

State Preservation & Restoration on iOS

Sebastian Hagedorn
CocoaHeads Dresden
09.07.2014

Übersicht

☞ Motivation

☞ Implementierung

☞ Debugging & GOTCHAs

☞ Quellen & Links

Motivation

- ☞ iOS beendet im Hintergrund laufende Apps (aus Nutzersicht) wahllos
- ☞ State Restoration: Verbergen, dass App beendet wurde
- ☞ Ziel: Kein Unterschied zwischen Rückkehr zu laufender App und Neustart einer beendeten App
- ☞ Vollständige Wiederherstellung des alten Zustands nicht immer sinnvoll

Implementierung

☞ Vorgehensweise:

☞ State Restoration aktivieren

☞ ViewController(hierarchie) wiederherstellen

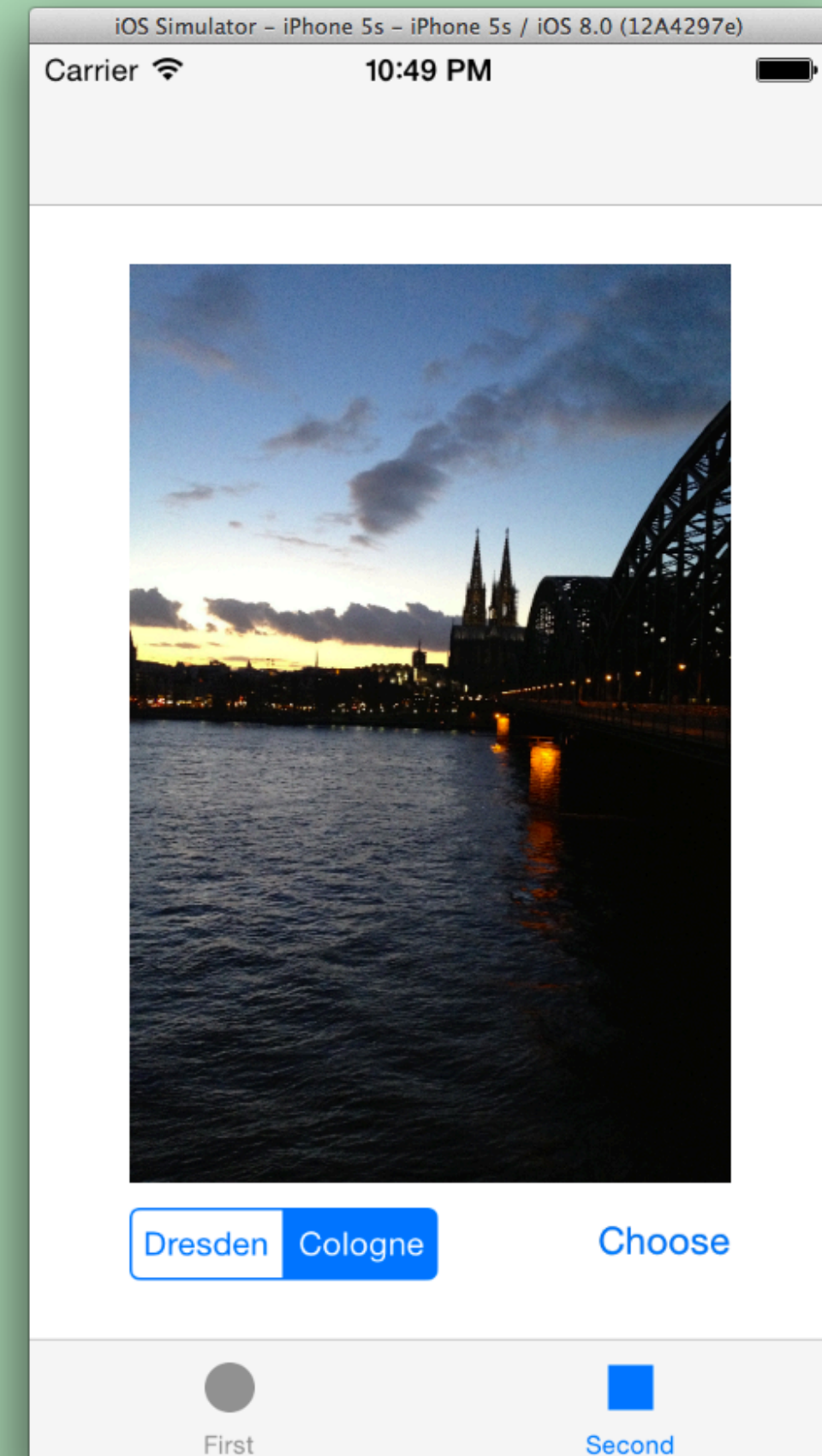
☞ Status/Daten wiederherstellen

Implementierung

☞ Beispielprojekt:

[github.com/shagedorn/
StateRestorationDemo](https://github.com/shagedorn/StateRestorationDemo)

☞ Tag Ausgangslage:
`NO_STATE_RESTORETION`



Implementierung: Aktivierung

☞ Opt-In im App Delegate:

- `(BOOL)application:(UIApplication *)application shouldSaveApplicationState:(NSCoder *)coder;`
- `(BOOL)application:(UIApplication *)application shouldRestoreApplicationState:(NSCoder *)coder;`

☞ Restoration-Datei wird angelegt, enthält globale App-Infos
(Version, Timestamp, Interface Idiom,...)

