

«MySafe»

Fabrice.Kordon@lip6.fr



Goal of the example

2



Demonstrate in-App authentication

- Handle success
- Handle the various types of authentication



Demo (code deactivated)



Demo (code)



Demo (touchID)



Demo (faceID)




In the info.plist

7



Remember, it is important to activate face ID

At least



```
<key>NSFaceIDUsageDescription</key>  
<string>We need to access faceId for authentication</string>
```


ViewController



Sake of simplicity...

Code located in a
ViewController

ViewController

```
import UIKit
import LocalAuthentication // Required

class ViewController: UIViewController {

    private let b = UIButton(type: .system)
    private let door = UIView()
    private let secret = UITextView()

    override var prefersStatusBarHidden: Bool {
        return true
    }
}
```


ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    self.view = UIView()
    self.view.backgroundColor = .white
    self.view.addSubview(secret)
    secret.isSelectable = false
    secret.isEditable = false
    secret.textAlignment = .center
    self.view.addSubview(door)
    door.backgroundColor = .black
    b.tintColor = .red
    b.setTitle("Unlock the safe", for: .normal)
    b.titleLabel?.font = UIFont.boldSystemFont(ofSize: 40)
    b.titleLabel?.numberOfLines = 3
    b.titleLabel?.textAlignment = .center
    b.addTarget(self, action: #selector(openDoor),
                for: .touchDown)
    self.view.addSubview(b)
    self.drawInFormat(UIScreen.main.bounds.size)
}
```


ViewController

```
@objc func openDoor() {
    let authContext = LAContext()
    // for code, use the line below
    let myPolicy = LAPolicy.deviceOwnerAuthentication
    // for biometrics only, use the line below
    //let myPolicy = LAPolicy.deviceOwnerAuthenticationWithBiometrics
    let err : NSErrorPointer = nil
    if authContext.canEvaluatePolicy
        (myPolicy,
         error: err) {
        authContext.evaluatePolicy(myPolicy,
                                   localizedReason: "Unlock your device please",
                                   reply: handleUnlock)
    } else {
        let a = UIAlertController(title: "Error",
                                   message: "Authentication problem",
                                   preferredStyle: .alert)
        a.addAction(UIAlertAction(title: "OK",
                                   style:.default, handler: nil))
        self.present(a, animated: true, completion: nil)
    }
}
```


ViewController

```
func handleUnlock (ok : Bool, err : Error?) {
    if ok {
        DispatchQueue.main.async{
            self.secret.text = "Duplexque isdem diebus ... reieci ad tempus."
            self.door.isHidden = true
            self.b.isHidden = true
        }
    }
}

func drawInFormat (_ s : CGSize) {
    self.view.frame = CGRect(x: 0, y: 0, width: s.width, height: s.height)
    secret.frame = CGRect(x: 50, y: 70, width: s.width - 100,
        height: s.height - 140)

    if !door.isHidden {
        b.frame = CGRect(x: s.width / 2 - 80, y: s.height / 2 - 40,
            width: 160, height: 80)
        door.frame = CGRect(x: 0, y: 0, width: s.width, height: s.height)
    }
}

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


ViewController

```
func handleUnlock (ok : Bool, err : Error?) {
    if ok {
        DispatchQueue.main.async{
            self.secret.text = "Duplexque isdem diebus ... reieci ad tempus."
            self.door.isHidden = true
            self.b.isHidden = true
        }
    }
}

func drawInFormat (_ s : CGSize) {
    self.view.frame = CGRect(x: 0, y: 0, width: s.width, height: s.height)
    secret.frame = CGRect(x: 50, y: 70, width: s.width - 100,
        height: s.height - 140)
    if !door.isHidden {
        b.frame = CGRect(x: s.width / 2 - 80, y: s.height / 2 - 40,
            width: 160, height: 80)
        door.frame = CGRect(x: 0, y: 0, width: s.width, height: s.height)
    }
}

override func draw(_ coordinator) {
    self.draw
}
```



Of course!!!

Never put your secret like this in code

As a conclusion...

 **Easy is'n't it?**



 **This is a useful feature for your Applications**

-  Permission before performing «touchy» actions