

« μ Nav»

Fabrice.Kordon@lip6.fr



As an introduction...

- 📱 **Implement a very basic web browser...**
 - 🔗 Just to play with WKWebKit...



Demo (4" device)



Demo (5.8" device)



Demo (9.7" device)



Embedded web files



In the project

The screenshot shows the Xcode IDE interface. The top status bar indicates the project name 'µNav', the device 'iPadPro9 SAR', and the build status 'Build Succeeded' at 18:14. The breadcrumb navigation shows the path: µNav > µNav > css.css > body. The left sidebar displays the project's file structure, with 'css.css' selected. The main editor area shows the following CSS code:

```
1 body {
2     font-family:helvetica;
3     font-size: 20px;
4     background-color: #ffffff;
5     color: #000000;
6 }
7 p {
8     text-align: justify;
9     margin-top: 3px;
10    margin-right: 3px;
11 }
12 h1 {
13     width: 500px;
14     font-size: 25px;
15     font-weight: bold;
16     line-height: 35px;
17     margin-top: 16px;
18     margin-right: 3px;
19     color: #003399;
20     border-bottom-width: 2px;
21     border-bottom-style: solid;
22     border-bottom-color: #003399;
23 }
24
25
```

Embedded web files

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <link href="css.css" rel="stylesheet" type="text/css" media="screen, print" />
  <title>Local page 1</title>
</head>
<body>
  <h1>Introduction</h1>
  <p>This is a page with some formatted text (HTML + CSS).</p>
  <p>This page is located in the project and embedded in the App bundle, so tis
technique can be used to establish some in-app documentation
  with a nice looking...</p>
  <p>I can even put <a href="https://lip6.fr">some link.</a></p>
  <p>And here is another <a href="https://lip6.fr/Fabrice.Kordon/5I452-2018/
semaine-01-EN.php">link to a redirected URL.</a></p>
  <p>&nbsp;</p>
  <p>Let us try to reach a <a href="https://toto.content.fr">wrong URL : https://
toto.content.fr</a></p>
</body>
</html>
```

Embedded web files

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <link href="css.css" rel="stylesheet" type="text/css" media="screen,print"/>
  <title>Local page 1</title>
</head>
<body>
  <h1>Another introduction</h1>
  <p>This is <b>another</b> text.</p>
  <p>This page is also part of the Xcode project and embedded in the bundle
App...</p>
  <p>&nbsp;</p>
  <a href="#" onClick="alert('Thanks for the click!')">Click me for a JavaScript
window popup</a>
</body>
</html>
```


ViewController

```
// ViewController.swift
// µNav
//
// Created by Fabrice Kordon on 02/10/2018.
// Copyright © 2018 Sorbonne Université. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view
        let v = MyView(frame: UIScreen.main.bounds)
        self.view = v
    }

    override func viewWillTransition(to size: CGSize,
                                     with coordinator:
                                         UINavigationControllerTransitionCoordinator) {
        super.viewWillTransition(to: size, with: coordinator)
        let v = self.view as! MyView
        v.drawInFormat(format:size)
    }
}
```

MyView

```
import UIKit
import WebKit

class MyView: UIView, WKNavigationDelegate {

    private var wv : WKWebView!
    private let b1 = UIButton(type: .system)
    private let b2 = UIButton(type: .system)
    private let l = UILabel()
```

MyView

```
override init(frame: CGRect) {
    b1.setTitle("Page 1", for: .normal)
    b2.setTitle("Back", for: .normal)
    l.font = UIFont.systemFont(ofSize: 12.0)
    l.textColor = .red
    l.textAlignment = .center
    // handling the web part
    wv = WKWebView(frame: .zero,
                   configuration: WKWebViewConfiguration())
    super.init(frame: frame) // then we will need self
    wv.navigationDelegate = self
    self.addSubview(wv)
    b1.addTarget(self, action: #selector(goThere), for: .touchDown)
    b2.addTarget(self, action: #selector(goBack), for: .touchDown)
    self.backgroundColor = UIColor(red:0.7, green:0.8, blue:0.9, alpha:1)
    self.addSubview(wv)
    self.addSubview(b1)
    self.addSubview(b2)
    self.addSubview(l)
    // Loading an URL
    let myURL = URL(string:"https://sorbonne-universite.fr")
    let myRequest = URLRequest(url: myURL!)
    wv.load(myRequest)
    // set-up the display
    self.drawInFormat(format: frame.size)
}
```

MyView

```
required init?(coder aDecoder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
}

@objc func goThere () {
    let crtURL : URL?
    if b1.titleLabel?.text == "Page 1" {
        crtURL = URL(fileURLWithPath:
            Bundle.main.path(forResource: "page-1", ofType: "html")!)
        b1.setTitle("Page 2", for: .normal)
    } else {
        crtURL = URL(fileURLWithPath:
            Bundle.main.path(forResource: "page-2", ofType: "html")!)
        b1.setTitle("Page 1", for: .normal)
    }
    let r = URLRequest(url: crtURL!)
    wv.load(r)
}
```

MyView

```
@objc func goBack () {
    wv.goBack()
    self.setNeedsDisplay()
}

func drawInFormat (format: CGSize) {
    var top = 20;
    if UIDevice.current.userInterfaceIdiom == .phone &&
        format.height >= 812 {
        top = 30
    } else if UIDevice.current.userInterfaceIdiom == .phone &&
        format.width > format.height {
        top = 0
    }
    let hBar = 40
    wv.frame = CGRect(x:0, y:top + hBar,
        width:Int(format.width),
        height:Int(format.height - 50))
    b1.frame = CGRect(x:10, y:top + 2, width:70, height:20)
    b2.frame = CGRect(x:Int(format.width) - 80,
        y:top + 2, width:70, height:20)
    l.frame = CGRect(x:10, y:top + 20,
        width:Int(format.width)-20, height:20)
}
```

