SCIENCES SORBONNE UNIVERSITÉ

«MyContacts»

Fabrice.Kordon@lip6.fr

LIP6

Goals of the example

Manipulate contacts

- Reading these
  - Selection of a contact
  - Display some gathered information
- Writing these
  - Adding a contact

Giving you an idea about this framework

- And let you feel like having a look at RTFM!

Demo (contact access deactivated)

Demo (contact access activated)

ViewController


```
import UIKit

class ViewController: UIViewController {

    private let v = MyView(frame: UIScreen.main.bounds)

    override var preferredStatusBarStyle : UIStatusBarStyle {
        return .lightContent
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        self.view = v
    }

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

MyView


```
import UIKit
import Contacts
import ContactsUI

class MyView: UIView, CNContactPickerDelegate, CNContactViewControllerDelegate {

    private let name = UILabel()
    private let firstname = UILabel()
    private let phone = UILabel()
    private let photo = UIImageView()
    private let toolb = UIToolbar()
    private let backg = UIImageView(image: UIImage(named: "background"))

    private let contactsC = CNContactPickerViewController()
    private let contactsS = CNContactStore()
    private var contactVC : CNContactViewController?
```

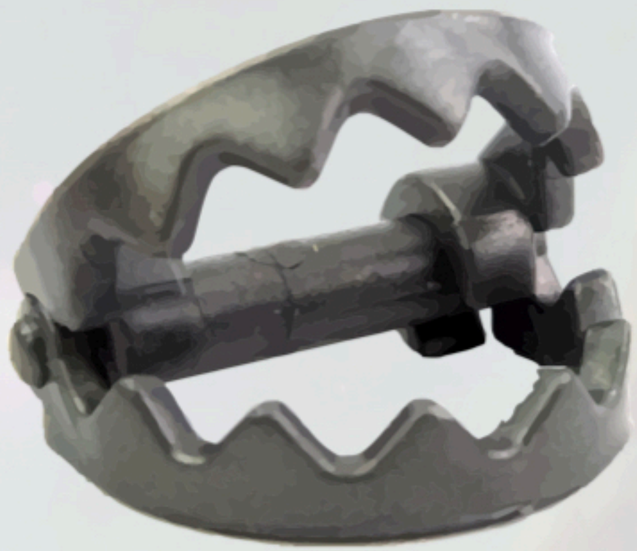
MyView


```
override init(frame: CGRect) {
    super.init(frame: frame)
    self.addSubview(backg)
    let b1 = UIBarButtonItem(barButtonSystemItem: .search,
                             target: self, action: #selector(search))
    let b2 = UIBarButtonItem(barButtonSystemItem: .add,
                             target: self, action: #selector(add))
    let sp = UIBarButtonItem(barButtonSystemItem: .flexibleSpace,
                             target: nil, action: nil)
    toolb.items = [b1, sp, b2];
    toolb.barStyle = .black
    self.addSubview(toolb);
    name.textColor = .white
    self.addSubview(name)
    firstname.textColor = .white
    self.addSubview(firstname)
    phone.textColor = .white
    self.addSubview(phone)
    photo.backgroundColor = UIColor(red: 0.8, green: 0.8, blue: 0.8, alpha: 0.7)
    self.addSubview(photo)
    self.displayInFormat(UIScreen.main.bounds.size)
    contactsC.delegate = self
    self.displayContacts(contact: nil)
}

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

MyView


```
@objc func search() {
    contactsS.requestAccess(for: .contacts) { (b : Bool, e : Error?) in
        // important, to have it executed on the main thread (to reach
        // UIApplication.shared.windows in a safe way
        DispatchQueue.main.async {
            let vc = UIApplication.shared.windows[0].rootViewController
            if !b {
                let a = UIAlertController(title: "Error",
                                          message: "Access to contact refused",
                                          preferredStyle: .alert)
                a.addAction(UIAlertAction(title: "OK",
                                          style: .default, handler: nil))
                vc?.present(a, animated: true, completion: nil)
            } else {
                vc?.present(self.contactsC, animated: true, completion: nil)
            }
        }
    }
}
```



