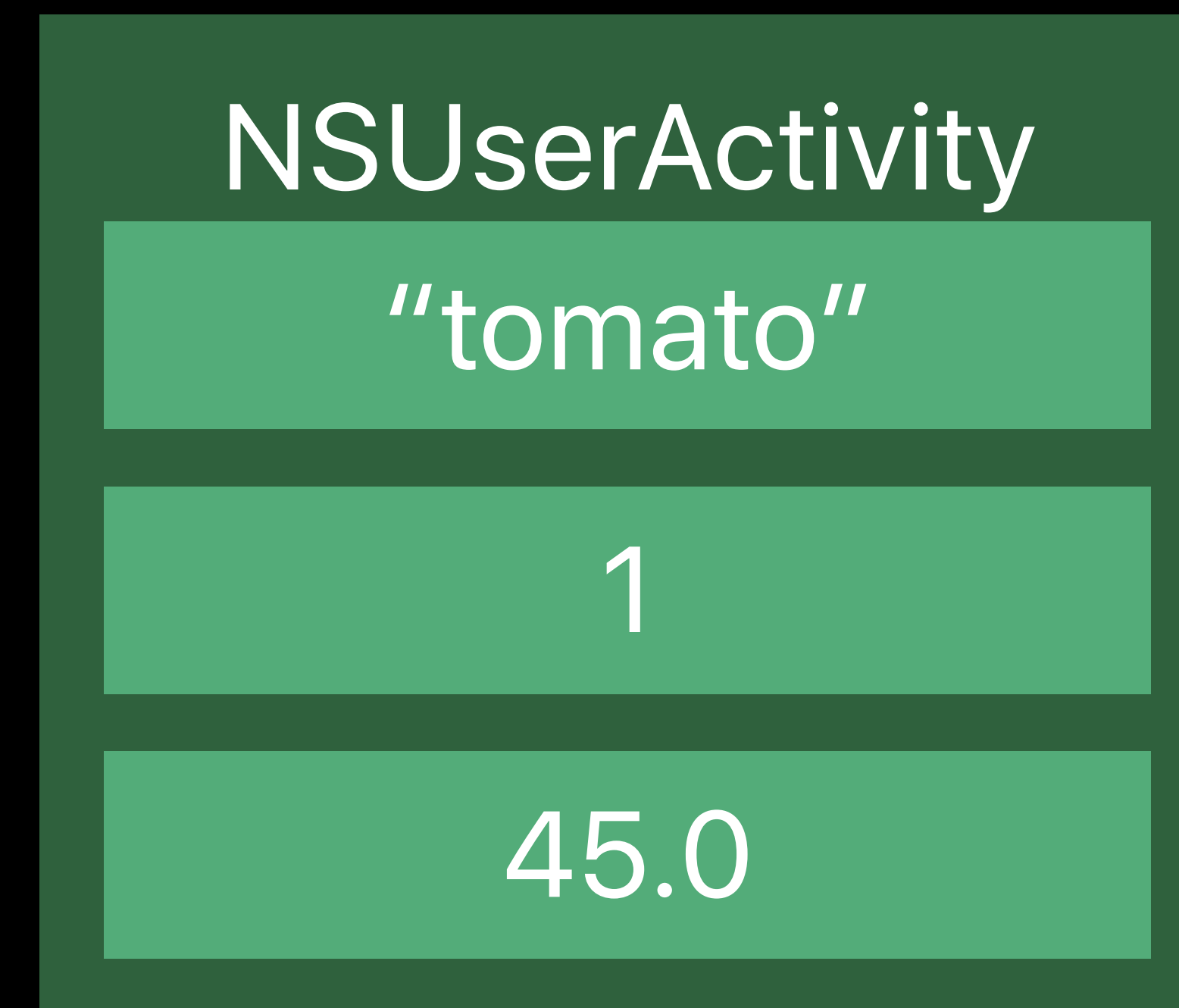
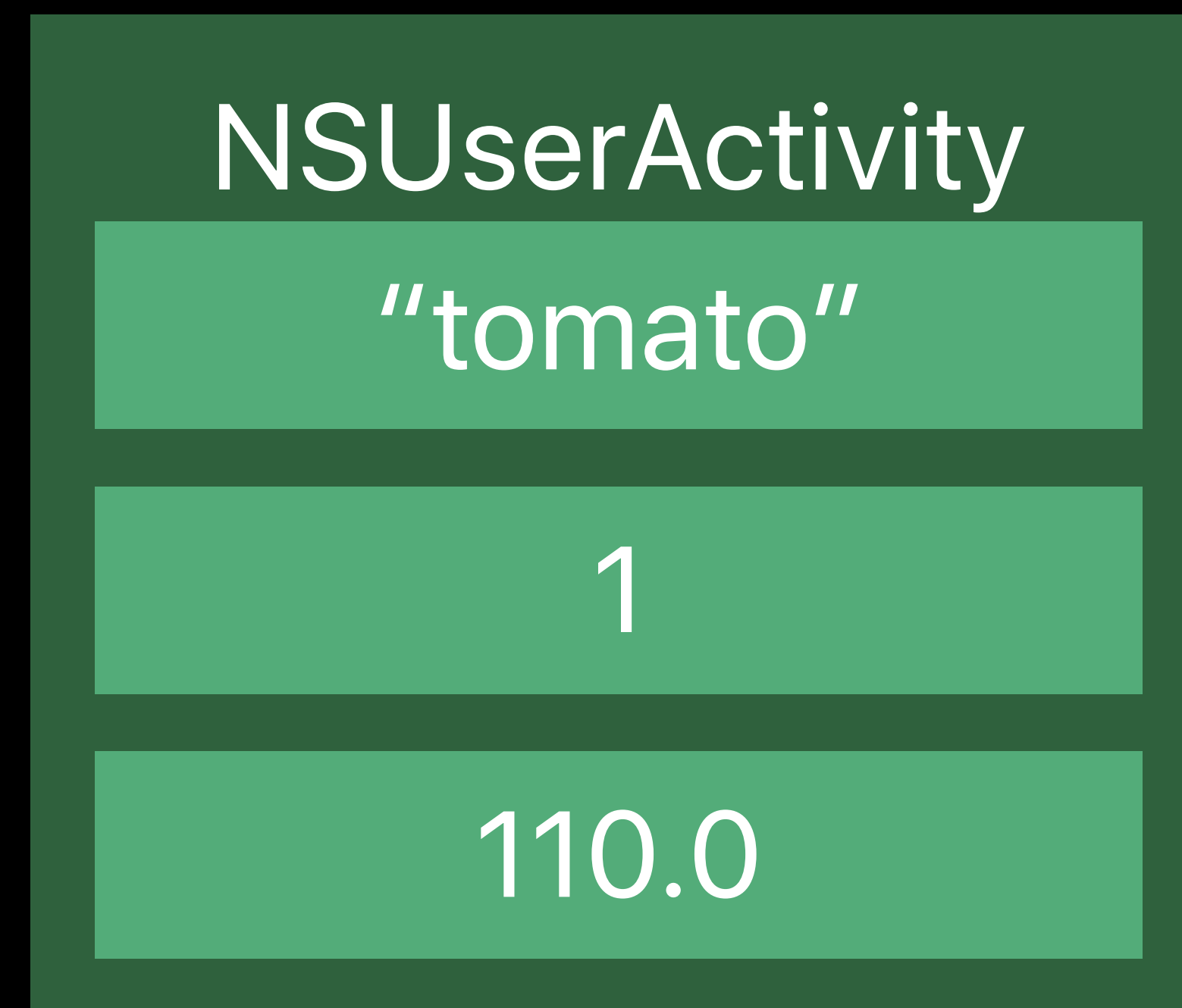
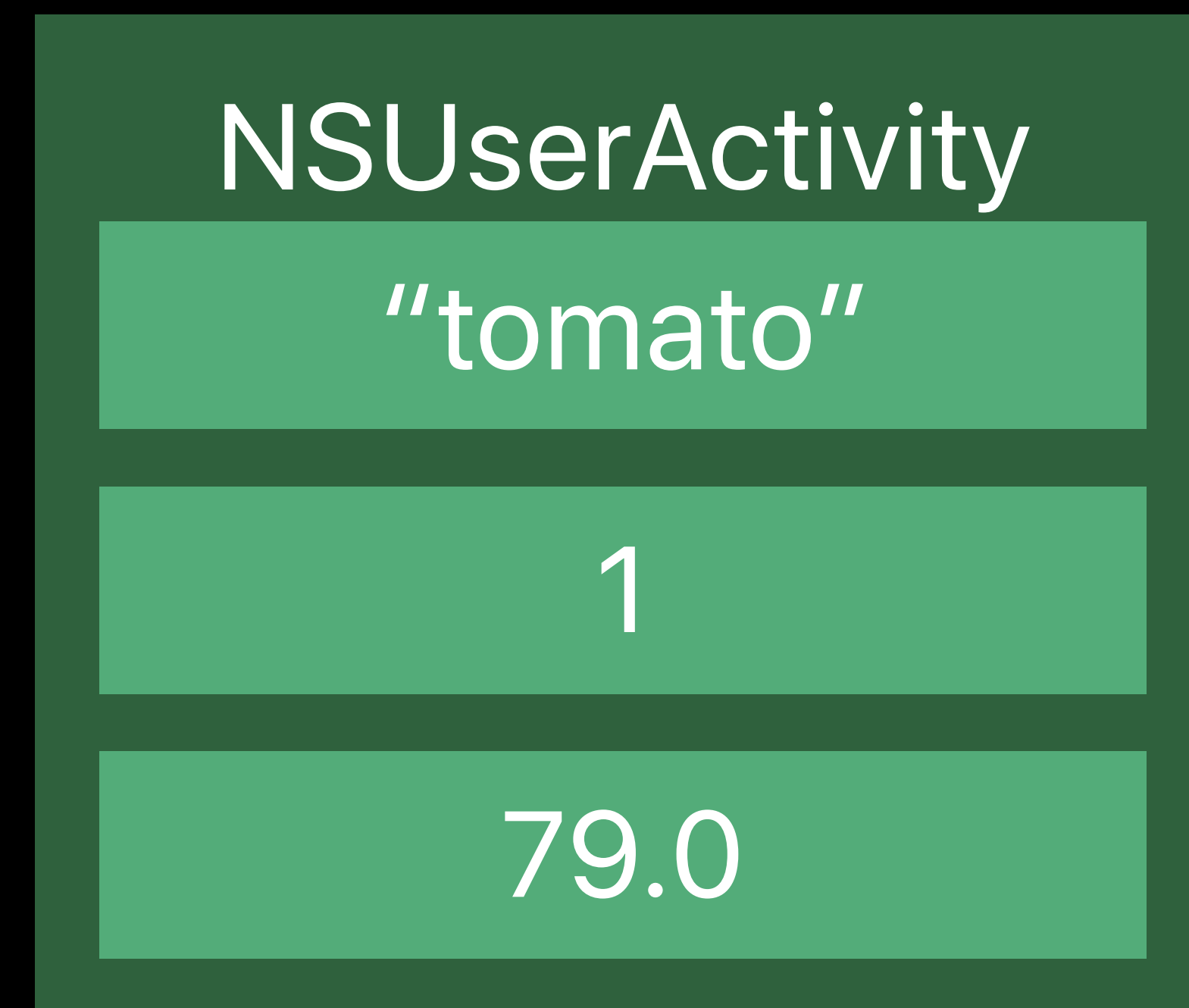
How is a Shortcut Suggested?

NSUserActivity



How is a Shortcut Suggested?

