

Twitter Style

as 2x image
PDF → scaled nicely
FlatIcon.com

import UIKit

```
class ViewController : UIViewController {
    let twitterImage = UIImageView(image: UIImage(named: "twitter")!)
    let splashView = UIView()

    override func viewDidLoad() {
        super.viewDidLoad()
        splashView.backgroundColor = UIColor(red: 0/255, green: 172/255, blue: 237/255, alpha: 1.0)
        view.addSubview(splashView)
        splashView.frame = CGRect(x: 0, y: 0, width: view.bounds.width, height: view.bounds.height)
        twitterImage.contentMode = .scaleAspectFit
        splashView.addSubview(twitterImage)
        twitterImage.frame = CGRect(x: splashView.frame.midX - 35, y: splashView.frame.midY - 35, width: 70, height: 70)
    }
}
```

```
override func viewDidAppear(_ animated: Bool) {
    super.viewDidAppear(animated)
    DispatchQueue.main.asyncAfter(deadline: .now() + 2) {
        scaleDownAnimation()
    }
}
```

```
func scaleDownAnimation() {
    UIView.animate(withDuration: 0.5) {
        self.twitterImage.transform = CGAffineTransform(scaleX: 0.1, y: 0.1)
    } success: in {
        self.scaleUpAnimation()
    }
}
```

```
func scaleUpAnimation() {
    UIView.animate(withDuration: 0.35, delay: 0.1, options: .curveEaseIn, animations: {
        self.twitterImage.transform = CGAffineTransform(scaleX: 5, y: 5)
    }) success: in {
        self.removeSplashScreen()
    }
}
```

```
func removeSplashScreen() {
    splashView.removeFromSuperview()
}
```

II Facebook Style

Launch Screen, storyboard embedded in navigation controller
navigation Bar → uncheck translucent → Bar tint 3B5998

view →
background color d8d8d8

Main.storyboard

↗ same thing

View Controller.swift

import UIKit

```
class ViewController: UIViewController {
```

```
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }
```

}

```
    override func viewDidAppear(_ animated: Bool) {  
        super.viewDidAppear(animated)  
        showSkeleton()  
    }
```

}

```
func showSkeleton() {  
    let skeletonView = UIView()  
    skeletonView.backgroundColor = UIColor.white  
    view.addSubview(skeletonView)  
    let viewHeight = view.frame.height  
    let cellHeight: CGFloat = 250  
    let numberOfcells = Int(viewHeight / 250) + 1  
    let skeletonViewHeight = cellHeight * CGFloat(numberOfcells)
```

```
skeletonView.frame = CGRect(x: 0, y: 0, width: view.bounds.width, height: skeletonViewHeight)  
let backgroundGradientLayer = CAGradientLayer()  
backgroundGradientLayer.frame = skeletonView.bounds  
backgroundGradientLayer.startPoint = CGPoint(x: 0, y: 0.5)  
backgroundGradientLayer.endPoint = CGPoint(x: 1.0, y: 0.5)  
backgroundGradientLayer.locations = [0.25, 0.5, 0.75]  
backgroundGradientLayer.colors = [UIColor(red: 184/255, green: 184/255, blue: 184/255, alpha: 1.0).cgColor,  
                                 UIColor(red: 211/255, green: 211/255, blue: 211/255, alpha: 1.0).cgColor,  
                                 UIColor(red: 184/255, green: 184/255, blue: 184/255, alpha: 1.0).cgColor]
```

```
skeletonView.layer.addSublayer(backgroundGradientLayer)  
let gradientAnimation = CABasicAnimation(keyPath: "locations")  
gradientAnimation.fromValue = [0.0, 0.0, 0.25]  
gradientAnimation.toValue = [0.65, 1.0, 1.0]  
gradientAnimation.duration = 1  
gradientAnimation.repeatCount = .greatestFiniteMagnitude  
gradientAnimation.fillMode = kCAFillModeForwards  
gradientAnimation.isRemovedOnCompletion = false
```

```
background GradientLayer. add (gradient Animation, for key: "n!()")  
let skeletonFrame = UIBezierPath()  
let gutterHeight = CGFloat(10)  
for i in 0... numberOfCells {  
    let offset = CGFloat(i) * cellHeight  
    let profileFrame = UIBezierPath(ovalIn: CGRect(x: 20, y: 10 + offset, width: 60, height: 60))  
    skeletonFrame.append(profileFrame)  
    let titleFrame = UIBezierPath(rect: CGRect(x: 90, y: 20 + offset, width: view.bounds.width * 0.6 - 90, height: 8))  
    skeletonFrame.append(titleFrame)  
    ; similar things for subtitles, intent frame  
  
    let gutter = UIView(frame: CGRect(x: 0, y: (cellHeight - gutterHeight) + offset, width:  
        view.bounds.width, height: gutterHeight))  
    gutter.backgroundColor = UIColor(red: 216/255, green: 216/255, blue: 216/255, alpha: 1.0)  
    skeletonView.addSubview(gutter)  
}  
let skeletonShape = AShapeLayer()  
skeletonShape.path = skeletonFrame.cgPath  
skeletonShape.fillColor = UIColor.black.cgColor  
skeletonShape.frame = skeletonView.bounds  
background GradientLayer.mask = skeletonShape
```

{

}