☞ `return NO` nach inkompatiblen Updates oder großer
Zeitspanne

Implementierung: Aktivierung

- System nimmt Snapshot bevor App in den Hintergrund geht
 - Ersetzt `Default.png`, falls mindestens ein ViewController State Preservation implementiert hat
- Tag: `STATE_RESTORETION_OPT_IN`

Implementierung: Controller

- ☞ Per-Controller Opt-In: `.restorationIdentifier`-Property setzen
 - ☞ IB/Storyboard oder Code
- ☞ Geschafft: State Preservation
 - ☞ Gespeichert wird ein Pfad von Restoration-Identifiers
- ☞ TODO: State Restoration
- ☞ Tag: `RESTORATION_IDENTIFIERS`

Implementierung: Controller

☞ Wiederherstellung von ViewControllern (Optionen):

- ① Controller setzen `.restorationClass`
- ② AppDelegate instanziert Controller auf Anfrage
- ③ Implizit: Controller wurden zum Zeitpunkt der State Restoration bereits erstellt
- ④ Implizit: Controller befinden sich im Storyboard

Implementierung: Controller

☞ Wiederherstellung von ViewControllern:

☞ AppDelegate instanziiert Controller auf Anfrage

```
– (UIViewController *)application:(UIApplication *)application  
  viewControllerWithRestorationIdentifierPath:(NSArray *)identifierComponents  
  coder:(NSCoder *)coder;
```

☞ `return nil`: Implizite Suche wird fortgesetzt

Implementierung: Controller

☞ Wiederherstellung von ViewControllern:

☞ Implizit: Controller wurden zum Zeitpunkt der State Restoration bereits erstellt

☞ Reihenfolge:

① `application:willFinishLaunchingWithOptions:`

② `[State Restoration]`

③ `application:didFinishLaunchingWithOptions:`

Implementierung: Controller

☞ Wiederherstellung von ViewControllern:

☞ Status der Demo-App bei Tag `BASIC_STATE_RESTORATION`:

☞ Initiale ViewController (Tabs) werden implizit hergestellt

☞ Später erstellte ViewController nicht

☞ Zustand der Controller/Views wird nicht hergestellt

Implementierung: Controller

👉 Wiederherstellung von ViewControllern:

➡ Controller setzen `.restorationClass`

- ☞ z.B. der Controller selbst: `UIViewControllerRestoration-Protokoll` implementieren

[illegible]

Implementierung: Controller

- ☞ Wiederherstellung von ViewControllern:
 - ☞ Controller definieren `.restorationClass`
 - ☞ Opt-Out durch `nil`
 - ☞ Verhindert implizite Wiederherstellung aus Storyboards
 - ☞ Zugriff auf Coder: Entscheidung über erfolgreiche Restoration möglich
 - ☞ Tag: `RESTORATION_CLASS`

Implementierung: State

☞ UIStateRestoring-Protokoll:

```
- (void)encodeRestorableStateWithCoder:(NSCoder *)coder {  
    [super encodeRestorableStateWithCoder:coder];  
    [coder encodeFloat:self.value forKey:@"encodingKeyValue"];  
}  
  
- (void)decodeRestorableStateWithCoder:(NSCoder *)coder {  
    ... // happens after viewDidLoad  
}
```

☞ Primitive Werte und andere Objekte, die selbst State Restoration implementieren

Implementierung: State

- ☞ `decodeRestorableStateWithCoder:` wird nach `viewDidLoad` aufgerufen
- ☞ Restoration kann nicht mehr abgebrochen werden
- ☞ Views können selbst State Restoration implementieren
 - ☞ funktioniert nur beschränkt automatisch
- ☞ Tag: `STATE_ENCODING`

Debugging

- ☞ Auslöser für State Preservation: App geht in Hintergrund
 - ☞ nicht in Xcode oder App Switcher killen
- ☞ Löschen von State Restoration-Daten:
 - ☞ Restoration schlägt fehl/App crasht
 - ☞ Nutzer beendet App
 - ☞ nicht im Simulator
 - ☞ kann mit iOS Profil umgangen werden

Debugging

- ☞ Infos werden in `data.data`-Dateien abgelegt
- ☞ Binary PLISTS: Auslesen mit `plutil`-Tool oder `restorationArchiveTool`
- ☞ <https://developer.apple.com/downloads/> → Suche nach "restoration"
- ☞ `restorationArchiveTool`
- ☞ `StateRestoration DebugLogging` Profil
- ☞ `StateRestoration DeveloperMode` Profil