NSUserActivity

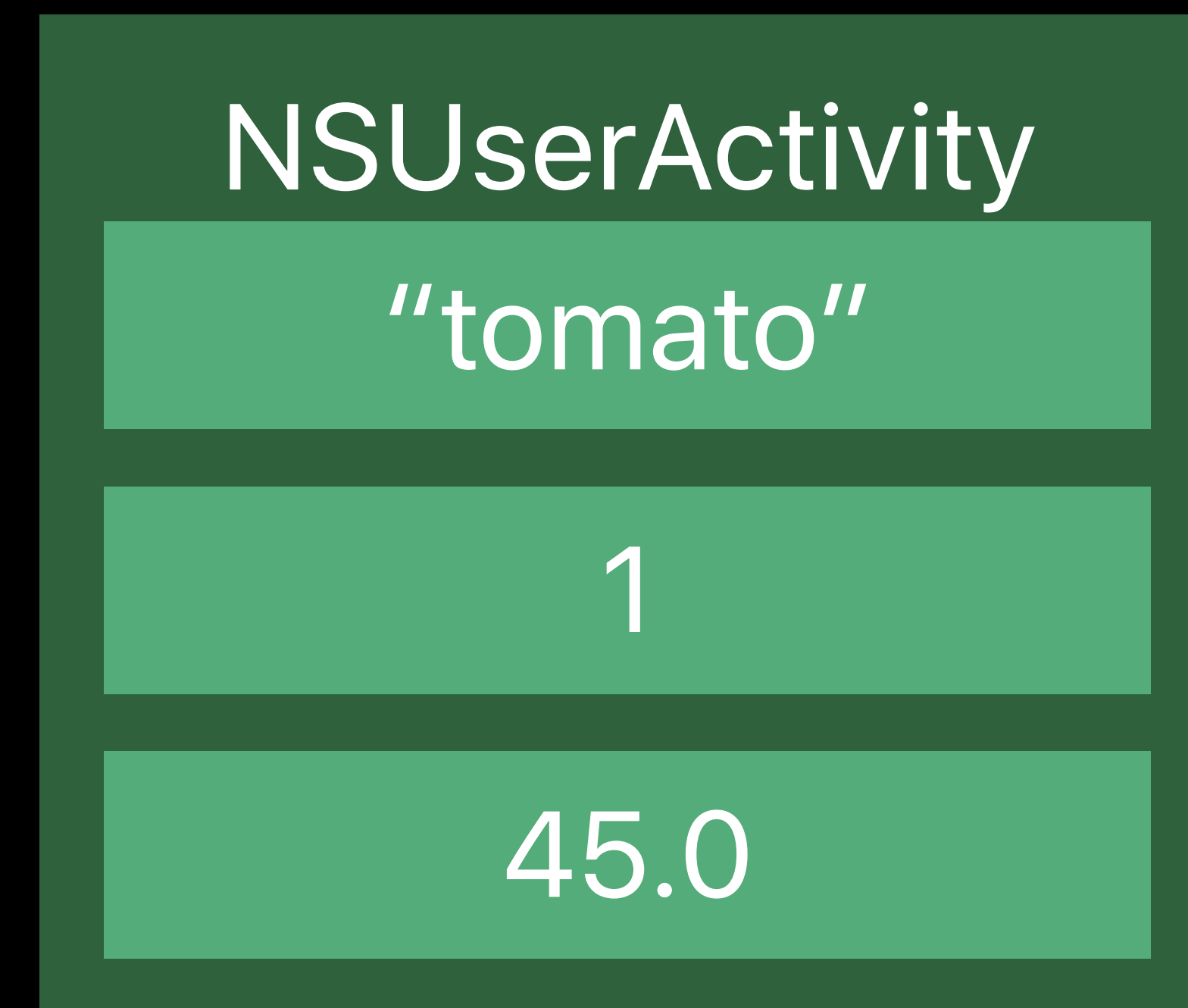
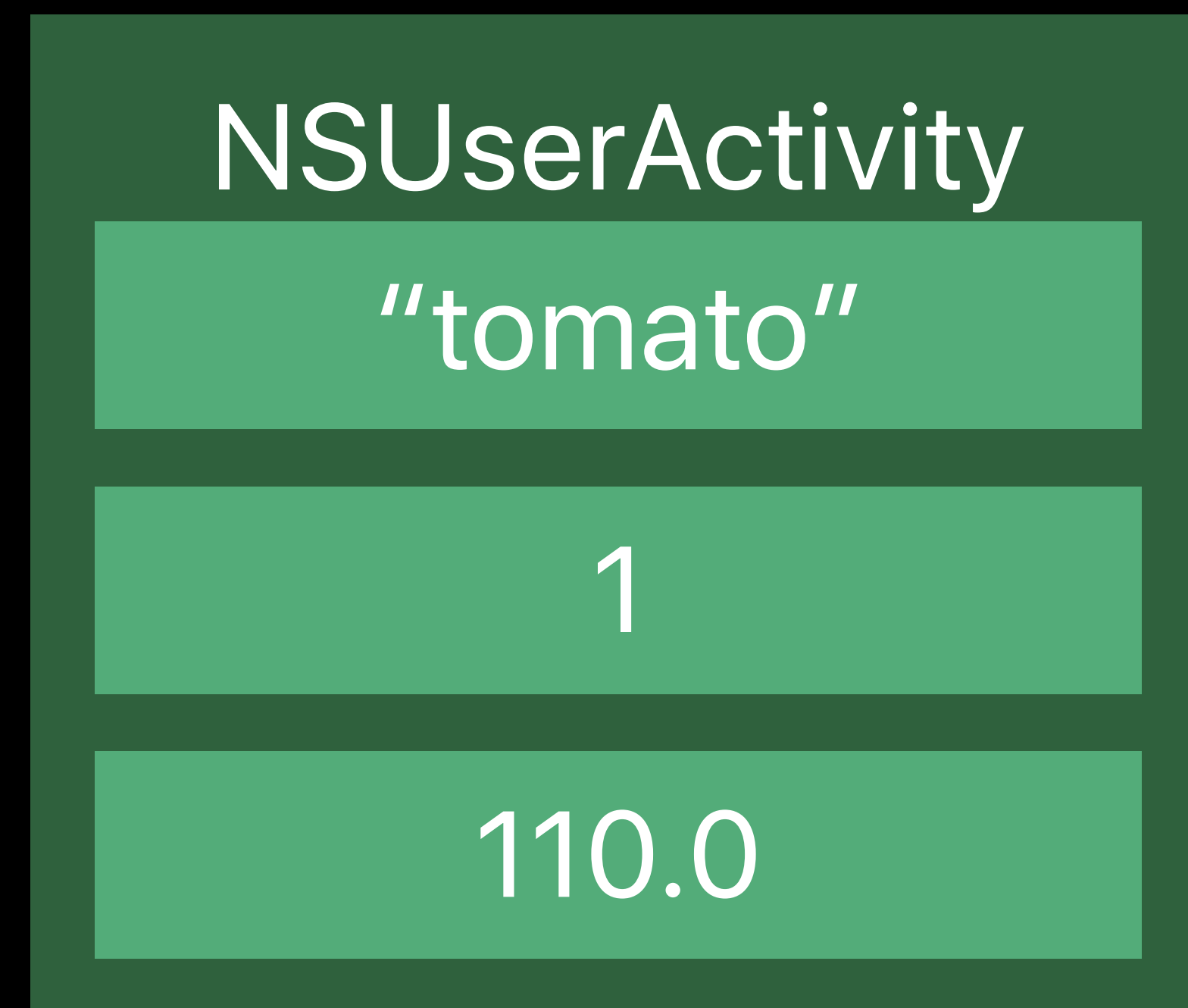
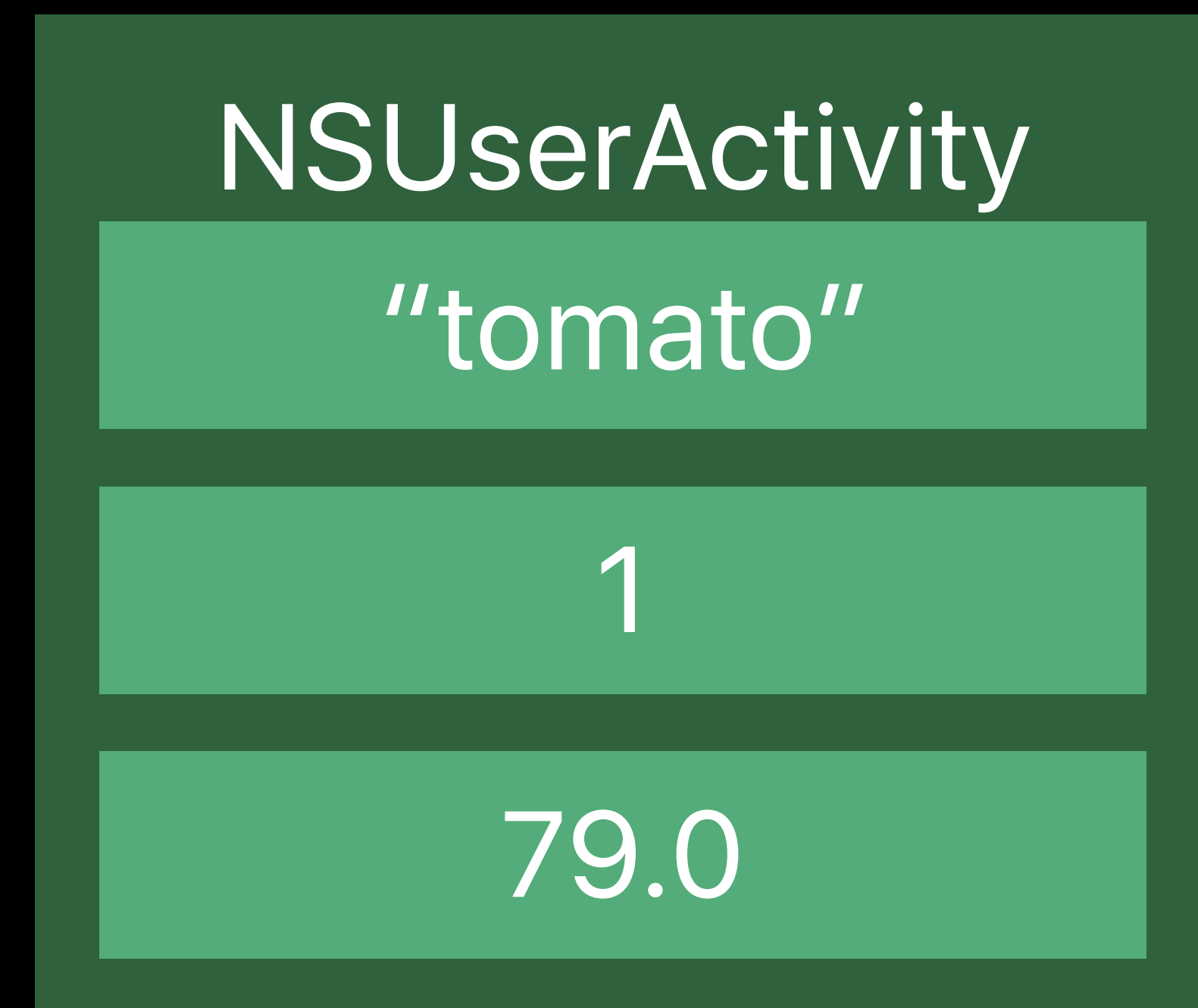


Time

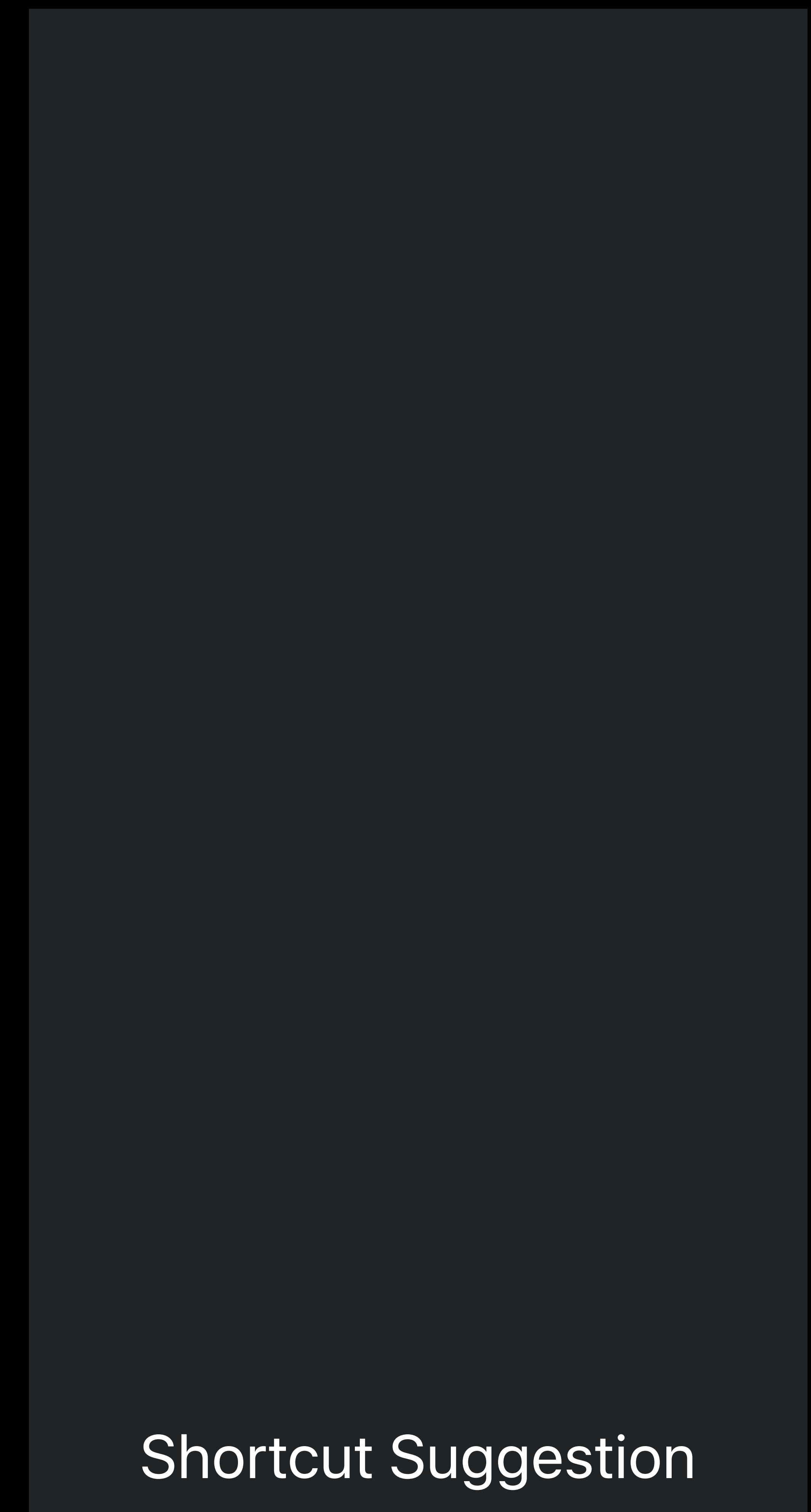


How is a Shortcut Suggested?

NSUserActivity



...



Time



How is a Shortcut Suggested?

NSUserActivity

NSUserActivity
"tomato"
1
79.0

NSUserActivity
"tomato"
1
110.0

NSUserActivity
"tomato"
1
45.0

...



Shortcut Suggestion

Time



How is a Shortcut Suggested?

NSUserActivity

requiredUserInfoKeys

- Minimal amount of information necessary for restoration

How is a Shortcut Suggested?

NSUserActivity

requiredUserInfoKeys

- Minimal amount of information necessary for restoration
- Used to find patterns

How is a Shortcut Suggested?

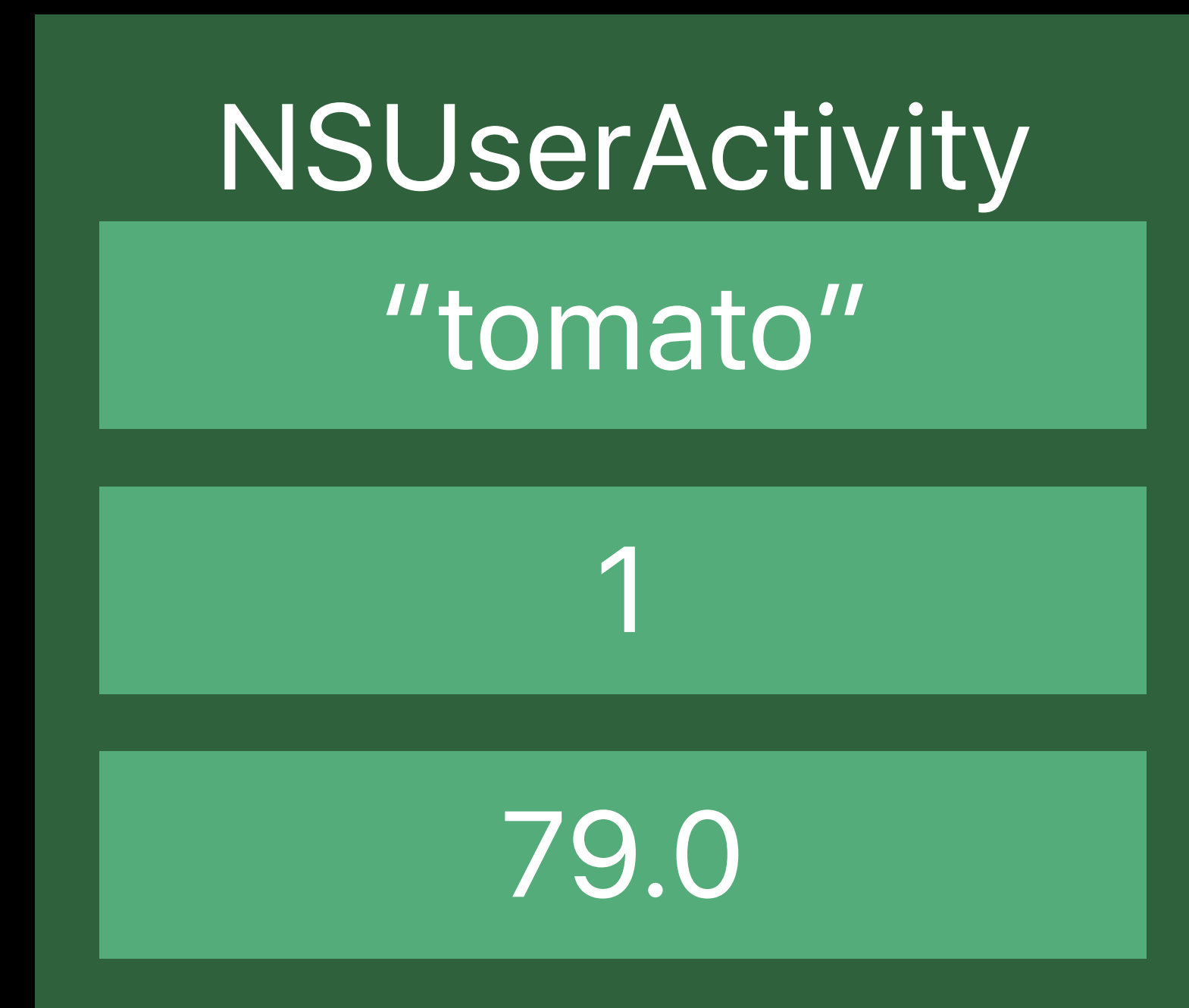
NSUserActivity

Time



How is a Shortcut Suggested?

NSUserActivity

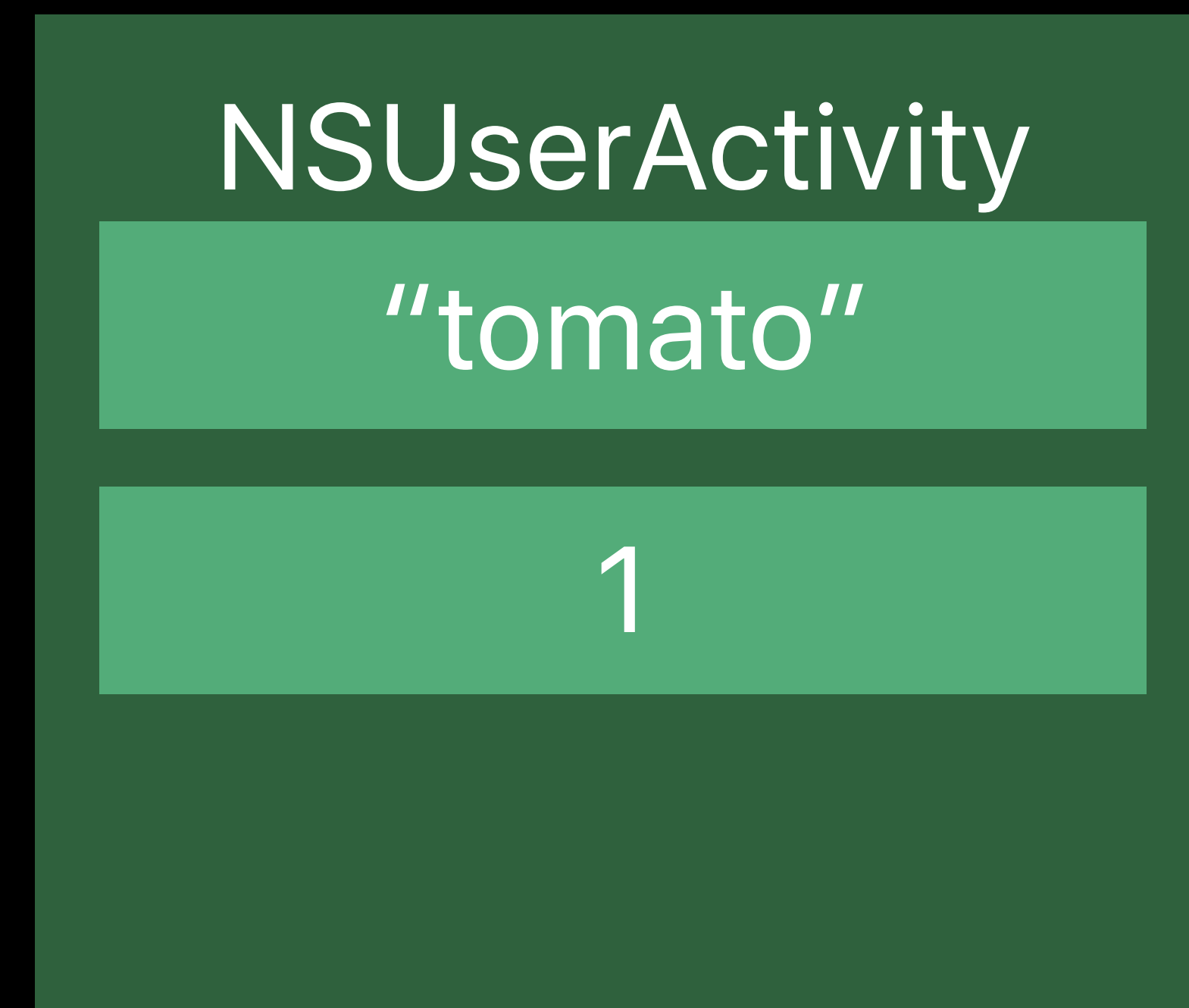


Time



How is a Shortcut Suggested?