MyView

```
// WKNavigationDelegate delegate protocol

func webView(_ webView: WKWebView,
             didFinishProvisionalNavigation navigation: WKNavigation!,
             withError error: Error) {
    l.text = error.localizedDescription
}

func webView(_ webView: WKWebView,
             didReceiveServerRedirectForProvisionalNavigation
             navigation: WKNavigation!) {
    let url = URL(string: webView.url!.absoluteString) // new redirected URL
    if url != nil {
        let request = URLRequest(url: url!)
        wv.load(request)
    } else {
        l.text = "Bad redirection!!!"
    }
}

func webView(_ webView: WKWebView, didFinish navigation: WKNavigation!) {
    let t = webView.title
    if t != nil {
        l.text = "«"+t!+"» loaded"
    }
}
}
```

Various Stuff

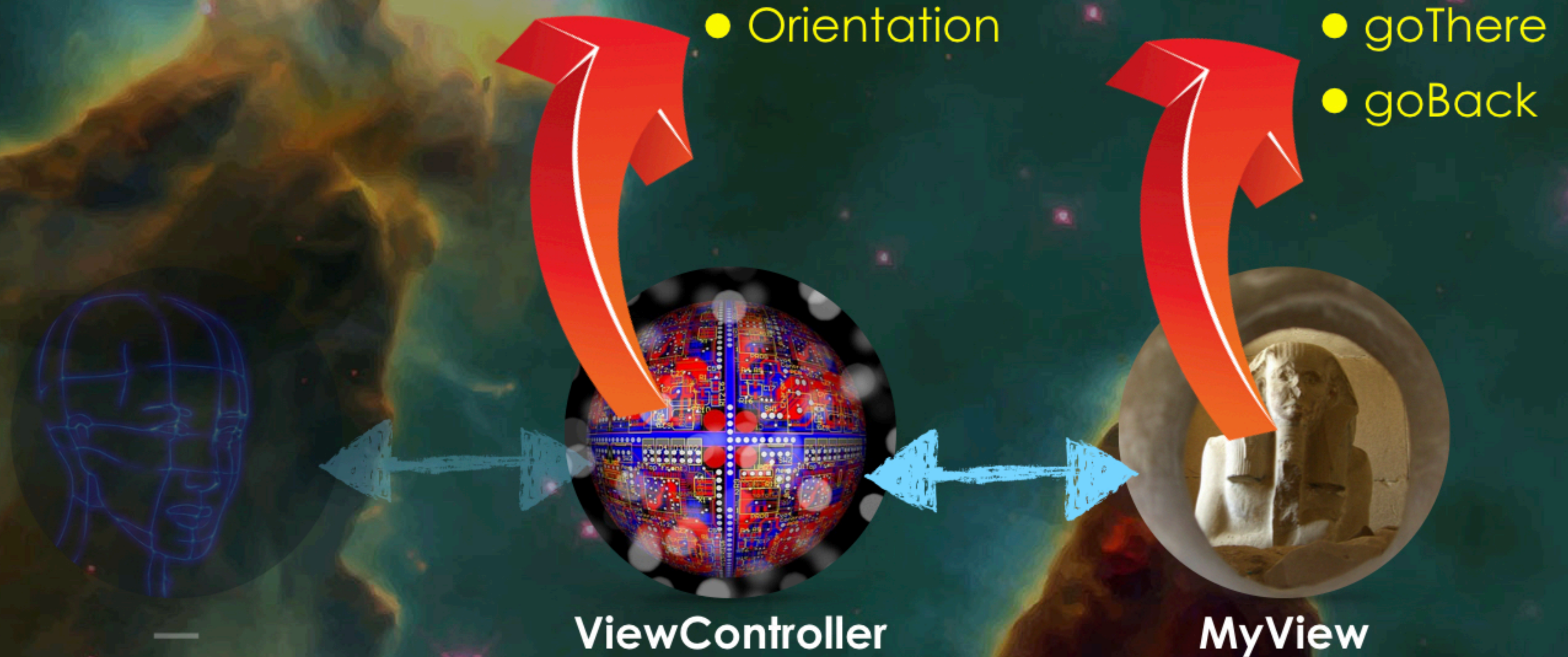


Allowing insecure communications...

The screenshot shows the Xcode interface with the Info.plist file open. The 'App Transport Security Settings' section is expanded, and the 'Allow Arbitrary Loads' key is set to YES. The table below lists the keys and values for the Info.plist file.

Key	Type	Value
▼ Information Property List Dictionary (16 items)		
Localization native development re...	String	\$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	µNav
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
▼ App Transport Security Settin...	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (i...	Array	(4 items)

What about MVC?



In MyView

- 📍 Catch of «local events»
 - ▶ This is sound
 - ▶ But but have the view controller handling everything too

As a conclusion...

Simple Isn't it?

- Objective-C version as an exercise
 - ▶ This is quite close (and his easy too)



You now know the basics of web manipulation in your iOS applications

- No excuse for lack of documentation...
- Become a **RTFM!** maker



By the way...

Some Apps are HTML-5 embedded in a WKWebView