

# WatchKit

# Apple Watch



# Apple Watch

- Neuartige Kategorie für Apple
- Komplett neues Klientel
- Neuartige Art der Software?

# WatchKit

# WatchKit

- Framework von Apple für Apple Watch Apps
- (Sehr) kleiner Bruder von UIKit
- Code läuft auf iPhone!

WatchKit UI-Element	Vergleichbares UIKit-Element
WKInterfaceController	UIViewController
WKUserNotificationInterfaceController	UIApplicationDelegate + UIAlertController
WKInterfaceDevice	UIDevice
WKInterfaceObject	UIView
WKInterfaceButton	UIButton
WKInterfaceDate	UILabel + NSDateFormatter
WKInterfaceGroup	UIScrollView
WKInterfaceImage	UIImageView
WKInterfaceLabel	UILabel
WKInterfaceMap	MKMapView
WKInterfaceSeparator	UITableView.separatorColor / .separatorStyle
WKInterfaceSlider	UIStepper + UISlider
WKInterfaceSwitch	UISwitch
WKInterfaceTable	UITableView
© Benjamin Herzog, 2015 WKInterfaceTimer	UILabel + NSDateFormatter + NSTimer

## Apple Watch

### WatchKit App

Storyboard

Resources

WatchKit



## iPhone

### WatchKit Extension

Code

Resources

WatchKit





# WatchKit

- 2 Targets
  - Extension
  - App
- Möglichkeit der Kommunikation mit iPhone App
- App ist immer Teil einer iPhone App

# WatchKit

- Apple Watch App
- Glances
- Notifications

# Apple Watch App

- Wird vom Homescreen bewusst geöffnet
- Relativ freie Funktionsgestaltung
- Zugriff auf alle WKInterface-Elemente
- Mehrere Controller

# Glances

- Nur ein einziger Controller
- Keine Action-Elemente (Buttons, Tables etc)
- Kein Scrolling
- Optional

# Notifications

- Werden für alle Notifications automatisch erstellt
- Eigene Controller können Extra-Content anzeigen
- Actions müssen im Payload der Notification stehen

# WKInterfaceController

```
class InterfaceController: WKInterfaceController {  
    override func awakeWithContext(context: AnyObject?) { ... }  
  
    override func willActivate() { ... }  
  
    override func didDeactivate() { ... }  
  
    override func table(table: WKInterfaceTable,  
        didSelectRowAtIndex rowIndex: Int) { ... }  
}
```

# Interface Elemente



**Das gesamte  
Interface steht  
zur Design-Zeit  
fest.**



# Kommunikation mit iPhone-App

- Watch

```
var userInfo = ["request": "doSomething"]
WKInterfaceController.openParentApplication(userInfo) {
    reply, error in
}
```

- iPhone

```
func application(application: UIApplication!,
    handleWatchKitExtensionRequest userInfo: [NSObject : AnyObject]!,
    reply: (([NSObject : AnyObject]!) -> Void)!) {
}
```

# Handoff!

# Handoff

- Watch

```
var userActivity = "Zeige Item in einer Liste"
var userInfo = ["item": currentItem]
updateUserActivity(userActivity, userInfo: userInfo,
                   webpageURL: nil)
```

- iPhone

```
func application(application: UIApplication,
                 willContinueUserActivityWithType userActivityType: String) -> Bool { ... }

func application(application: UIApplication,
                 continueUserActivity userActivity: NSUserActivity,
                 restorationHandler: ([AnyObject]!) -> Void) -> Bool { ... }
```

# Weitere Besonderheiten

- Animationen funktionieren, aber nur über Einzelframes
- Länger dauernde Requests wie GPS oder Netzwerk immer über iPhone-App machen
- Extension ist Hintergrundtask auf iPhone
- Watch hat internen Speicher. 5MB persistenter Cache pro App

# Xcode