NSUserActivity

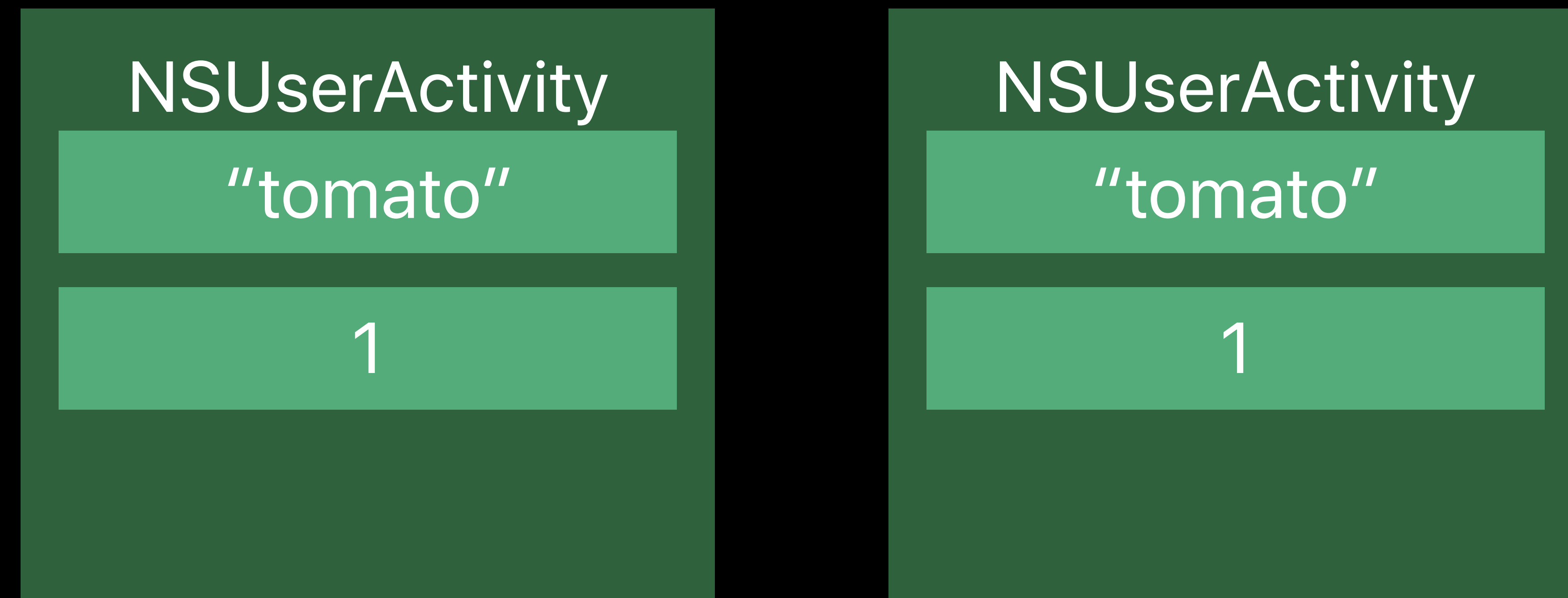


Time



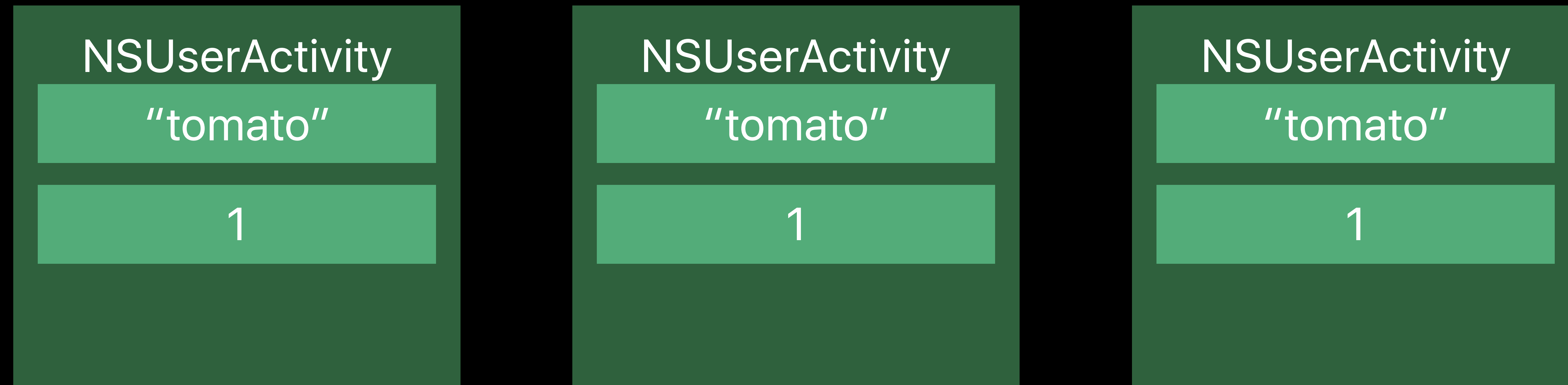
How is a Shortcut Suggested?

NSUserActivity



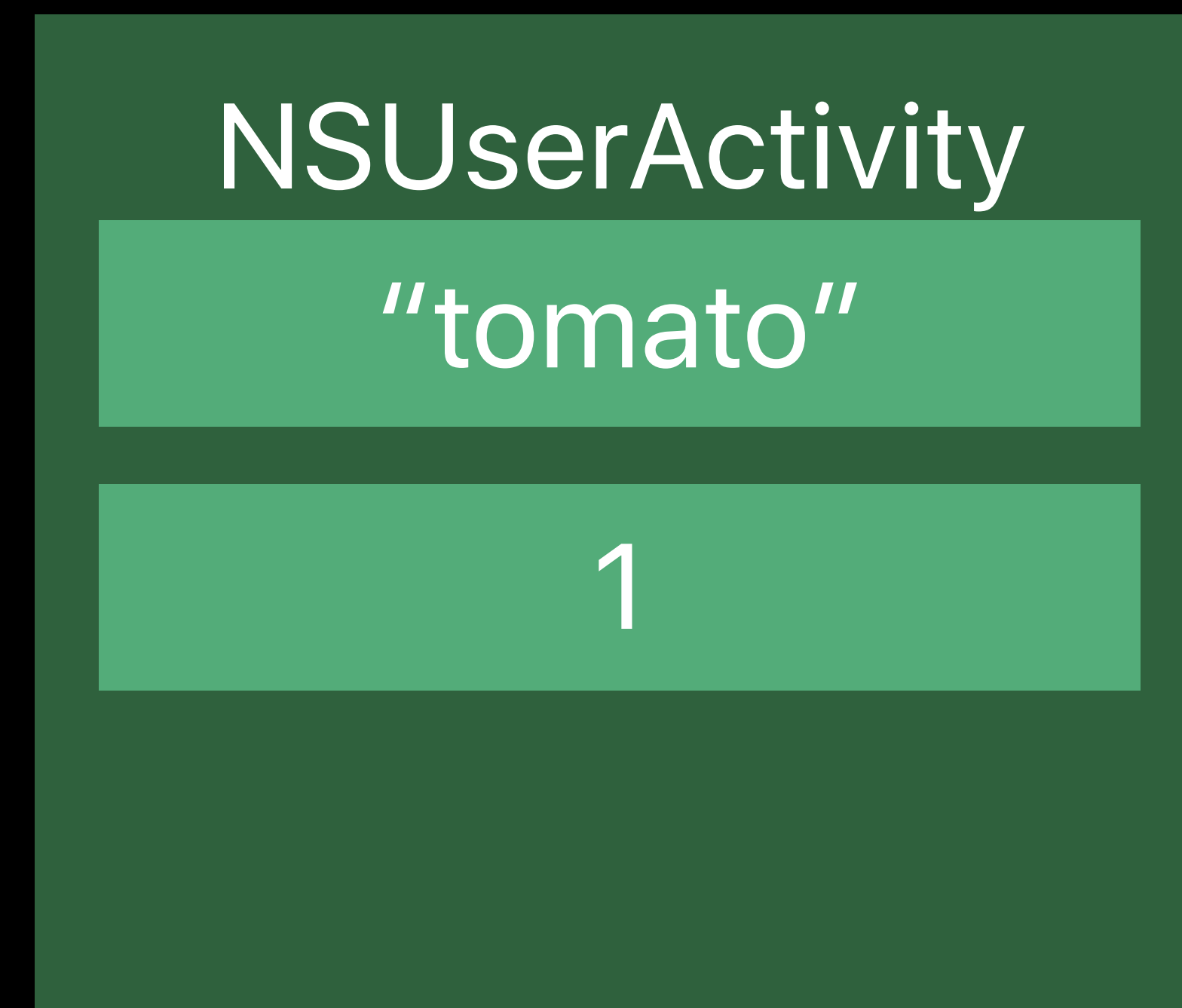
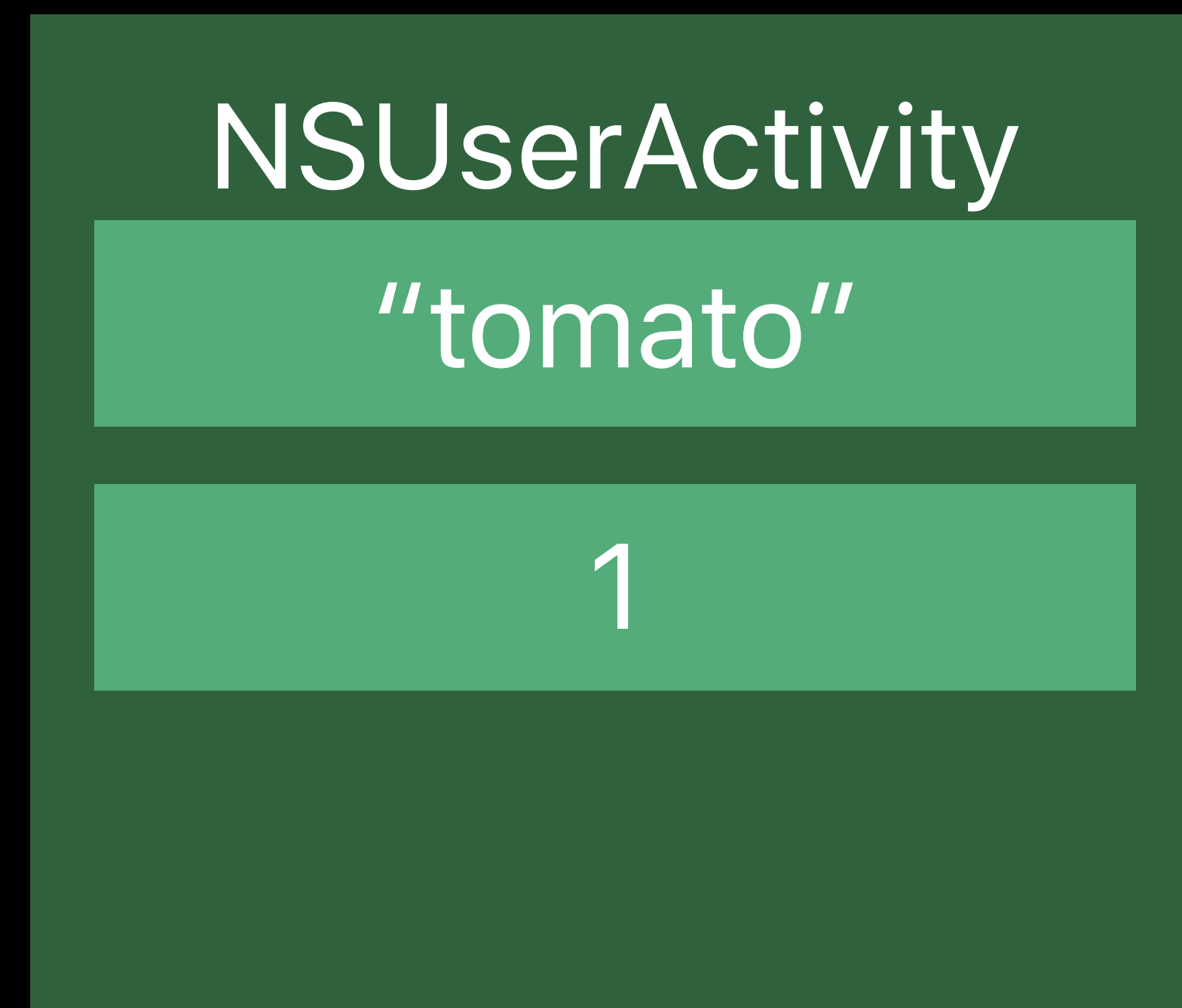
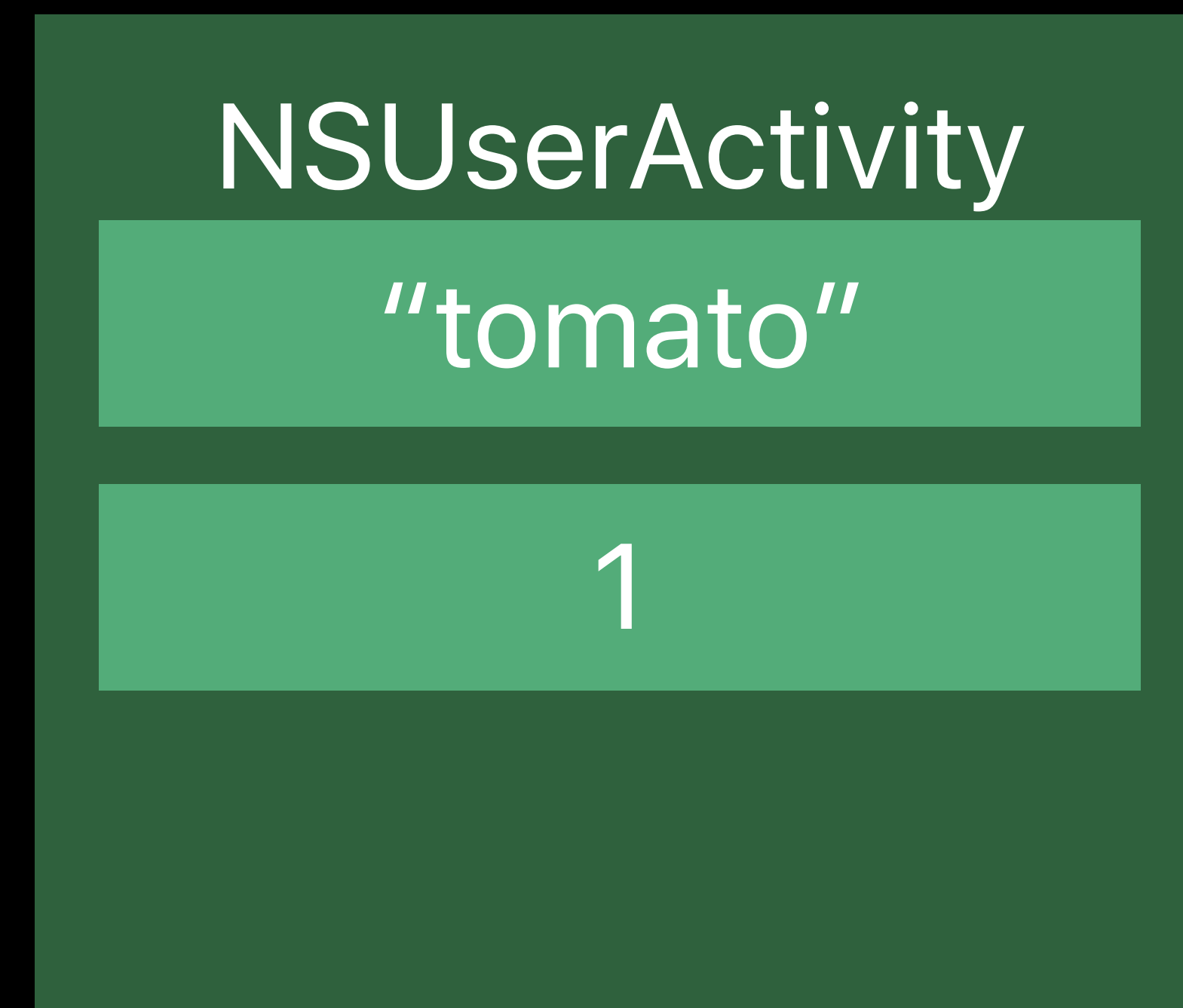
How is a Shortcut Suggested?

