

HANSER



Erratum

Das Swift-Handbuch

von Thomas Sillmann

Print-ISBN: 978-3-446-45505-4

E-Book-ISBN: 978-3-446-45730-0

E-Pub-ISBN: 978-3-446-46107-9

© Carl Hanser Verlag München

Errata zu „Das Swift-Handbuch“

Hier finden Sie alle Korrekturen zu den Inhalten des Swift-Handbuchs. Die Errata wurde zum letzten Mal am 27.06.2019 aktualisiert.

■ Abschnitt 21.6.2: Einblenden neuer View-Controller

Seite 457

Die drei aufgeführten Methoden zum Einblenden neuer View-Controller tragen eine andere Bezeichnung, siehe die hier eingefügte Tabelle 21.1.

Tabelle 21.1 Neue Methodennamen zum Einblenden von View-Controllern

Methodenname im Buch	Aktuelle Version
<code>presentViewControllerAsModalWindow(_:)</code>	<code>presentAsModalWindow(_:)</code>
<code>presentViewControllerAsSheet(_:)</code>	<code>presentAsSheet(_:)</code>
<code>presentViewController(_:asPopoverRelativeTo:of:preferredEdge:behavior:)</code>	<code>present(_:asPopoverRelativeTo:of:preferredEdge:behavior:)</code>

Seite 463

Auch die Handhabung der auf Seite 463 beschriebenen IDs hat sich minimal geändert. Die beschriebenen Typen `NSStoryboard.Name` sowie `NSStoryboard.SceneIdentifier` sind einfache Typ-Aliase für den Typ `String`. So genügt es, direkt die gewünschte ID zu übergeben, ohne diese in einen der beschriebenen Typen zu packen. Das führt zu den in der hier eingefügten Tabelle 21.2 aufgeführten Code-Änderungen.

Tabelle 21.2 Erstellung von Storyboard-IDs

Bisheriges Verfahren im Buch	Aktuelle Version
<code>NSStoryboard.Name(rawValue: "Main")</code>	<code>"Main"</code>
<code>NSStoryboard.SceneIdentifier(rawValue: "Destination")</code>	<code>"Destination"</code>

Aufgrund dieser Änderungen finden Sie im Folgenden noch die aktualisierte Version von Listing 21.6 und Listing 21.7.

Listing 21.6 Erzeugen der `destinationViewController`-Property (aktualisiert)

```
private let destinationViewController: NSViewController = {
    let mainStoryboard = UIStoryboard(name: "Main", bundle: nil)
    let destinationViewController = mainStoryboard.instantiateController(with
Identifier: "Destination") as! NSViewController
    return destinationViewController
}()
```

Listing 21.7 Vollständige Implementierung der `ViewController`-Klasse zum Einblenden eines neuen View-Controllers (aktualisiert)

```
class ViewController: NSViewController {

    @IBOutlet weak var popoverButton: NSButton!

    private let destinationViewController: NSViewController = {
        let mainStoryboard = UIStoryboard(name: "Main", bundle: nil)
        let destinationViewController = mainStoryboard.instantiateController(with
Identifier: "Destination") as! NSViewController
        return destinationViewController
    }()

    @IBAction func presentModalWindow(_ sender: Any) {
        presentAsModalWindow(destinationViewController)
    }

    @IBAction func presentSheet(_ sender: Any) {
        presentAsSheet(destinationViewController)
    }

    @IBAction func presentPopover(_ sender: Any) {
        let popoverSize = NSRect(x: 0, y: 0, width: 300, height: 300)
        present(destinationViewController, asPopoverRelativeTo: popoverSize, of:
popoverButton, preferredEdge: .maxX, behavior: .semitransient)
    }
}
```

■ Abschnitt 21.6.3: Ausblenden eines View-Controllers

Seite 467

Die Methode zum Ausblenden eines View-Controllers heißt `dismiss(_:)`, nicht `dismissViewController(_:)`. Im Folgenden finden Sie daher zusätzlich auch die aktualisierte Version von Listing 21.8.

Listing 21.8 Ausblenden eines View-Controllers bei Betätigen eines Buttons (aktualisiert)

```
class DestinationViewController: NSViewController {  
  
    @IBAction func dismiss(_ sender: Any) {  
        dismiss(self)  
    }  
  
}
```

■ Abschnitt 24.4.3: Auf Ein- und Ausblenden des Keyboards reagieren

Seite 812

Die Namen der beschriebenen Notifications, die über das Ein- und Ausblenden der Bildschirmstatur informieren, haben sich wie folgt geändert:

- `UIKeyboardWillShow` entspricht jetzt `UIResponder.keyboardWillShowNotification`
- `UIKeyboardDidShow` entspricht jetzt `UIResponder.keyboardDidShowNotification`
- `UIKeyboardWillHide` entspricht jetzt `UIResponder.keyboardWillHideNotification`
- `UIKeyboardDidHide` entspricht jetzt `UIResponder.keyboardDidHideNotification`

Darüber hinaus lautet der Schlüssel zum Zugriff auf den Keyboard-Frame nicht länger `UIKeyboardFrameEndUserInfoKey`, sondern `UIResponder.keyboardFrameEndUserInfoKey`.

Seite 814

Nachfolgend sehen Sie die entsprechend aktualisierte Version von Listing 24.36:

Listing 24.36 Reaktion auf Ein- und Ausblenden der virtuellen Bildschirmstatur (aktualisiert)

```
class ViewController: UIViewController {  
  
    @IBOutlet weak var textView: UITextView!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

```
NotificationCenter.default.addObserver(self, selector:#selector(keyboardDidShow(_:)), name: UIResponder.keyboardDidShowNotification, object: nil)
NotificationCenter.default.addObserver(self, selector:
#selector(keyboardDidHide(_:)), name: UIResponder.keyboardDidHideNotification, object:
nil)
}

@IBAction func dismissKeyboard() {
    textView.resignFirstResponder()
}

@objc private func keyboardDidShow(_ notification: Notification) {
    let endFrame = notification.userInfo![UIResponder.keyboardFrameEndUserInfoKey]
as! CGRect
    textView.contentInset.bottom = endFrame.size.height
    textView.scrollIndicatorInsets = textView.contentInset
}

@objc private func keyboardDidHide(_ notification: Notification) {
    textView.contentInset.bottom = 0
    textView.scrollIndicatorInsets = textView.contentInset
}

deinit {
    NotificationCenter.default.removeObserver(self, name: UIResponder.
keyboardDidShowNotification, object: nil)
    NotificationCenter.default.removeObserver(self, name: UIResponder.
keyboardDidHideNotification, object: nil)
}
}
```