Debugging

☞ Tipp: data.data-Suche im Simulator-Ordner in Finder-Sidebar speichern

☞ Aufruf: `restorationArchiveTool path/to/data.data`

```
UIApplicationStateRestorationSystemVersion.....8.0
UIApplicationStateRestorationTimestamp.....2014-07-06 16:05:18 +0000
UIApplicationStateRestorationUserInterfaceIdiom...iPhone

[View Controller] TabController (Class: UITabBarController)
-----

kTabBarSelectedViewControllerKey...Object Identifier Proxy: TabController/:[1]:/NavigationController
kViewControllerViewWasLoadedKey....Yes

[View Controller] TabController/:[0]:/_TtC20StateRestorationDemo19FirstViewController (Class: _TtC20StateRestorationDemo19FirstViewController)
-----

encodingKeySliderValue.....0.8926057
kViewControllerViewWasLoadedKey...Yes
```

Debugging

👉 DebugLogging Profil: Sehr verbose, inkl. Zeitmessung

👉 Warning: Unable to create restoration in progress marker file

👉 Keine data.data-Datei gefunden

```
2014-07-07 22:24:50.412 StateRestorationDemo[1321:60b] void _restoreState(UIApplication
*, NSData *, NSObject<UIApplicationDelegate> *, NSURL *, NSString *,
UIStateRestorationRestoreStateBeginHandler): State restoration archive was saved with
major version 2, minor version 1. Current major version 2, current minor version 1.
2014-07-07 22:24:50.417 StateRestorationDemo[1321:60b] void _restoreState(UIApplication
*, NSData *, NSObject<UIApplicationDelegate> *, NSURL *, NSString *,
UIStateRestorationRestoreStateBeginHandler): Root restoration identifier paths are (
    TabController,
    "TabController/:[0]:/_TtC20StateRestorationDemo19FirstViewController",
    "TabController/:[1]:/NavigationController",
    "TabController/:[1]:/NavigationController/:[0]:/
_TtC20StateRestorationDemo20SecondViewController",
    "TabController/:[1]:/NavigationController/:[1]:/
_TtC20StateRestorationDemo18CityViewController"
)
2014-07-07 22:24:50.420 StateRestorationDemo[1321:60b] void _restoreState(UIApplication
*, NSData *, NSObject<UIApplicationDelegate> *, NSURL *, NSString *,
UIStateRestorationRestoreStateBeginHandler): Restoration Class map: {
    "TabController/:[1]:/NavigationController/:[1]:/
_TtC20StateRestorationDemo18CityViewController" =
    "_TtC20StateRestorationDemo18CityViewController";
}
2014-07-07 22:24:50.422 StateRestorationDemo[1321:60b] void _restoreState(UIApplication
*, NSData *, NSObject<UIApplicationDelegate> *, NSURL *, NSString *,
UIStateRestorationRestoreStateBeginHandler): Object in root set for index [0] for
identifier path TabController: <UITabBarController: 0x1455cb50>
2014-07-07 22:24:50.425 StateRestorationDemo[1321:60b] void _restoreState(UIApplication
*, NSData *, NSObject<UIApplicationDelegate> *, NSURL *, NSString *,
```

Debugging

- ☞ Can't find Child View Controller at index 1 with identifier TabController/[1]:/
NavigationController/[1]:/
_TtC20StateRestorationDemo18CityViewController,
truncating child array
- ☞ Controller nicht gefunden

GOTCHAs/Tipps

- ☞ `.restorationIdentifier` durch Überschreiben des Getters setzen
- ☞ Ermöglicht State Preservation: `data.data` ist ok
- ☞ Keine Restoration: Registrierung im Setter?
- ☞ `.restorationIdentifier` in `init` setzen:
- ☞ Encapsulation vs. Wiederverwendbarkeit

GOTCHAs/Tipps

- ☞ Nicht alle Subviews werden von Snapshots erfasst
 - ☞ Bsp.: AlertViews
- ☞ Konsistenz zwischen Snapshots und App anstreben
- ☞ Keine temporären Fehlermeldungen wiederherstellen
- ☞ Snapshot ignorieren:

```
[UIApplication sharedApplication] ignoreSnapshotOnNextApplicationLaunch];
```

Tipps

- ☞ `.window` muss nicht explizit gesichert werden, aber Preservation des Windows fügt unter iOS 8 Informationen über Size Classes hinzu
- ☞ Tag: `SIZE_CLASSES`

Quellen & Links

- Doku: <https://developer.apple.com/library/ios/documentation/iphone/conceptual/iphonesprogrammingguide/StatePreservation/StatePreservation.html>
- Ausführlich, aber nicht ganz aktuell
- grundsätzlich ab iOS 6, Ergänzungen (fehlen hier) in iOS 7
- WWDC 2013 Session 222: What's New in State Restoration
- Ausführlich & aktuell