NSUserActivity

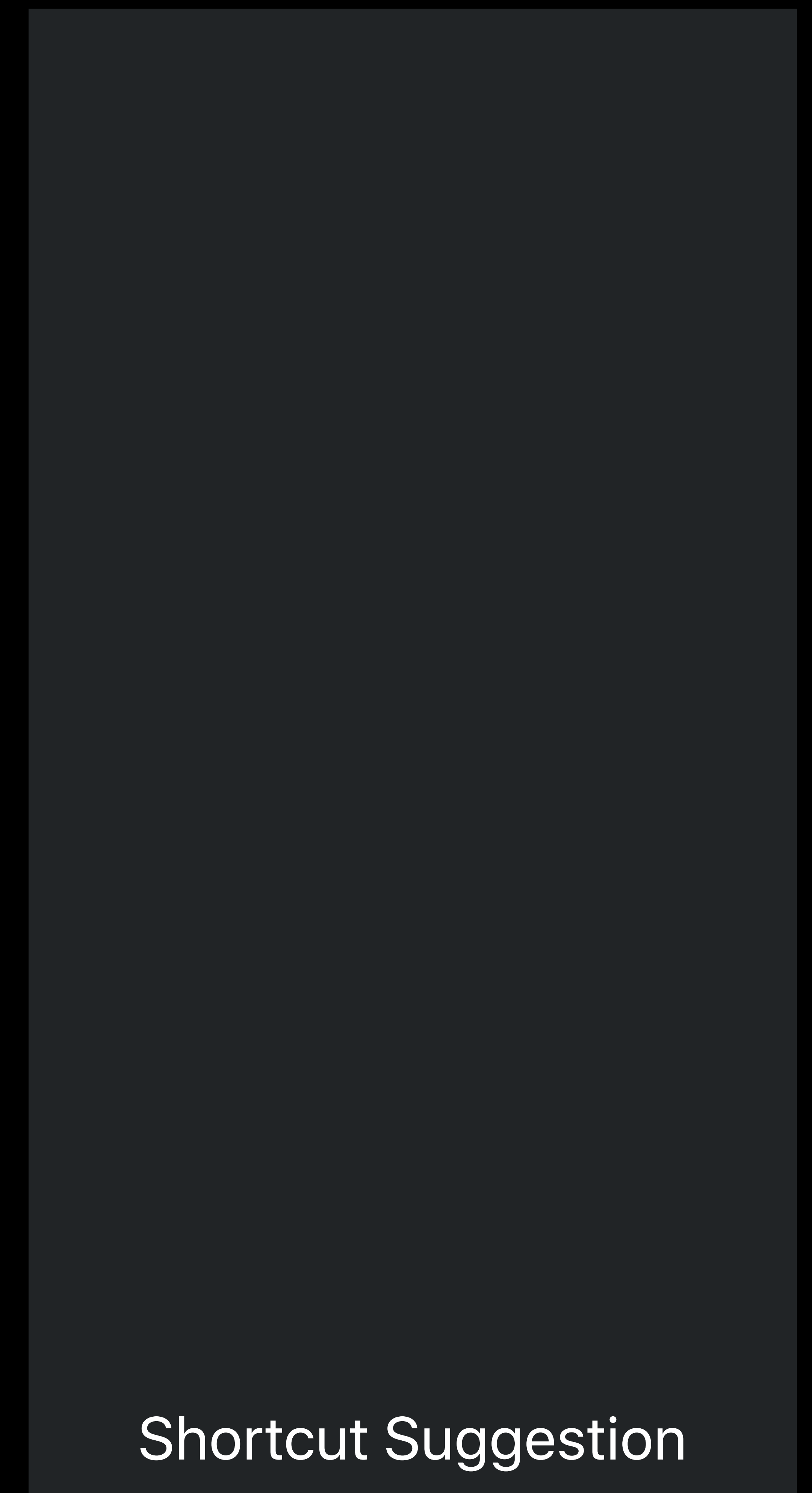


How is a Shortcut Suggested?

NSUserActivity



...

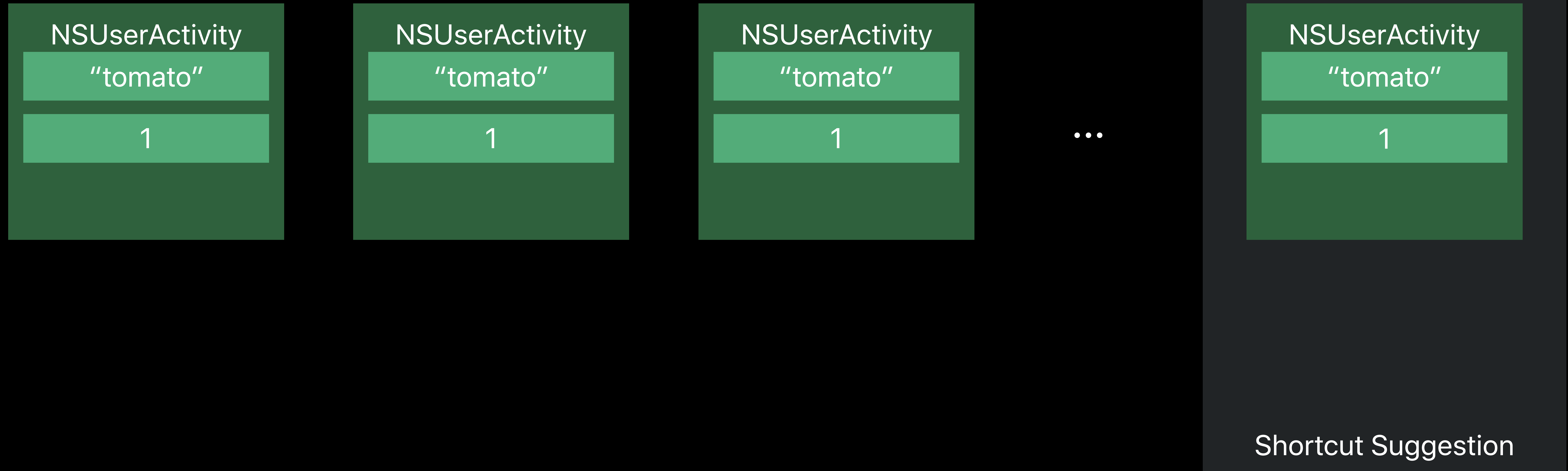


Time

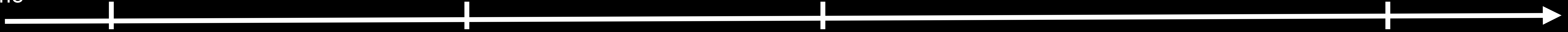


How is a Shortcut Suggested?

NSUserActivity



Time



How is a Shortcut Suggested?

Intents

Shortcut Types

▼ **Shortcut Types**

Parameter Combination

soup
soup, options
soup, quantity, options
soup, quantity

+ -

Title

Subtitle

Background Supports background execution

How is a Shortcut Suggested?

Intents

OrderSoup

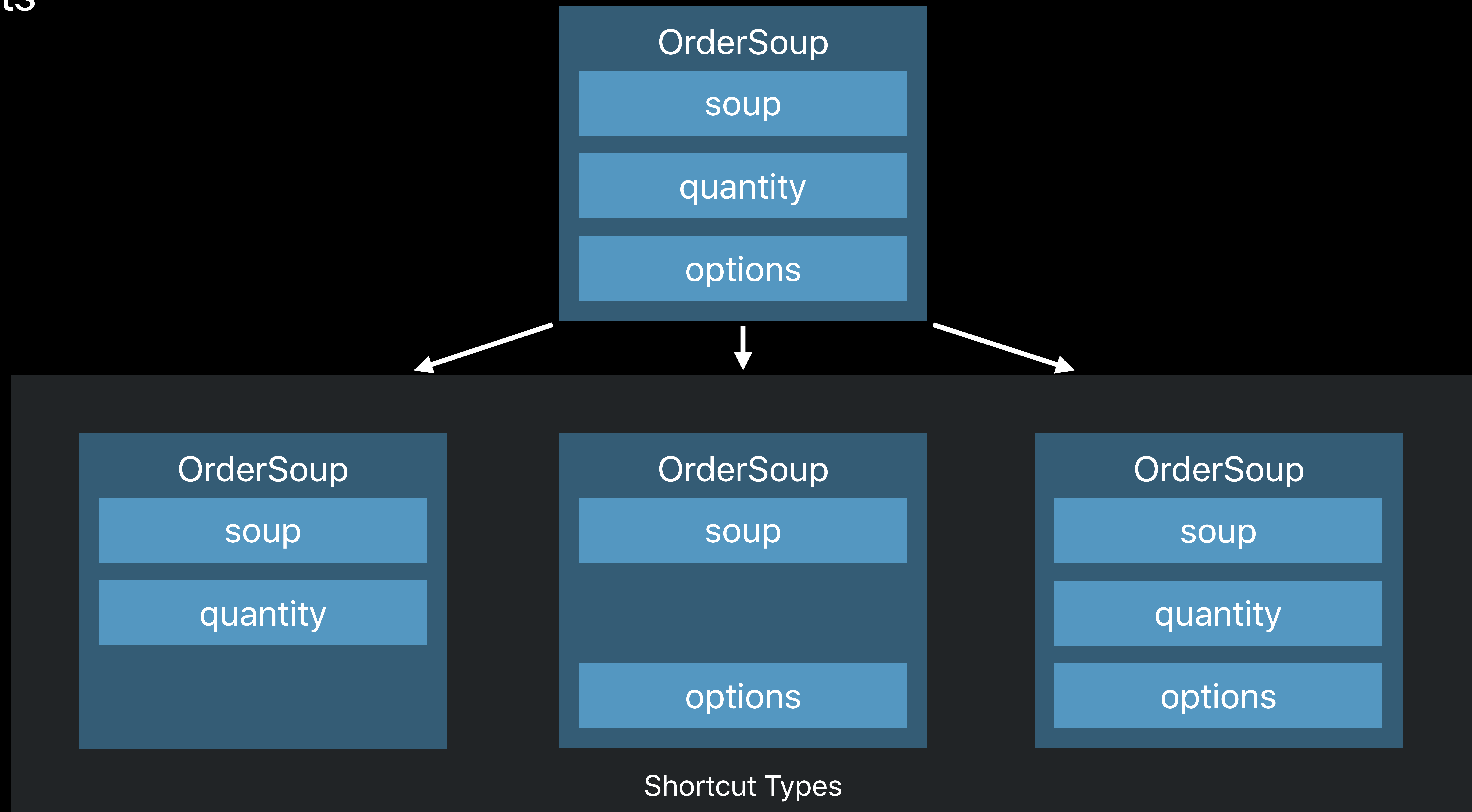
soup

quantity

options

How is a Shortcut Suggested?

Intents



How is a Shortcut Suggested?

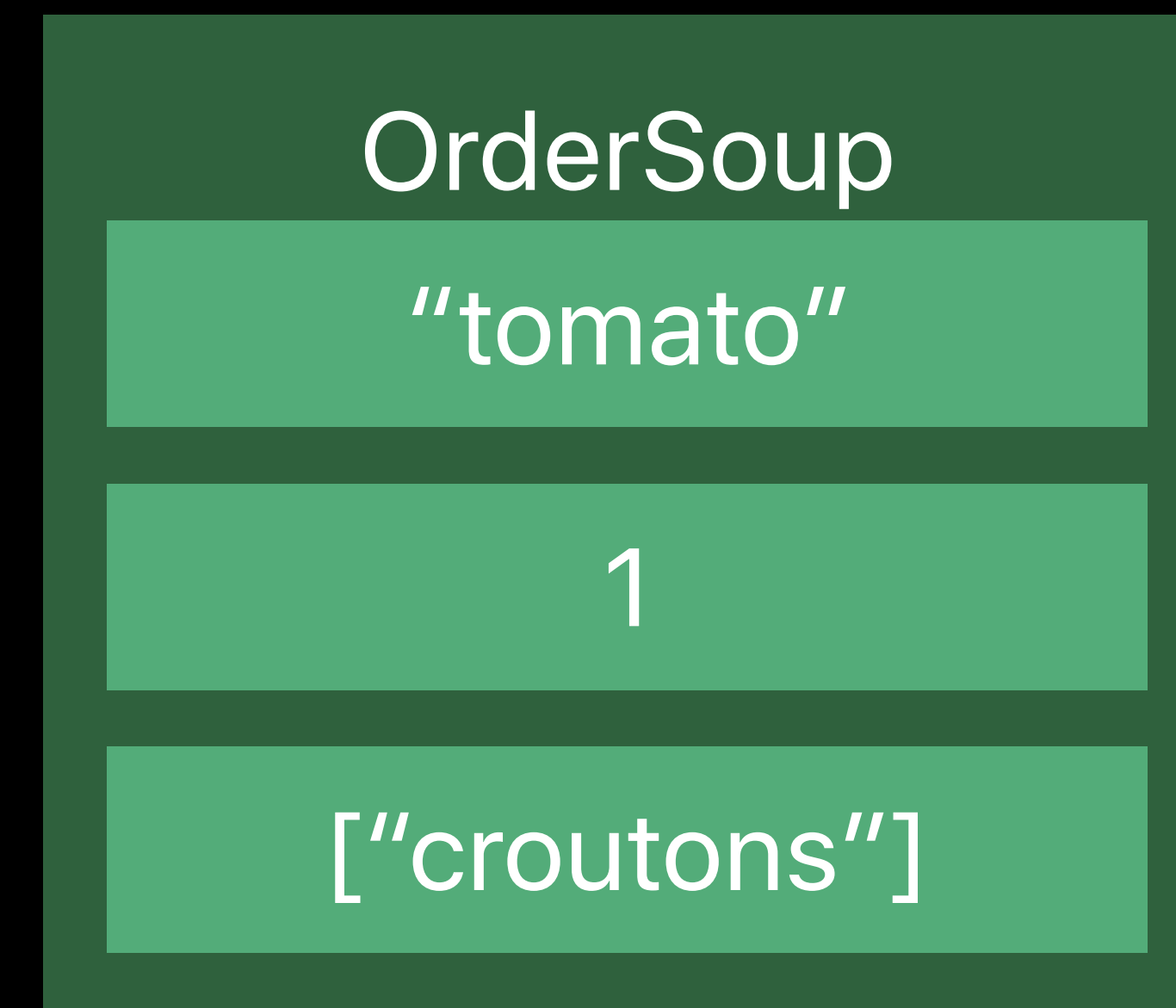
Intents

Time



How is a Shortcut Suggested?