MyView


```
@objc func add() {
    contactsS.requestAccess(for: .contacts) { (b : Bool, e : Error?) in
        let vc = UIApplication.shared.windows[0].rootViewController
        if !b {
            let a = UIAlertController(title: "Error",
                                      message: "Access to contact refused",
                                      preferredStyle: .alert)
            a.addAction(UIAlertAction(title: "OK",
                                      style: .default, handler: nil))
            vc?.present(a, animated: true, completion: nil)
        } else {
            let x = CNMutableContact()
            self.contactVC = CNContactViewController(forNewContact: x)
            self.contactVC!.delegate = self
            let n = UINavigationController(rootViewController: self.contactVC!)
            vc?.present(n, animated: true, completion: nil)
        }
    }
}
```

MyView


```
func displayContacts(contact : CNContact?) {
    if contact != nil {
        name.text = "Name : "
        if contact!.isKeyAvailable(CNContactFamilyNameKey) {
            name.text = name.text! + contact!.familyName
        }
        firstname.text = "Firstname : "
        if contact!.isKeyAvailable(CNContactGivenNameKey) {
            firstname.text = firstname.text! + contact!.givenName
        }
        phone.text = "Phone : "
        if contact!.isKeyAvailable(CNContactPhoneNumbersKey) {
            for n: CNLabeledValue in contact!.phoneNumbers {
                let num = n.value.stringValue // Get the right number format
                phone.text = phone.text! + num + ", "
            }
        } else {
            phone.text = "Phone : <none>"
        }
        if contact?.thumbnailImageData != nil {
            photo.image = UIImage(data: contact!.thumbnailImageData!)
            self.displayInFormat(UIScreen.main.bounds.size)
        }
    } else {
        name.text = "Name :"
        firstname.text = "Firstname :"
        phone.text = "Phone :"
        photo.image = nil
    }
}
```

MyView


```
func displayInFormat(_ s: CGSize) {
    var top = 20;
    if UIDevice.current.userInterfaceIdiom == .phone &&
        s.height >= 812 {
        top = 30
    } else if UIDevice.current.userInterfaceIdiom == .phone &&
        s.width > s.height {
        top = 0
    }
    toolb.frame = CGRect(x: 0, y: top,
                         width: Int(s.width), height: 60)
    name.frame = CGRect(x: 10, y: top + 80,
                        width: Int(s.width - 20), height: 30)
    firstname.frame = CGRect(x: 10, y: top + 120,
                             width: Int(s.width - 20), height: 30)
    phone.frame = CGRect(x: 10, y: top + 160,
                         width: Int(s.width - 20), height: 30)
    var w = photo.image?.size.width ?? 0.0
    var h = photo.image?.size.height ?? 0.0
    while w > UIScreen.main.bounds.width {
        w = w / 2
        h = h / 2
    }
    photo.frame = CGRect(x: Int(s.width / 2 - w / 2),
                         y: top + 200,
                         width: Int(w),
                         height: Int(h))
    backg.center = CGPoint(x: s.width / 2, y: s.height / 2)
}
```

MyView


```
// CNContactPickerDelegate protocol

func contactPicker(_ picker: CNContactPickerViewController,
                   didSelect contact: CNContact) {
    self.displayContacts(contact: contact)
}

//func contactPicker(_ picker: CNContactPickerViewController,
//                   didSelect contacts: [CNContact]) {
//    // to implement for multi-selection of contacts
//}

    func contactPickerDidCancel(_ picker: CNContactPickerViewController) {
        self.displayContacts(contact: nil)
    }

    // ContactViewControllerDelegate protocol

    func contactViewController(_ viewController: CNContactViewController,
                               didCompleteWith contact: CNContact?) {
        contactVC?.dismiss(animated: true, completion: nil)
        contactVC = nil
        self.displayContacts(contact: contact)
    }
}
```

As a conclusion…

We did it!

Remember

- This is a rich framework
  - **Refined selection, update, etc.**
- This is a standard Framework
  - **It enables secure access for your users**

You must use id… if you need it f course