

UIActivityIndicatorView

Fabrice.Kordon@lip6.fr



As an introduction...





When processing takes some time...

- The user must «feel» something is going on
- Your solution : `UIActivityIndicatorView`
 - ▶ Can start and stop at will
 - ▶ Disappear when stops (default behavior)
- Various styles : `UIActivityIndicatorViewStyle`
 - ▶ `UIActivityIndicatorViewStyleWhiteLarge`, `UIActivityIndicatorViewStyleWhite`, `UIActivityIndicatorViewStyleGray`
 - ▶ `whiteLarge`, `white`, `gray`

Principles

Main methods

-  `init(style:)` / `initWithActivityIndicatorStyle:`
-  `startAnimating()` / `startAnimating`
-  `stopAnimating()` / `stopAnimating`

Computed attribute

-  `isAnimating` / `animating`

Attributes

-  `style` / `activityIndicatorViewStyle`
-  `color` / `color`
-  `hidesWhenStopped` / `hidesWhenStopped`

Principles

Main methods

- `init(style:)` / `initWithActivityIndicatorStyle:`
- `startAnimating()` / `startAnimating`
- `stopAnimating()` / `stopAnimating`

Computed attribute

- `isAnimating` / `animating`

Attributes

- `style` / `activityIndicatorViewStyle`
- `color` / `color`
- `hidesWhenStopped` / `hidesWhenStopped`



Example



ViewController



Sake of simplicity...

Code located in a
ViewController

ViewController

```
import UIKit
```

```
class ViewController: UIViewController {
```

```
    fileprivate let b = UIButton(type: .system)
```

```
    private let smallAct = UIActivityIndicatorView()
```

```
    private let largeAct = UIActivityIndicatorView()
```

```
    private let verylAct = UIActivityIndicatorView()
```

ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    self.view = UIView()
    self.view.backgroundColor = UIColor.white
    b.setTitle("Start", for: .normal)
    b.addTarget(self, action: #selector(startAndStop),
                for: .touchDown)
    smallAct.color = UIColor.red
    largeAct.style = .whiteLarge
    verylAct.style = .whiteLarge
    // do now otherwise color set to default
    largeAct.color = UIColor.blue
    verylAct.color = UIColor.orange
    // we need to "cheat" to have larger sizes...
    verylAct.transform = CGAffineTransform(scaleX: 3, y: 3)
    self.view.addSubview(b)
    self.view.addSubview(smallAct)
    self.view.addSubview(largeAct)
    self.view.addSubview(verylAct)
    self.displayWithSize(size: UIScreen.main.bounds.size)
}
```


ViewController

```
func displayWithSize(size : CGSize) {
    b.frame = CGRect(x: Int(size.width)/2 - 40, y: 70,
                    width: 80, height: 20)
    // Size are predefined for activities
    smallAct.frame = CGRect(x: Int(size.width)/2 - 5, y: 120,
                            width: 10, height: 10)
    largeAct.frame = CGRect(x: Int(size.width)/2 - 10, y: 160,
                            width: 20, height: 20)
    // We can enlarge because of the CGAffineTransform
    verylAct.frame = CGRect(x: Int(size.width)/2 - 30, y: 240,
                            width: 60, height: 60)
}

@objc func startAndStop() {
    if smallAct.isAnimating {
        smallAct.stopAnimating()
        largeAct.stopAnimating()
        verylAct.stopAnimating()
        b.setTitle("Stop", for: .normal)
    } else {
        smallAct.startAnimating()
        largeAct.startAnimating()
        verylAct.startAnimating()
        b.setTitle("Start", for: .normal)
    }
}
```

ViewController

```
override func viewWillTransition(to size: CGSize,  
                                with coordinator: UIViewControllerTransitionCoordinator) {  
    super.viewWillTransition(to: size, with: coordinator)  
    self.displayWithSize(size:size)  
}  
}
```

As a conclusion...



Now your users may be happy waiting...

• While a WKWebView is loading...

- ▶ Start animating when loading starts
- ▶ Stop animating when loading stops