Intents



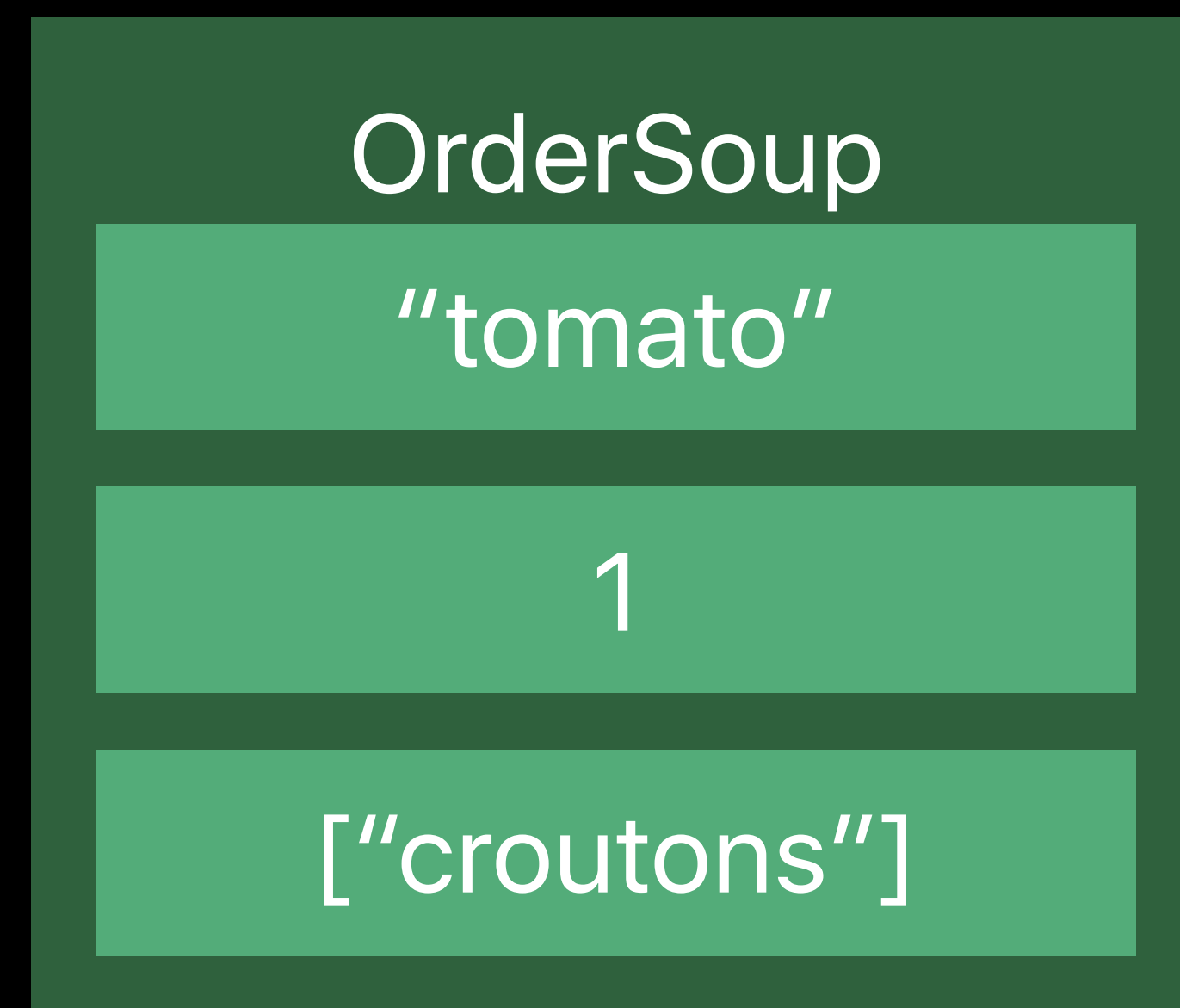
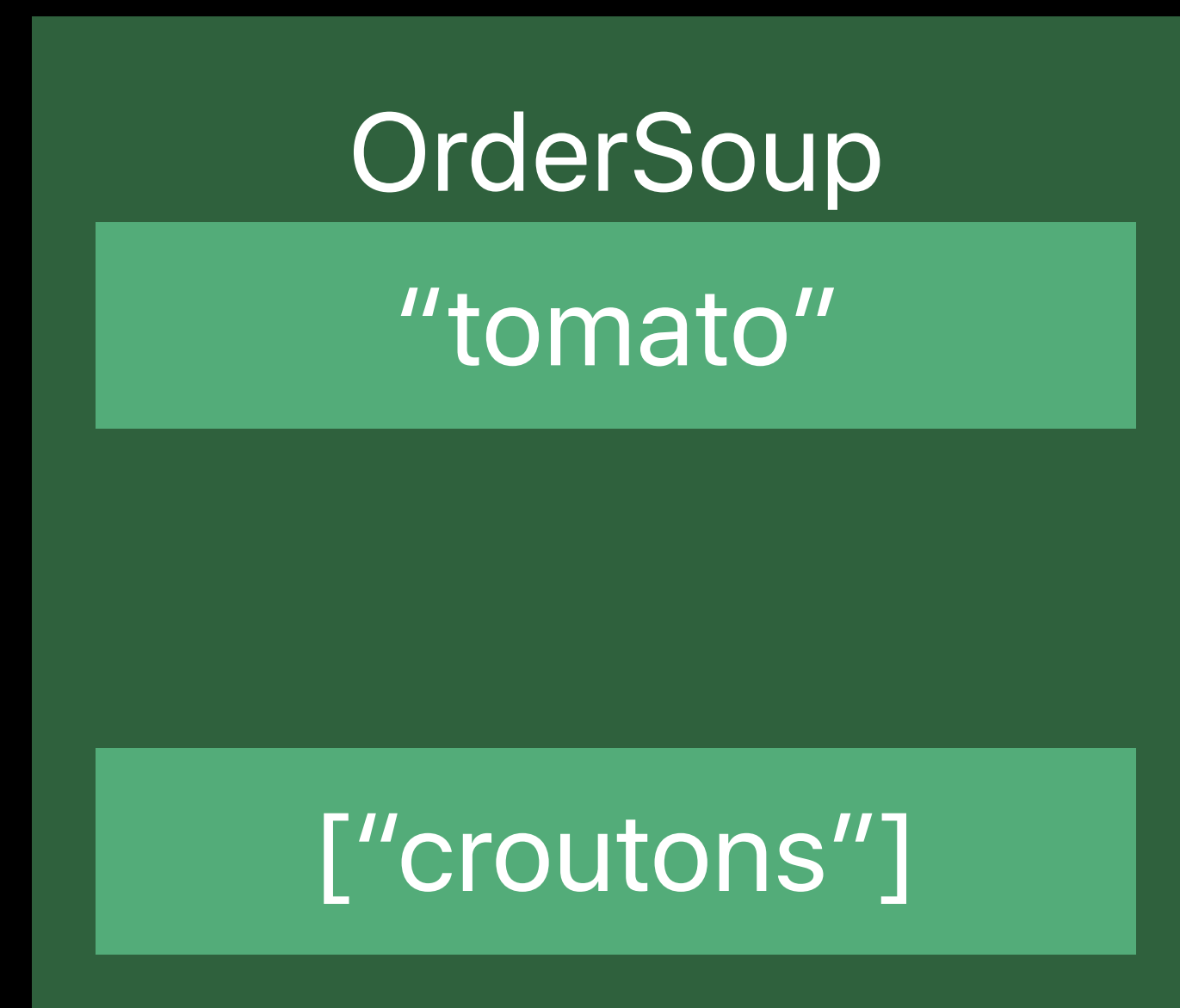
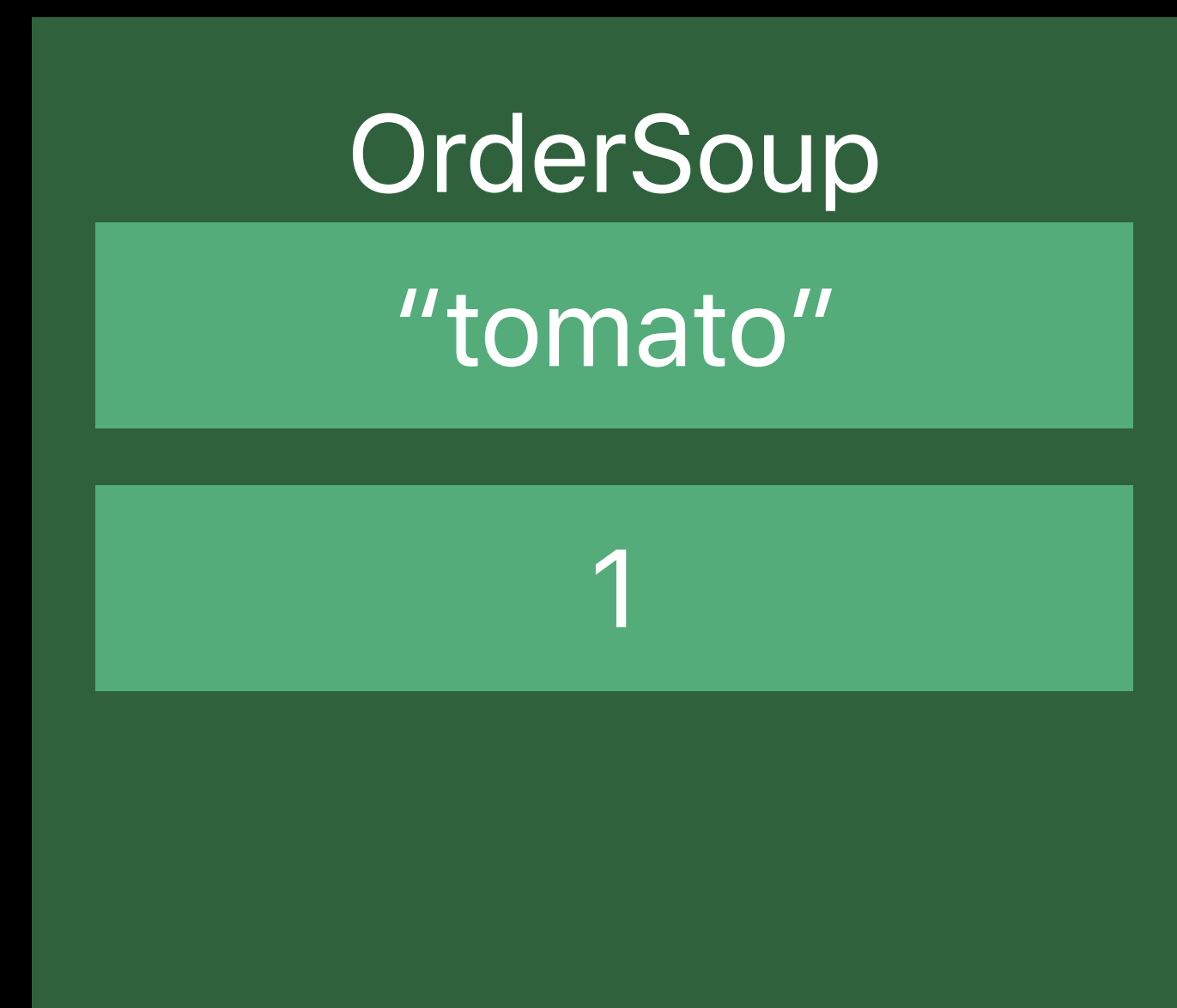
Monday, 11:57 AM

Time



How is a Shortcut Suggested?

Intents



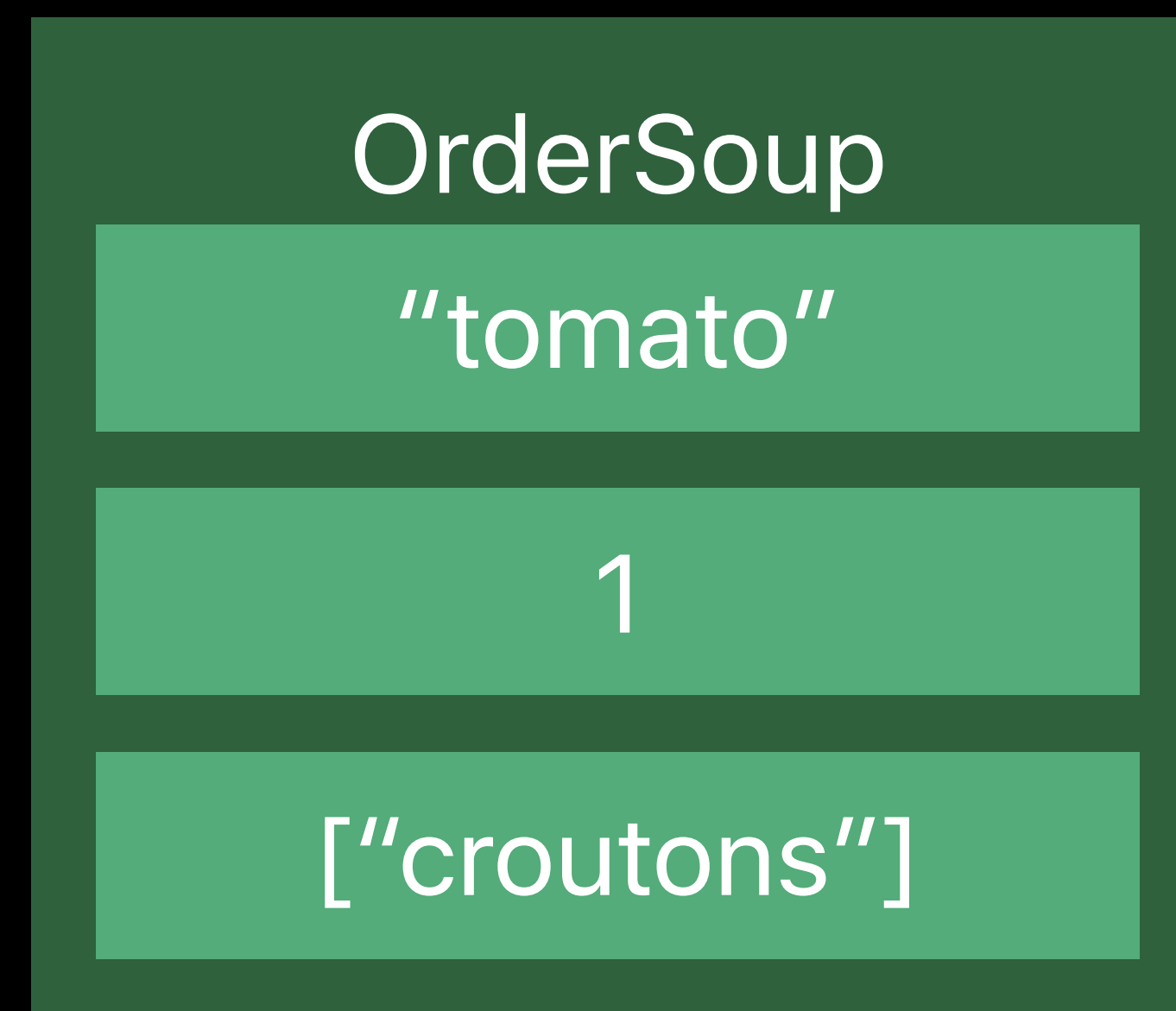
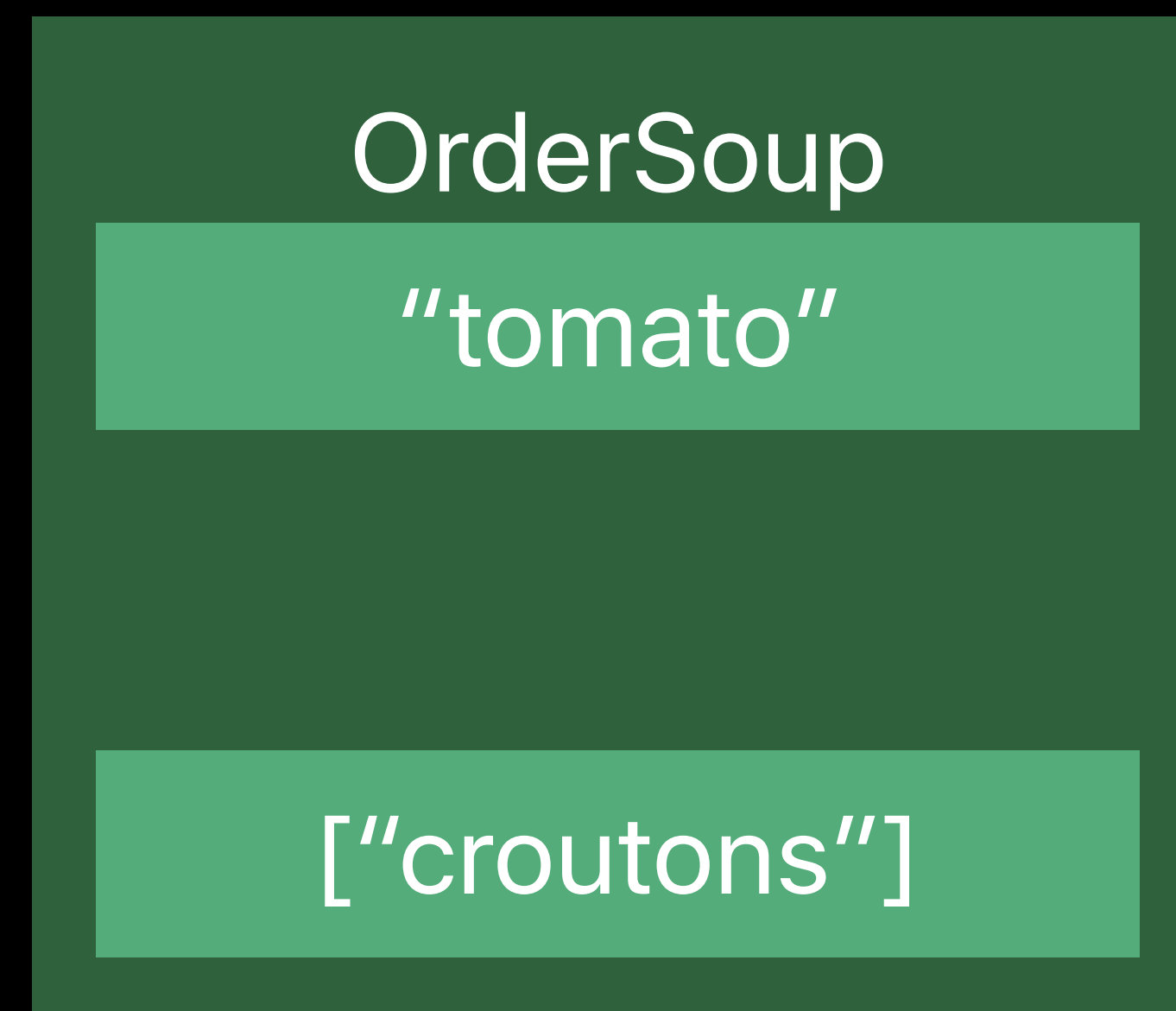
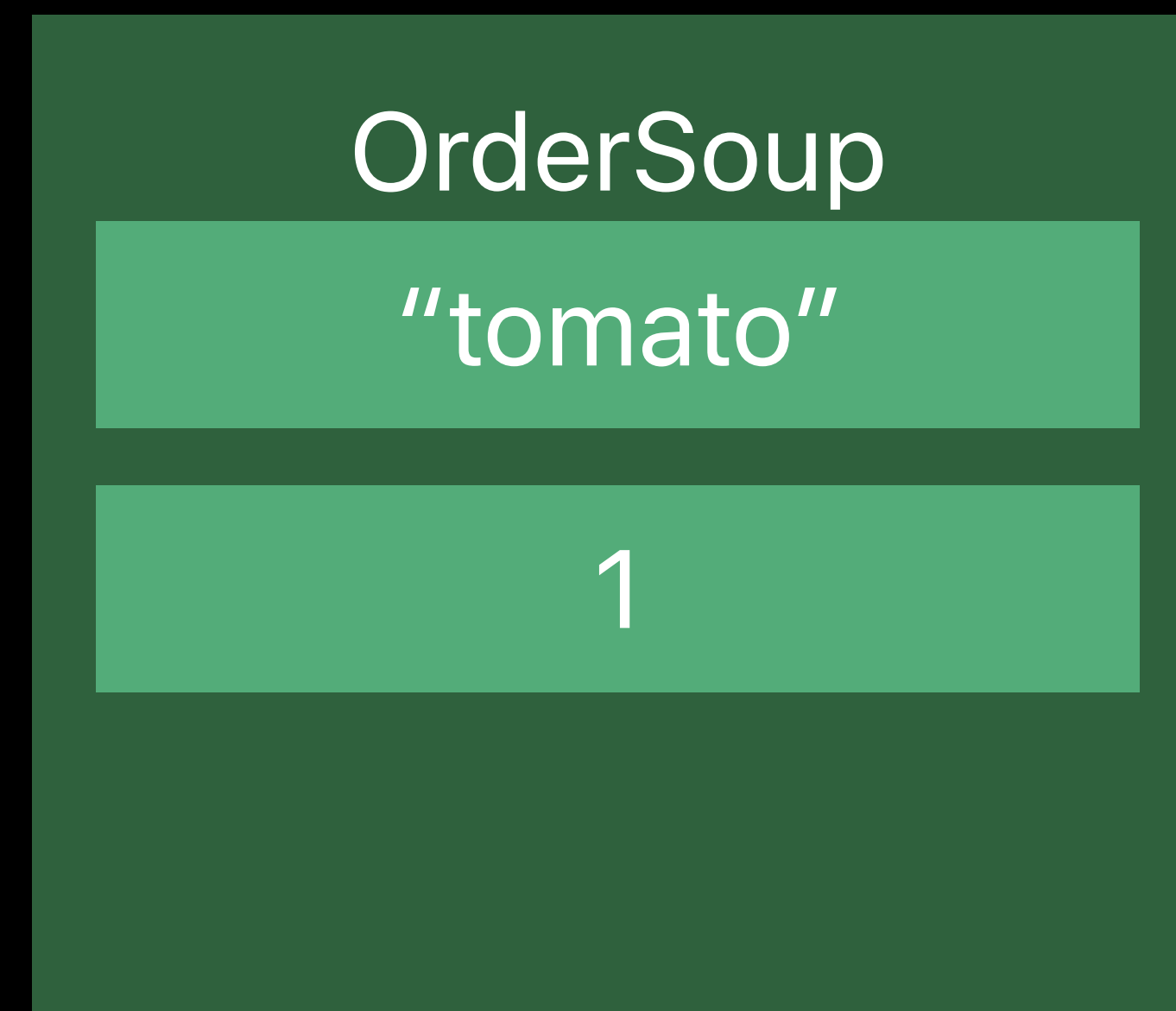
Monday, 11:57 AM

