

# WWDC 2019: Hidden Amazement

Bruce Evans  
**LINE** App Dev 2

# App Store

- watchOS 6 Apps are *not* included in iOS cellular download limit
  - Older watchOS gets app thinning
- Review Guidelines
  - No 3rd party tracking in apps for Kids
  - No digital commerce in HTML games (inside apps)
  - "Sign in with Apple" *mandatory* if you support 3rd party sign in
- App Deletions can be viewed in App Store Analytics

# Dependency Injection with UIStoryboard

// Link via Segue

```
@IBSegueAction func followMySegue(coder: NSCoder) -> MyViewController {  
    return MyViewController(coder: coder, name: "Initial")  
}
```

// Instantiate via Code

```
let initialVC = storyboard?.instantiateInitialViewController(creator:  
{ coder -> UIViewController? in  
    let viewController = MyViewController(coder: coder, name: "Initial")  
    return viewController  
})  
  
let followupVC = storyboard?.instantiateViewController(identifier: "MyViewController",  
                                                         creator:  
{ coder -> UIViewController? in  
    let myController = MyViewController(coder: coder, name: "ViewController Jr.")  
    return myController  
})
```

# RelativeDateFormatter

```
let formatter = RelativeDateTimeFormatter()  
let relativeTime = formatter.localizedString(for: dayOfLegend!, relativeTo: now)  
let intervalTime = formatter.localizedString(fromTimeInterval: 300)  
let timeByComponents = formatter.localizedString(from: untilWeekend)
```

```
let famousWords = "ビッグウェーブさん was \$(relativeTime)"
```

ビッグウェーブさん was 10 years ago

```
let prediction = "My presentation will end \$(intervalTime)"
```

My presentation will end in 5 minutes

```
let betterPrediction = "The weekend is \$(timeByComponents)"
```

The weekend is in 3 days

# ListFormatter

```
ListFormatter.localizedString(byJoining: ["Sushi", "Hamburger", "Pizza"]
```

```
let formatter = ListFormatter()  
formatter.string(from: ["Sushi", "Hamburger", "Pizza"]
```

```
// Both return "Sushi, Hamburger, and Pizza"
```

## Can read Siri's prompt style

```
AVAudioSession.PromptStyle.none
```

```
AVAudioSession.PromptStyle.normal
```

```
AVAudioSession.PromptStyle.short
```

# Discretionary BG tasks

```
let config = URLSessionConfiguration.background(withIdentifier: "test.url")
config.isDiscretionary = true
config.allowsExpensiveNetworkAccess = false
let session = URLSession(configuration: config)
let downloadTask = session.downloadTask(with: url)
downloadTask.countOfBytesClientExpectsToReceive = 1_073_741_824
```

```
let processingRequest = BGProcessingTaskRequest(identifier: "test.request")
processingRequest.requiresExternalPower = true
processingRequest.requiresNetworkConnectivity = true
try BGTaskScheduler.shared.submit(processingRequest)
```

# OperationQueue.addBarrierBlock

```
let operationQueue = OperationQueue()
operationQueue.maxConcurrentOperationCount = 5
var opCount = 0
operationQueue.addOperation {
    opCount += 1
    sleep(3)
    print("Ran and slept")
}
operationQueue.addOperation {
    opCount += 1
    print("Running a second time")
}
operationQueue.addOperation {
    opCount += 1
    print("Running a third time")
}
operationQueue.addBarrierBlock { print("Ran \(opCount) operations") }
operationQueue.addOperation {
    opCount += 1
    print("Finished running \(opCount) operations")
}
```



