SCIENCES SORBONNE UNIVERSITÉ

The iOS notification system

Fabrice.Kordon@lip6.fr

LIP6

As an introduction…

Already approached…

- Context of predefined entities
  - **UIDevice**
  - **MPMusicPlayerController**

Objective

- Let two objects with no relation interact
  - **Multicast communication mechanism**
  - **No direct reference between sender and receiver**

Principle (reminder)

Message passing (inside the App)

- Relies on a notification center (NotificationCenter)
  - **Fetch a reference (one shared instance for all Apps)**
  - **Property default**
- One can handle several notification queues

NotificationCenter

Registering to an event

```
func addObserver(forName name: NSNotification.Name?,
                 object obj: Any?,
                 queue: OperationQueue?,
                 using block: @escaping (Notification) -> Void)
    -> NSObjectProtocol
func addObserver(_ observer: Any,
                 selector aSelector: Selector,
                 name aName: NSNotification.Name?,
                 object anObject: Any?)
```

- You may build you own NSNotification
  - NSNotification.Name = alias to NSString (created as a rawValue)

Unregister (at last in the deinit/dealloc)

```
func removeObserver(_ observer: Any,
                    name aName: NSNotification.Name?,
                    object anObject: Any?)
func removeObserver(_ observer: Any)
```

Post

```
func post(_ notification: Notification)
func post(name aName: NSNotification.Name, object anObject: Any?)
```

Notification

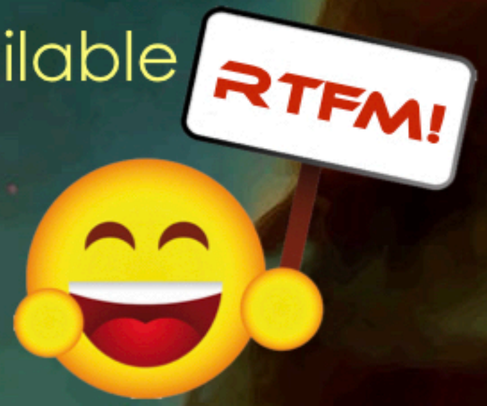
Attributes

- name
  - Name into the registry (selection of receivers)
- object
  - An object that can be sent to the receivers
- userInfo
  - More storage for sending objects (as a dictionary)

Building a Notification

```
init(name: Notification.Name,
     object: Any? = default,
     userInfo: [AnyHashable : Any]? = default)
```

- You also have NSNotification available
  - Both in Swift and Objective-C



Demo

BlaBlaClass


```swift
import UIKit

class BlaBlaClass: NSObject {

    private let myNS = NotificationCenter.default
    private var count = 1
    private var myName = ""
    private var myLabel : UILabel?

    init(_ name: String, label: UILabel) {
        super.init()
        myName = name
        myLabel = label
    }

    func prepare () {
        if myName == "A" { // It's A
            myNS.addObserver(self,
                             selector: #selector(BlaBlaClass.receive),
                             name: NSNotification.Name(rawValue: "toA"),
                             object: nil)
        } else { // It's B
            myNS.addObserver(self,
                             selector: #selector(BlaBlaClass.receive),
                             name: NSNotification.Name(rawValue: "toB"),
                             object: nil)
        }
    }
```

BlaBlaClass


```
func send () {
    if myName == "A" { // It's A
        myNS.post(Notification(name: Notification.Name(rawValue: "toB"),
                               object: nil))
    } else { // It's B
        myNS.post(Notification(name: Notification.Name(rawValue: "toA"),
                               object: nil))
    }
}

@objc func receive () {
        myLabel?.text = String(format: "Received : %d", count)
        count += 1
    }
}
```

ViewController


```
import UIKit

class ViewController: UIViewController {

    private let a : BlaBlaClass!
    private let b : BlaBlaClass!

    private let la = UILabel()
    private let lb = UILabel()
    private let acta = UILabel()
    private let actb = UILabel()

let ba = UIButton(type: .system)
let bb = UIButton(type: .system)

// use it as a main init for the ViewController
required init?(coder aDecoder: NSCoder) {
    a = BlaBlaClass("A", label: acta)
    b = BlaBlaClass("B", label: actb)
    super.init(coder : aDecoder)
}
```

ViewController


```
override func viewDidLoad() {
        super.viewDidLoad()
        let w = UIScreen.main.bounds.size.width
        let v = UIView()
        v.backgroundColor = UIColor(red: 1.0, green: 1.0, blue: 0.6, alpha: 1.0)
        self.view = v
        la.frame = CGRect(x: 20.0, y: 60.0, width: w / 2.0 - 22.0, height: 30.0)
        la.backgroundColor = UIColor.blue
        la.textColor = UIColor.white
        la.font = UIFont.boldSystemFont(ofSize: 18)
        la.textAlignment = .center
        la.text = "Object A"
        v.addSubview(la)
        lb.frame = CGRect(x: w / 2.0 + 2.0, y: 60.0, width: w / 2.0 - 22.0, height: 30.0)
        lb.backgroundColor = UIColor.red
        lb.textColor = UIColor.white
        lb.font = UIFont.boldSystemFont(ofSize: 18)
        lb.textAlignment = .center
        lb.text = "Object B"
        v.addSubview(lb)
        ba.setTitle("A to B", for: .normal)
        ba.backgroundColor = UIColor.blue
        ba.tintColor = UIColor.yellow
        ba.frame = CGRect(x: 20.0, y: 92.0, width: w / 2.0 - 22.0, height: 30.0)
        v.addSubview(ba)
        bb.setTitle("B to A", for: .normal)
        bb.backgroundColor = UIColor.red
        bb.tintColor = UIColor.yellow
        bb.frame = CGRect(x: w / 2.0 + 2.0, y: 92.0, width: w / 2.0 - 22.0, height: 30.0)
        v.addSubview(bb)
        acta.frame = CGRect(x: 20.0, y: 124.0, width: w / 2.0 - 22.0, height: 30.0)
        acta.backgroundColor = UIColor.blue
        acta.textColor = UIColor.white
        acta.font = UIFont.systemFont(ofSize: 15)
        acta.textAlignment = .center
        v.addSubview(acta)
        actb.frame = CGRect(x: w / 2.0 + 2.0, y: 124.0, width: w / 2.0 - 22.0, height: 30.0)
        actb.backgroundColor = UIColor.red
        actb.textColor = UIColor.white
        actb.font = UIFont.systemFont(ofSize: 15)
        actb.textAlignment = .center
        v.addSubview(actb)
        ba.addTarget(self, action: #selector(activateA(s:)),
                     for: .touchDown)
        bb.addTarget(self, action: #selector(activateB(s:)),
                     for: .touchDown)
        // Prepare the classes for communication
        a.prepare()
        b.prepare()
}
```

ViewController


```
    @objc func activateA(s : UIButton) {
        acta.text = "A sends"
        actb.text = ""
        a.send()
    }

    @objc func activateB(s : UIButton) {
        actb.text = "B sends"
        acta.text = ""
        b.send()
    }
}
```

As a conclusion…

Easy is'n't it?

When do you use notifications?

- Between loosely coupled objects
  - **A priori unknown to each other**
- In «multicast» situations
  - **Anonymous receivers & senders**
- Quite flexible mechanism

Be careful

- Complex to debug
- Be aware of events «flooding»