Time

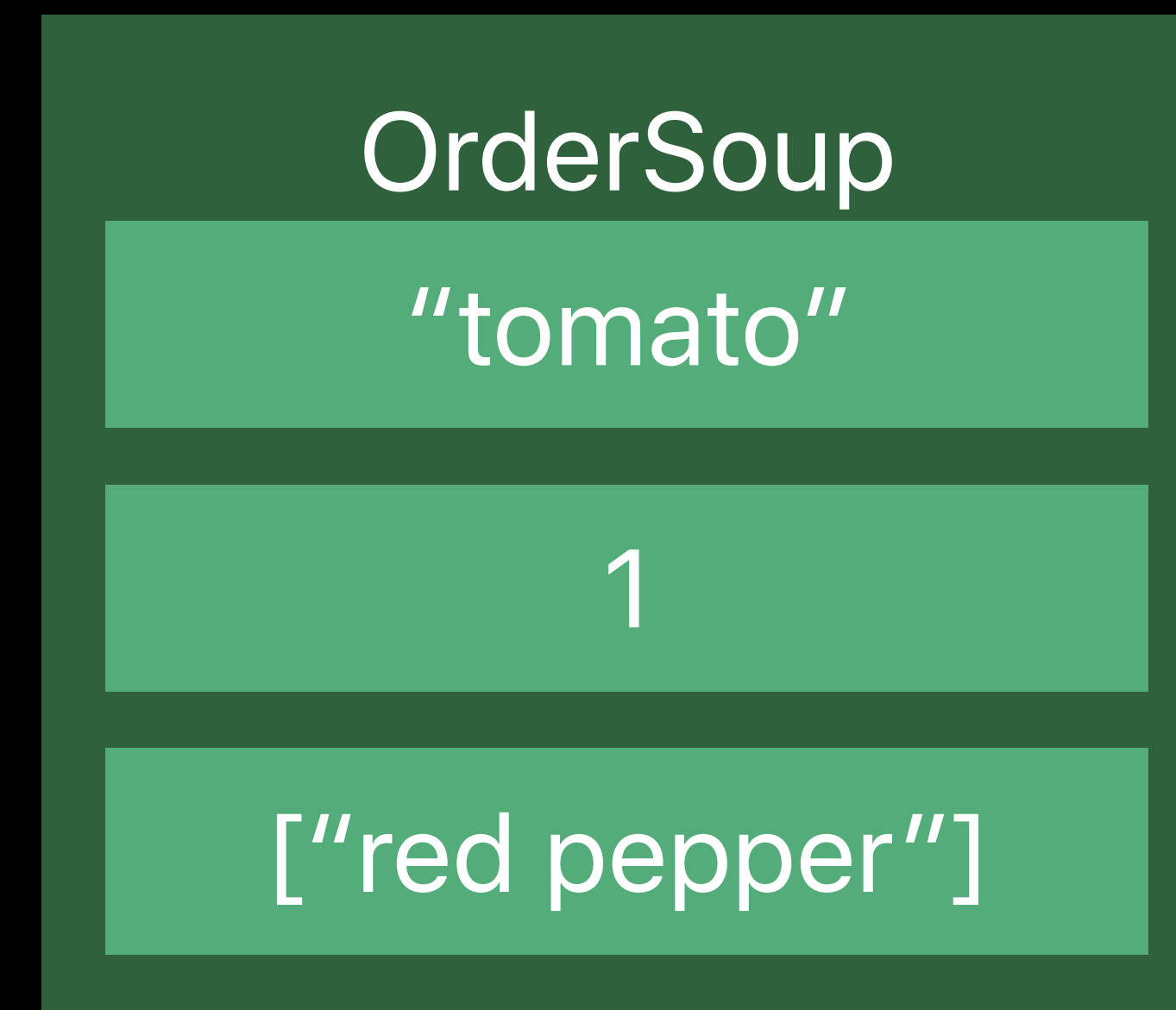


How is a Shortcut Suggested?

Intents



Monday, 11:57 AM



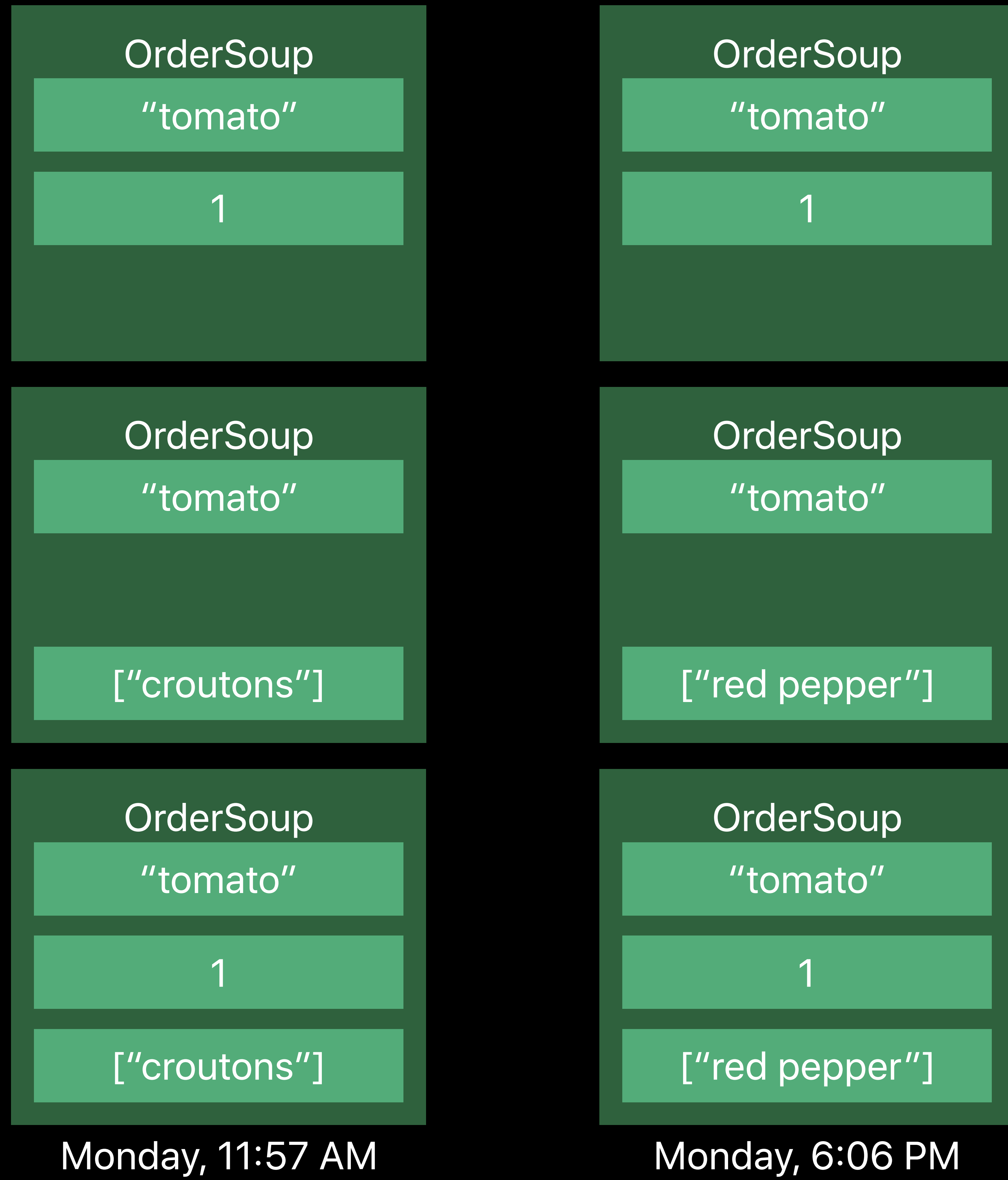
Monday, 6:06 PM

Time



How is a Shortcut Suggested?

Intents

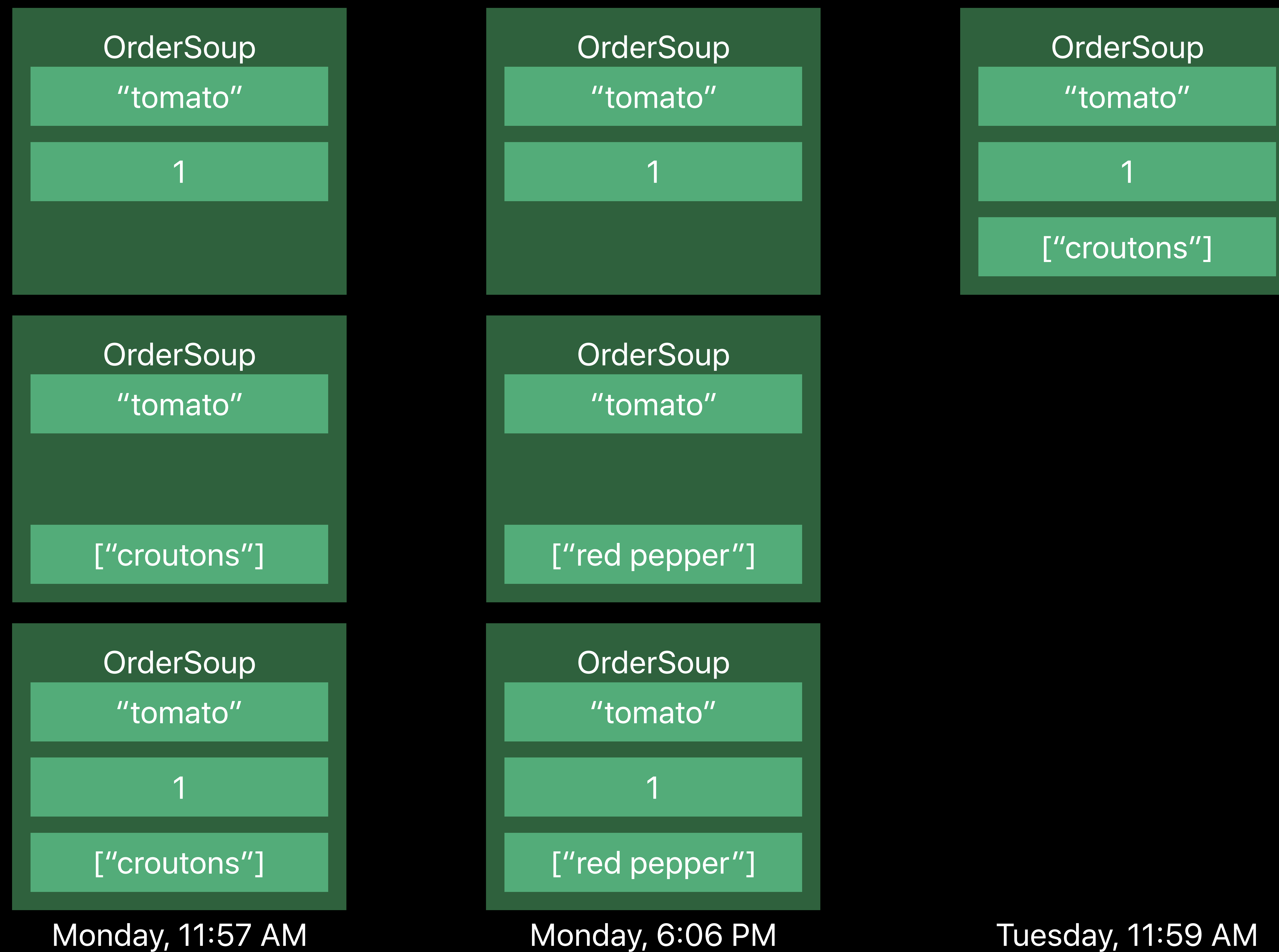


Time



How is a Shortcut Suggested?

Intents

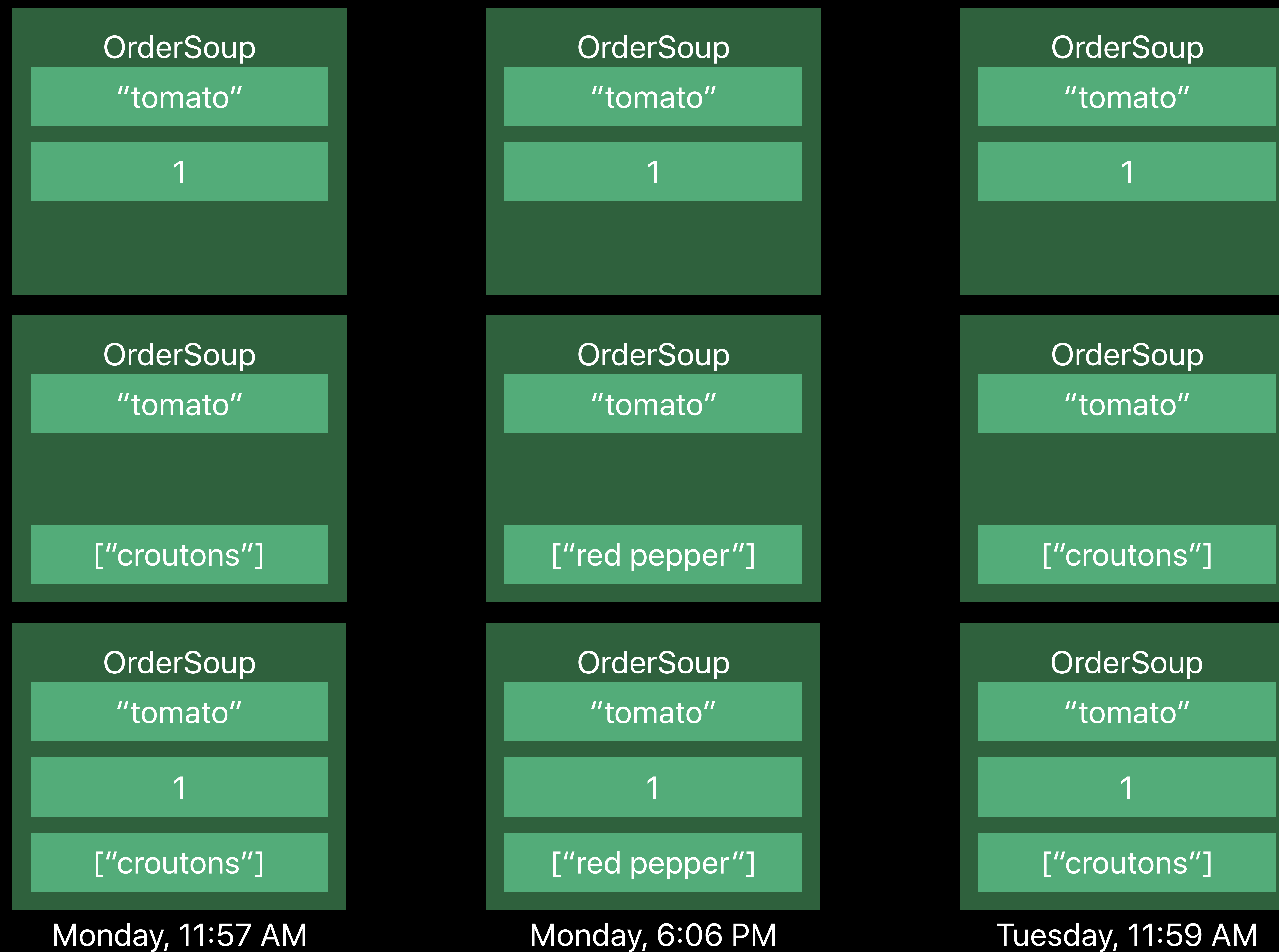


Time



How is a Shortcut Suggested?