# OperationQueue.addBarrierBlock



**Running a second time**

**Running a third time**

**Ran and slept**

**Ran 3 operations**

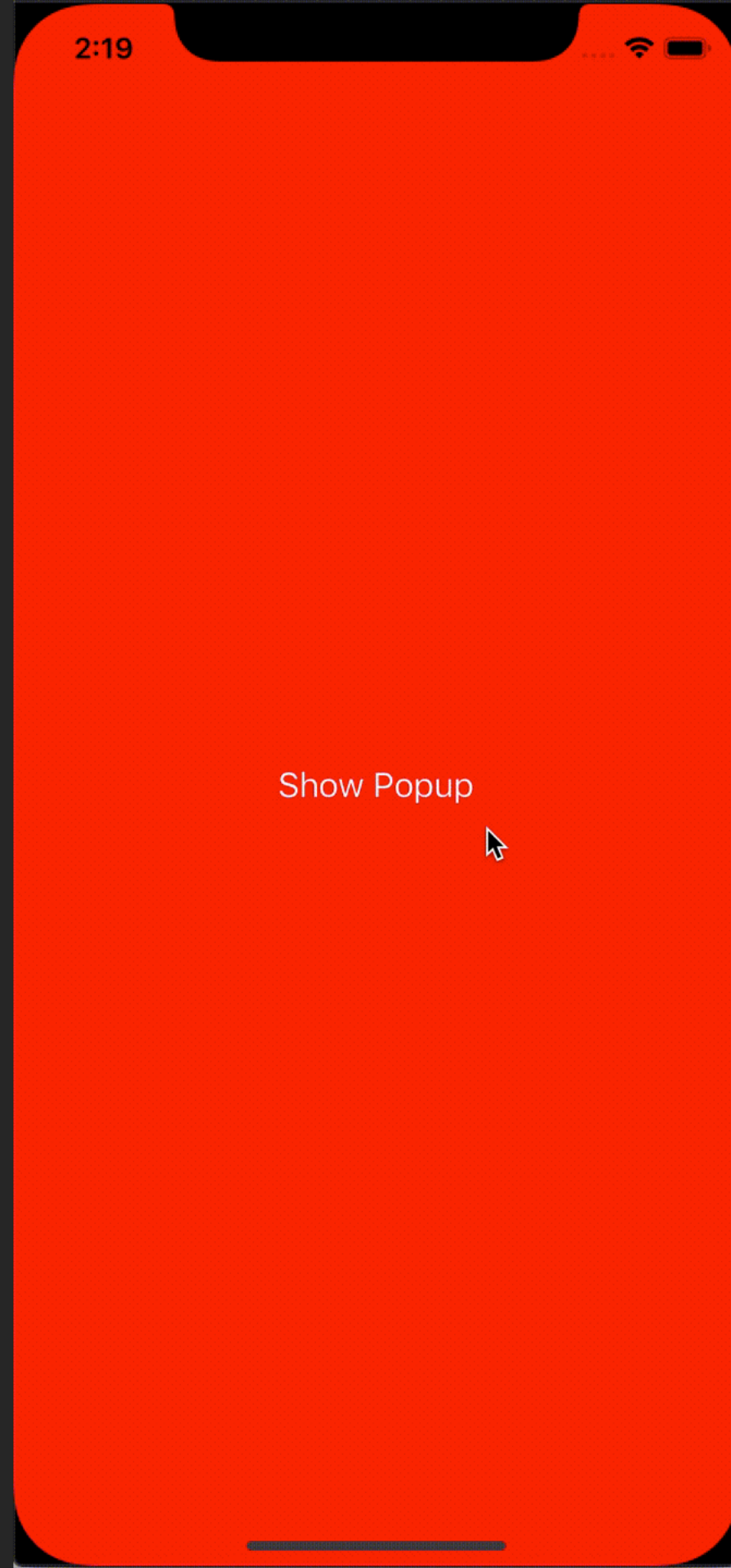
**Finished running 4 operations|**

# NSData.compress(using:)

```
let compressedData = try! data.compressed(using: .zlib)
print(data.length < compressedData.length) // true (usually)
```

```
let decompressedData: NSData = try! compressedData.decompressed(using: .zlib)
print(data.isEqual(to: decompressedData as Data)) // true
```

```
let decompressAgain = try? decompressedData.decompressed(using: .zlib) // nil
```



---

- Swipe to Dismiss

- Haptic Use

---

- MetricKit

- Low Data Mode

---

- Per-App Languages

- **git stash** in Xcode

---

- Text Catalogs

- WebSockets

---

- Collection Diffing

- Read / Write NFC

---

- Augmenting CoreML models on device

# Thanks

[https://github.com/brevansio/  
wwdc2019meetup](https://github.com/brevansio/wwdc2019meetup)

@brevansio

