PDFKit Basics

Mitch Cohen
mitch@mitchcohen.com
Twitter: @mitchcohen
Micro.Blog: @mitch

Mitch and the Memory Leakers

The Second Greatest Apple Developer Rock n Roll Band!

Ticket For

This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event.

Ticket Holder's Name

Unique QR Code

Unique Link to Band Web Site

PDF Basics

- *The Portable Document Format (PDF) (redundantly: PDF format) is a file format developed by Adobe in the 1990s to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems*. (Wikipedia)
- Current version’s spec is 971 pages
- Page layout language, originally for printed pages, extended with some modern stuff
- Based on Postscript, which is itself is very useful despite being a horrific mess. PDF is both more useful and more horrific.

PDF Layout Basics

- Coordinate system has origin at the lower-left
- Text
- Vector shapes
- Bitmap images
- Annotations
  - User-editable text
  - Text/Images with actions

PDFKit Basics

- On iOS and macOS; not on watchOS or tvOS
  - Added in iOS 9, but mostly useless until iOS 11
- PDFView - A UIView that displays a PDF
- PDFDocument - the PDF displayed in the PDFView
- PDFPage - A single page of the PDFDocument
- PDFAnnotation - A single annotation (actionable element) on a PDFPage
- PDFAction - The optional action of a PDFAnnotation

PDFDocument

- Create from URL, Data/NSData, or from nothing at all…

```
let url = Bundle.main.url(forResource: "someFile",
    withExtension: "pdf")
let document = PDFDocument.init(url: url!)
```

PDFView

- Initialize the view, then add the document

```
let view = PDFView.init(frame: self.view.frame)
view.document = document
```

```
view.autoScales = true
```

PDFView

- Multiple presentation options
  - Single Page
  - Single Page Continuous
  - Two-Up
  - Two-Up Continuous

Don’t Do This Any More!

```
let data = (Data of a PDF file)
let webView = WKWebView()…
webView.load(data,
             mimeType: "application/pdf",
             characterEncodingName: “utf-8",
             baseURL: NSURL.init(string: ""))
```

Do This Instead

```
let url = Bundle.main.url(forResource: “myFile",
                                  withExtension: "pdf")
let document = PDFDocument.init(url: url!)
let pdfView = PDFView(frame: self.view.frame)
pdfView?.document = pdfDocumentFromFile()
pdfView?.autoScales = true
self.view.addSubview(pdfView!)
```

PDFView Points

- It’s all 72 points per inch, regardless of display size
- A letter-size PDF is 612 x 792 points
- Getting the page bounds:

```
let displayBox = self.myPDFView.displayBox
let firstPage = self.myPDFView.document?.page(at: 0)
let pageBounds = firstPage?.bounds(for: displayBox)
print("pageBounds: \(pageBounds)”)
   pageBounds: (0.0, 0.0, 612.0, 792.0)
```

PDFAnnotation

- Add a text annotation

```
let frame = CGRect(x: 320, y: 600, width: 200, height: 80)
let annotation = PDFAnnotation.init(bounds: frame,
                    forType: .freeText, withProperties: nil)
annotation.contents = “John Wilker”
annotation.fontColor = UIColor.red
annotation.font = UIFont.systemFont(ofSize: 36)
annotation.color = UIColor.clear

let firstPage = self.myPDFView.document?.page(at: 0)
firstPage?.addAnnotation(annotation)
```

PDFActionURL

- Add an Action - clickable/tappable annotation

```
let url = URL.init(string: "http://www.cocoaheadsboston.org")
let action = PDFActionURL(url: url!)
annotation.action = action
```

PDFAnnotation with UIImage

```
public class ImageAnnotation: PDFAnnotation {
    private var _image: UIImage?
    public init(imageBounds: CGRect, image: UIImage?) {
        self._image = image
        super.init(bounds: imageBounds, forType: .stamp, withProperties: nil)
    }
    required public init?(coder aDecoder: NSCoder) {
        fatalError("init(coder:) has not been implemented")
    }
    override public func draw(with box: PDFDisplayBox, in context: CGContext) {
        guard let cgImage = self._image?.cgImage else {
            return
        }
        let drawingBox = self.page?.bounds(for: box)
        context.draw(cgImage, in: self.bounds.applying(CGAffineTransform(
            translationX: (drawingBox?.origin.x)! * -1.0,
            y: (drawingBox?.origin.y)! * -1.0)))
    }
}

let annotation = ImageAnnotation(imageBounds: frame, image: image)
```

PDFPage

```
PDFView.document?.page(at: 0)
```

```
PDFView.document?.currentPage
```

```
PDFView.page(for: CGPoint, nearest: true)
```

```
let page = PDFPage.init(image: UIImage)
```

Finding Annotations

- PDFView.annotationForPoint(CGPoint)
- Annotations are rectangles, even when they don’t look like them!
    - Circles, bezier curves

Save the File

```
PDFDocument.write(to: path)

Or if you just want the Data:
PDFDocument.dataRepresentation()
```

Get an Image of a page

```
let box = self.myPDFView.displayBox
Let image = self.myPDFView.currentPage?.thumbnail(of:
CGSize, for: box)

Also
draw(with: PDFDisplayBox, to: CGContext)
```

Ticket For

John Wilker

This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event. This is a fake ticket for some amazing event.

Click for Mitch!

Ticket Holder's Name

Unique QR Code

Unique Link to Band Web Site

Thank You!

Quick Demo

Presentation and Sample Code

github.com/mitchcohen/360iDev2019-PDFKit