Intents



Time



How is a Shortcut Suggested?

Intents

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["croutons"]

OrderSoup
"tomato"
1
["croutons"]

Monday, 11:57 AM

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["red pepper"]

OrderSoup
"tomato"
1
["red pepper"]

Monday, 6:06 PM

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["croutons"]

OrderSoup
"tomato"
1
["croutons"]

Tuesday, 11:59 AM

...

Shortcut Suggestion

Friday, 12:00 PM

Time



How is a Shortcut Suggested?

Intents

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["croutons"]

OrderSoup
"tomato"
1
["croutons"]

Monday, 11:57 AM

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["red pepper"]

OrderSoup
"tomato"
1
["red pepper"]

Monday, 6:06 PM

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["croutons"]

OrderSoup
"tomato"
1
["croutons"]

Tuesday, 11:59 AM

...

Shortcut Suggestion

Friday, 12:00 PM

Time



How is a Shortcut Suggested?

Intents

OrderSoup
"tomato"
1

OrderSoup
"tomato"
1

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["croutons"]

OrderSoup
"tomato"
["red pepper"]

OrderSoup
"tomato"
["croutons"]

...

OrderSoup
"tomato"
1
["croutons"]

OrderSoup
"tomato"
1
["red pepper"]

OrderSoup
"tomato"
1
["croutons"]

Shortcut Suggestion

Monday, 11:57 AM

Monday, 6:06 PM

Tuesday, 11:59 AM

Friday, 12:00 PM

Time



How is a Shortcut Suggested?

Intents

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["croutons"]

OrderSoup
"tomato"
1
["croutons"]

Monday, 11:57 AM

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["red pepper"]

OrderSoup
"tomato"
1
["red pepper"]

Monday, 6:06 PM

OrderSoup
"tomato"
1

OrderSoup
"tomato"
["croutons"]

OrderSoup
"tomato"
1
["croutons"]

Tuesday, 11:59 AM

...

Shortcut Suggestion

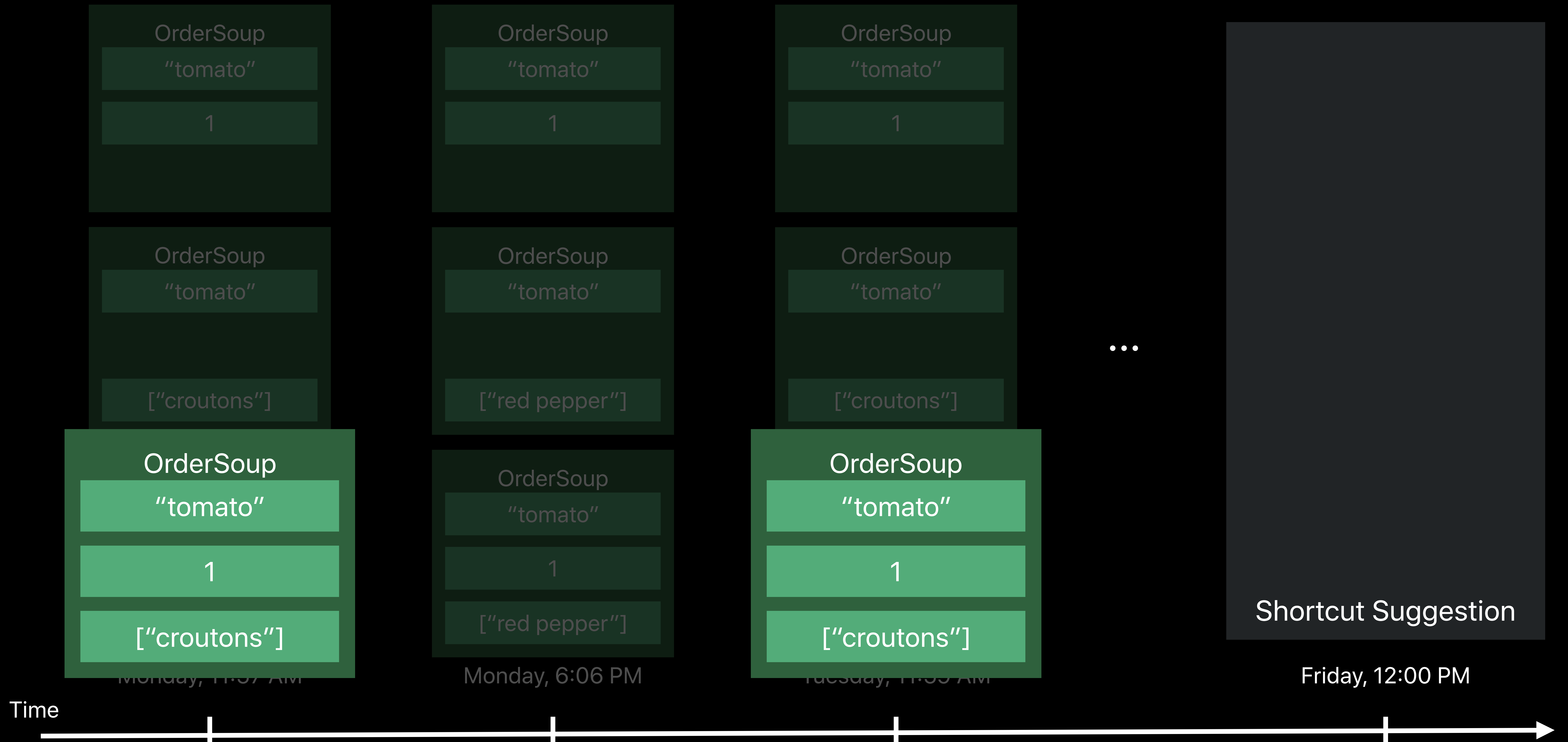
Friday, 12:00 PM

Time



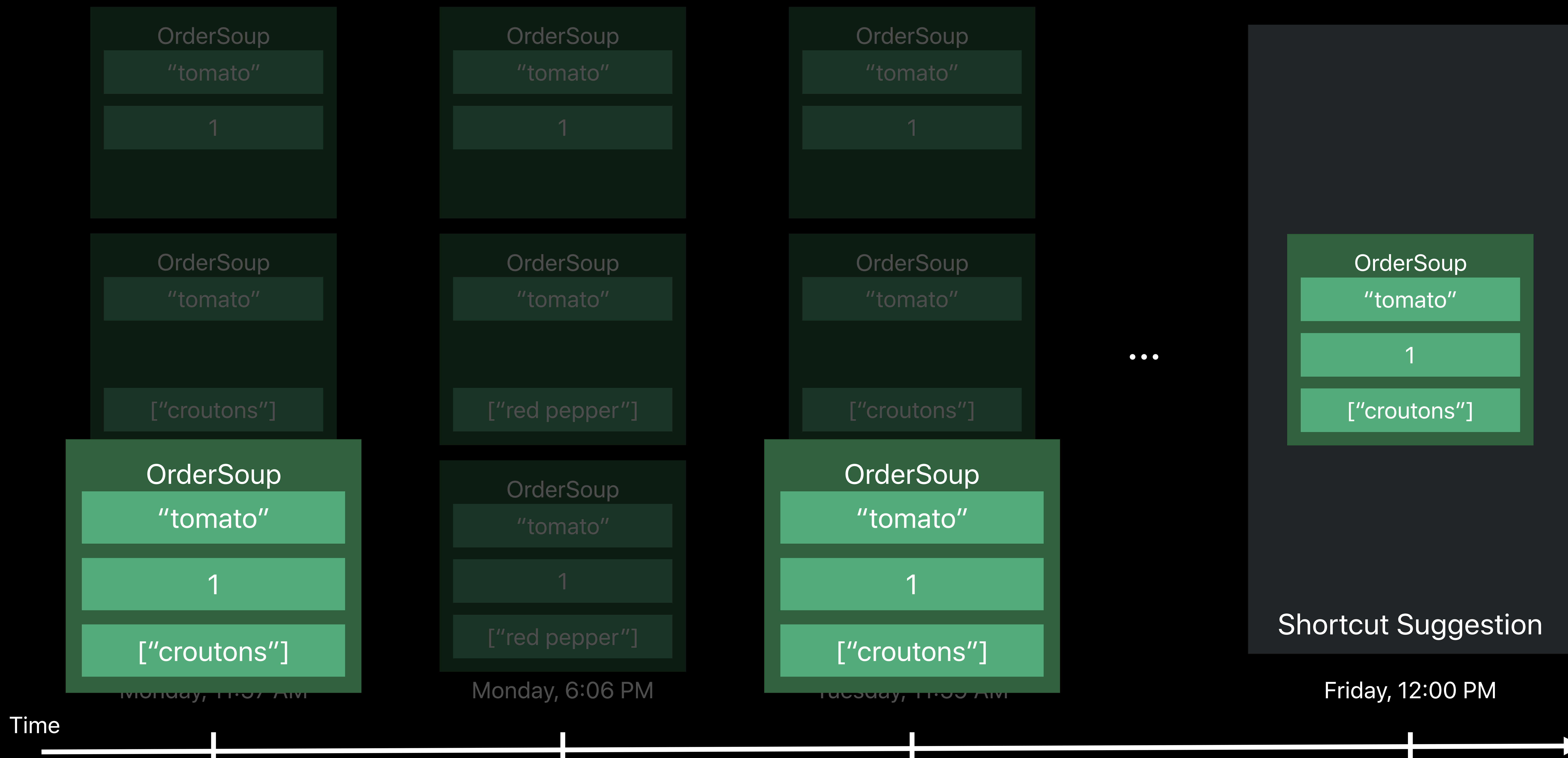
How is a Shortcut Suggested?

Intents



How is a Shortcut Suggested?

Intents



What Makes a Good Donation?

Likely to be repeated

What Makes a Good Donation?

Likely to be repeated

Ensure the payload you're donating is consistent across donations

What Makes a Good Donation?

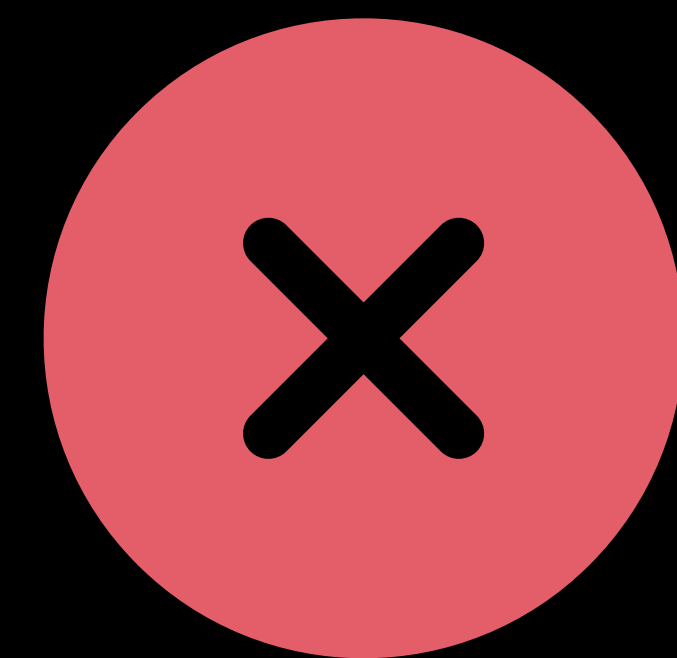
Likely to be repeated

Ensure the payload you're donating is consistent across donations

Don't include timestamps

What Makes a Good Donation?

"Show my meetings
for June 5th"



"Show my meetings
for today"



What Makes a Good Donation?

Likely to be repeated

Ensure the payload you're donating is consistent across donations

Don't include timestamps

Donate once and only once for each user action

Custom Intents

Parameters

Use enums when the values for a parameter are clearly bounded

Custom Intents

Parameters

Use enums when the values for a parameter are clearly bounded

Custom Intents

Parameters

“Custom” results in an `INObject`

Custom Intents

Parameters

“Custom” results in an `INObject`

- Combines an identifier with a display string

Custom Intents

Parameters

“Custom” results in an `INObject`

- Combines an identifier with a display string
- Identifier can reference an internal object

Custom Intents

Parameters

“Custom” results in an `INObject`

- Combines an identifier with a display string
- Identifier can reference an internal object
- Display string conveys that object to the user

Custom Intents

Parameters

“Custom” results in an `INObject`

- Combines an identifier with a display string
- Identifier can reference an internal object
- Display string conveys that object to the user
- Prevents possible implicit dependencies between parameters

Custom Intents

Parameters

```
public class MyIntent: INIntent {  
    public var objectIdentifier: String?  
    public var objectString: String?  
}
```

Custom Intents

Parameters

```
public class MyIntent: INIntent {  
    public var objectIdentifier: String?  
    public var objectString: String?  
}
```

```
public class MyIntent: INIntent {  
    public var object: INObject?  
}
```

Custom Intents

Parameters

```
public class MyIntent: INIntent {  
    public var objectIdentifier: String?  
    public var objectString: String?  
}
```



```
public class MyIntent: INIntent {  
    public var object: INObject?  
}
```



