

WatchKit

Apple Watch



Apple Watch

- Neue Gerätekategorie für Apple
- Neue Art der Kommunikation
- Neue mobile Apps?!

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
WKInterfaceTimer	UILabel + NSDateFormatter + NSTimer

Apple Watch

WatchKit App

Storyboard

Resources

WatchKit



iPhone

WatchKit Extension

Code

Resources

WatchKit



WatchKit

- 2 Bestandteile
 - Storyboards und Assets
 - WatchKit-Extension
- Immer nur ein Bestandteil einer iOS-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

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!

Besonderheiten in der Implementation

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

Besonderheiten in der Implementation

- Sehr kurze App-Nutzungsdauer
- Daten können über App Group geteilt werden.
- Code kann über embedded Frameworks geteilt werden
- Im Asset Katalog können unterschiedliche Bilder für beide Modelle angegeben werden

In welche Richtung sollten Watch Apps gehen?

- Eher eine Art Ergänzung zur iPhone App
- Unterstützung für Notifications
- Schnelle, kurze und relevante Infos

Zusammenfassung

Zusammenfassung

- Watch ist mehr eine Art "verlängertes" Display
- Kein Zugriff auf Sensoren
- Kein Mikrofon- oder Lautsprecherzugriff
- Hoffen auf v2 und native SDK :)

Diskussion