Shortcut Display

Titles, subtitles and images

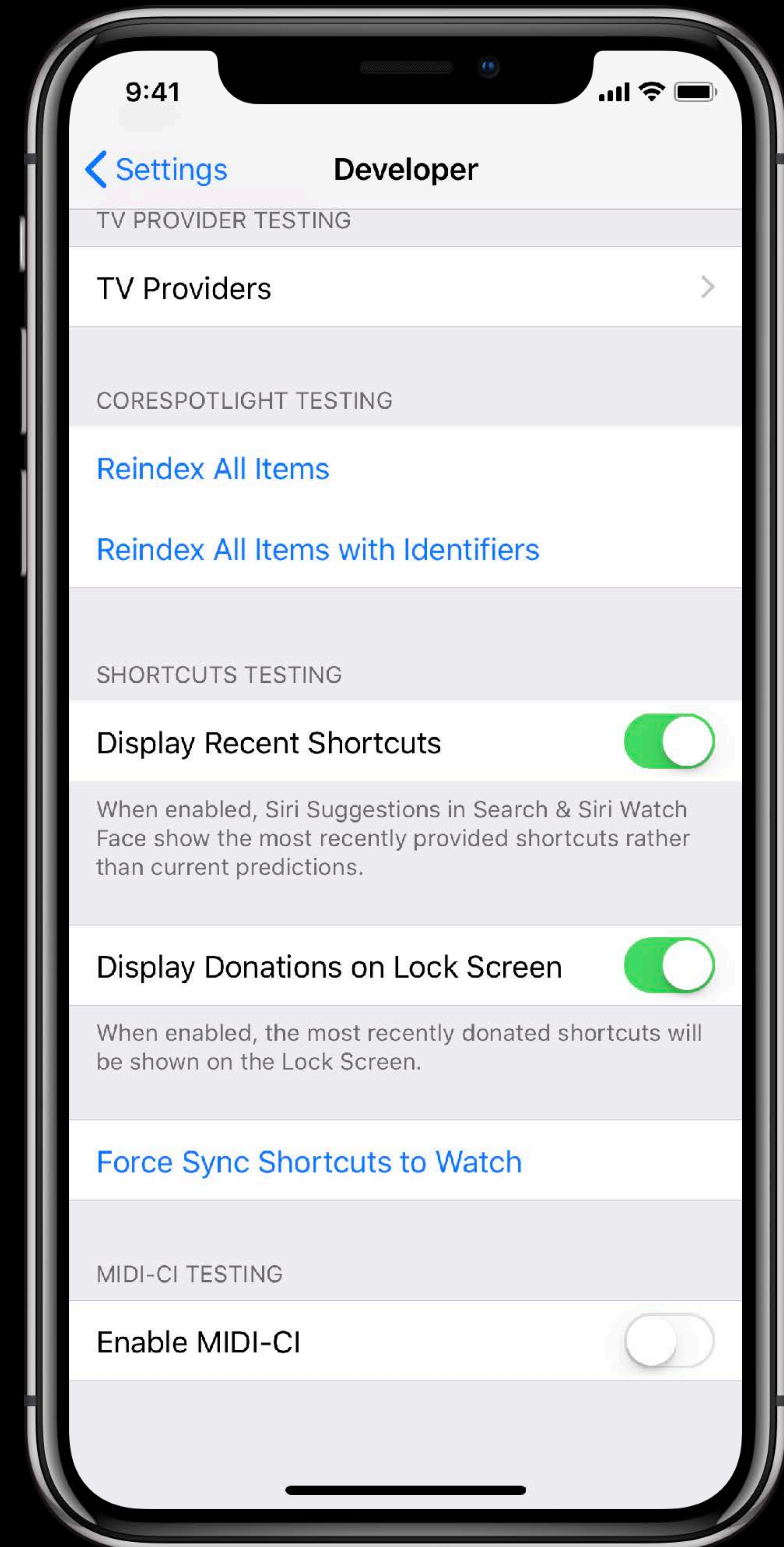
Understandable

Represent what will happen

Shortcut Testing

Shortcut Testing

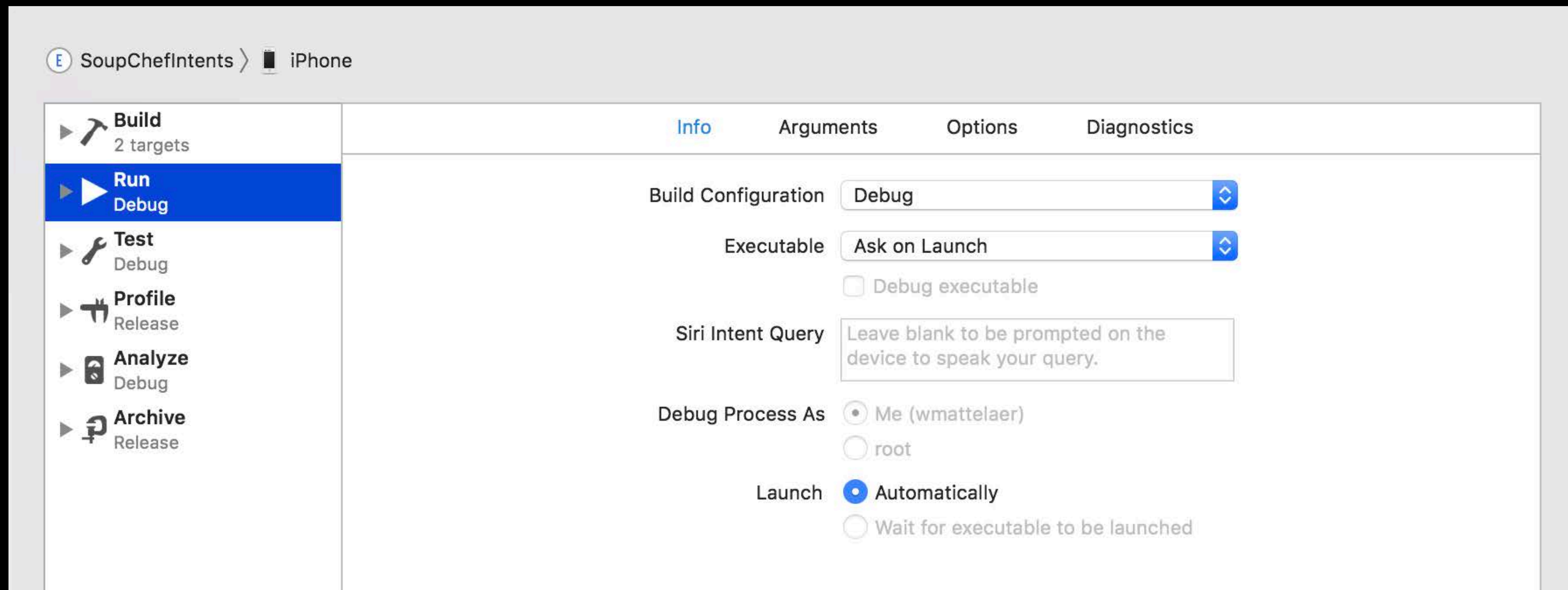
Developer settings on device



Shortcut Testing

Developer settings on device

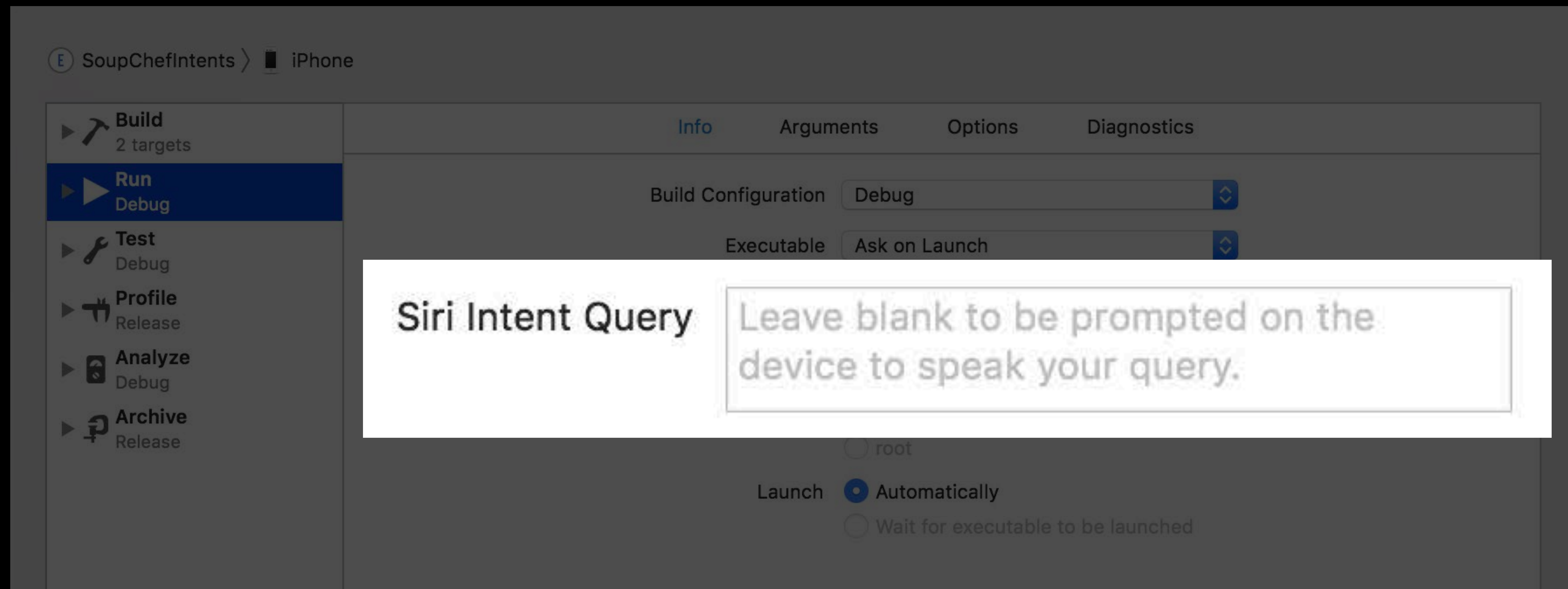
Set up a shortcut with Siri and edit the Xcode scheme to test



Shortcut Testing

Developer settings on device

Set up a shortcut with Siri and edit the Xcode scheme to test



Shortcut Testing

Developer settings on device

Set up a shortcut with Siri and edit the Xcode scheme to test

Create a custom shortcut in the Shortcuts app



Privacy

Deleting Donations

User expectations

Prevent suggesting content that's no longer relevant

Deleting Donations

NSUserActivity

Deleting user activities with corresponding Spotlight items

Deleting Donations

NSUserActivity

Deleting user activities with corresponding Spotlight items

```
userActivity.contentAttributeSet?.relatedUniqueIdentifier = "my-identifier"
```

```
let searchableIndex = CSSearchableIndex.default()  
searchableIndex.deleteSearchableItems(withIdentifiers: ["my-identifier"]) { (error) in  
}
```


Deleting Donations

NSUserActivity

Deleting user activities with corresponding Spotlight items

```
userActivity.contentAttributeSet?.relatedUniqueIdentifier = "my-identifier"
```

```
let searchableIndex = CSSearchableIndex.default()  
searchableIndex.deleteSearchableItems(withIdentifiers: ["my-identifier"]) { (error) in  
}
```

Deleting Donations

NSUserDefaults



NEW

Persistent identifier property on UserDefaults

Use the identifier to later delete specific UserDefaults



NEW

```
// Deleting NSUserActivity using persistent identifier
```

```
let userActivity = NSUserActivity(activityType: "my-activity-type")
```

```
// Configure user activity...
```

```
let persistentIdentifier = NSUserActivityPersistentIdentifier("persistent-identifier")
```

```
userActivity.persistentIdentifier = persistentIdentifier
```

```
NSUserActivity.deleteSavedUserActivities(withPersistentIdentifiers: [persistentIdentifier]) {  
}
```

```
NSUserActivity.deleteAllSavedUserActivities {  
}
```



NEW

```
// Deleting NSUserActivity using persistent identifier
```

```
let userActivity = NSUserActivity(activityType: "my-activity-type")
```

```
// Configure user activity...
```

```
let persistentIdentifier = NSUserActivityPersistentIdentifier("persistent-identifier")
```

```
userActivity.persistentIdentifier = persistentIdentifier
```

```
NSUserActivity.deleteSavedUserActivities(withPersistentIdentifiers: [persistentIdentifier]) {  
}
```

```
NSUserActivity.deleteAllSavedUserActivities {  
}
```



NEW

```
// Deleting NSUserActivity using persistent identifier
```

```
let userActivity = NSUserActivity(activityType: "my-activity-type")
```

```
// Configure user activity...
```

```
let persistentIdentifier = NSUserActivityPersistentIdentifier("persistent-identifier")
```

```
userActivity.persistentIdentifier = persistentIdentifier
```

```
NSUserActivity.deleteSavedUserActivities(withPersistentIdentifiers: [persistentIdentifier]) {  
}
```

```
NSUserActivity.deleteAllSavedUserActivities {  
}
```

Deleting Donations

Intents

Identifier and group identifier properties on INInteraction

Use the identifier or group identifier to later delete one or more INInteractions

```
// Donating
let interaction = INInteraction(intent: intent, response: response)
interaction.identifier = "my-identifier"
interaction.groupIdentifier = "my-group-identifier"
interaction.donate { (error) in
}

// Deleting the donation
INInteraction.delete(with: ["my-identifier"]) { (error) in
}

INInteraction.delete(with: "my-group-identifier") { (error) in
}

INInteraction.deleteAll { (error) in
}
```

```
// Donating
let interaction = INInteraction(intent: intent, response: response)
interaction.identifier = "my-identifier"
interaction.groupIdentifier = "my-group-identifier"
interaction.donate { (error) in
}
```

```
// Deleting the donation
```

```
INInteraction.delete(with: ["my-identifier"]) { (error) in
}
```

```
INInteraction.delete(with: "my-group-identifier") { (error) in
}
```

```
INInteraction.deleteAll { (error) in
}
```



```
// Donating
let interaction = INInteraction(intent: intent, response: response)
interaction.identifier = "my-identifier"
interaction.groupIdentifier = "my-group-identifier"
interaction.donate { (error) in
}
```

```
// Deleting the donation
INInteraction.delete(with: ["my-identifier"]) { (error) in
}
```

```
INInteraction.delete(with: "my-group-identifier") { (error) in
}
```

```
INInteraction.deleteAll { (error) in
}
```

```
// Donating
let interaction = INInteraction(intent: intent, response: response)
interaction.identifier = "my-identifier"
interaction.groupIdentifier = "my-group-identifier"
interaction.donate { (error) in
}
```

```
// Deleting the donation
INInteraction.delete(with: ["my-identifier"]) { (error) in
}
```

```
INInteraction.delete(with: "my-group-identifier") { (error) in
}
```

```
INInteraction.deleteAll { (error) in
}
```

Media Shortcuts

Play Media Intent

NEW

INPlayMediaIntent

Play Media Intent

NEW

`INPlayMediaIntent`

Launches app in background

Play Media Intent

NEW

INPlayMediaIntent

Launches app in background

Suggested in playback controls on Lock Screen

Play Media Intent

NEW

INPlayMediaIntent

Launches app in background

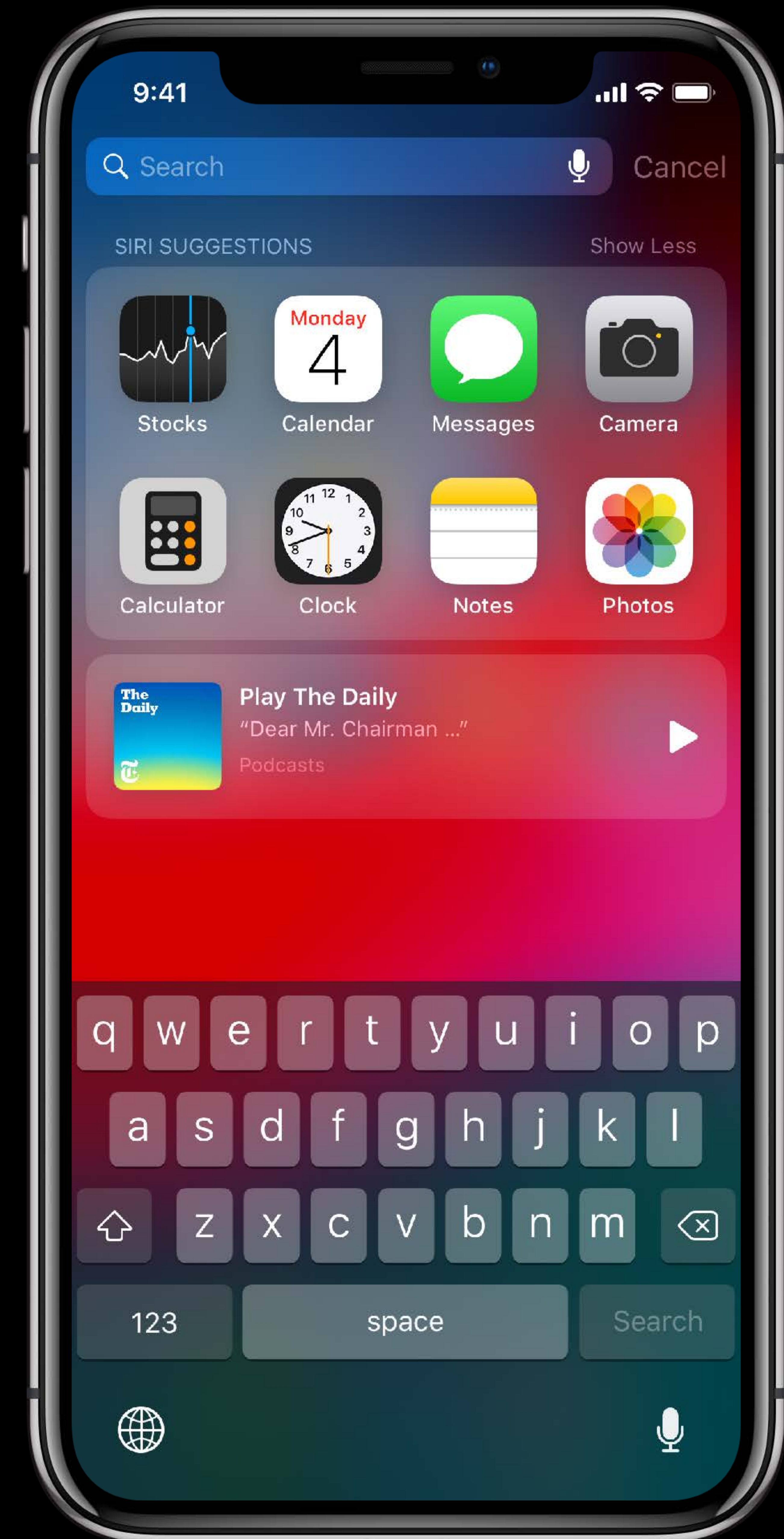
Suggested in playback controls on Lock Screen

Works on HomePod

Upcoming Media

Use `INUpcomingMedia` API to tell the system about new media the user hasn't yet consumed

Great for podcasts, TV shows



Summary

Enables powerful new experiences

Provides new ways to engage users

Adopt with `NSUserActivity` or Intents

See “Building for Voice with Siri Shortcuts”

More Information

<https://developer.apple.com/wwdc18/211>

Building for Voice with Siri Shortcuts

Hall 2

Wednesday 10:00AM

Siri Shortcuts Lab

Technology Lab 7

Wednesday 12:00PM

Siri Shortcuts Lab

Technology Lab 11

Thursday 11:00AM

 **WWDC18**