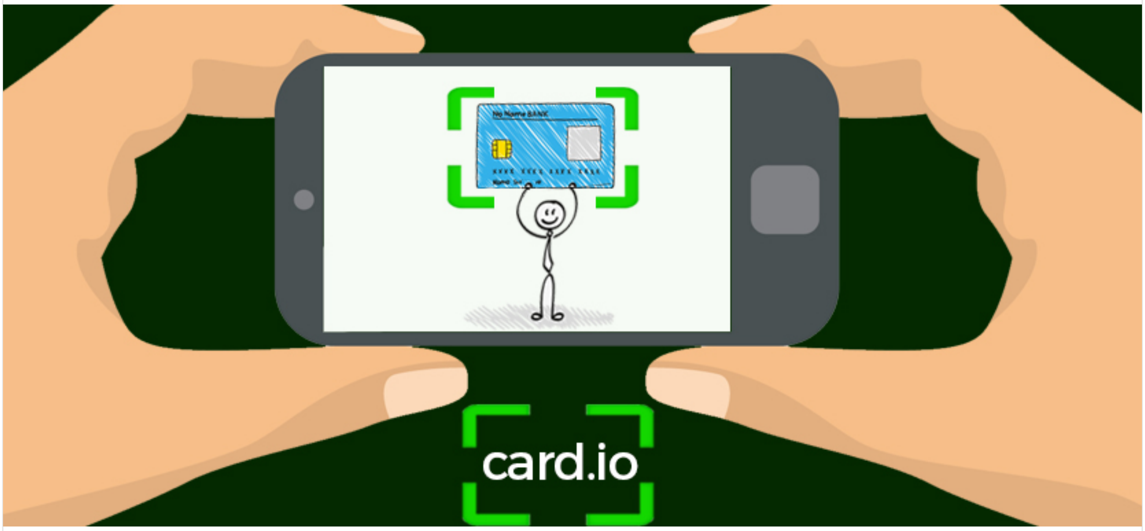
A Swift tutorial to integrate card.io



By Mobisoft Team In MGuide Posted April 28, 2016

What is CARD.IO ?

The card.io provides friendly API's and methods ready to use. With card.io, you can scan the credit card details like card no, exp. date, name of cardholder in your iOS / Android app in a fast and easy way.

Getting Started

1. Download the starter project 'CardIO_Starter_Project' from here and open it in the Xcode.

2. Next step = Do you use cocoa pods ? Yes, Jump to step #5 : No, Jump to step #3;

3. Download the latest version of card.io SDK from here.

4. Add the 'CardIO' directory into the project and skip the next step.

5. Simply add pod 'CardIO' into the podfile and update.

6. Go to the TARGET -> BUILD SETTINGS and set OTHER LINKER FLAG to "-lc++" and "-ObjC"

7. Add following frameworks as Optional libraries

- Accelerate
- AVFoundation
- AudioToolbox

Search here

LATEST POSTS

Divide and Conquer Approach to Quality Assurance

Apple's Search Ads Is Here. Get $ Credits Towards Your First Ad Campaign!

Microsoft Ignite 2016 Focuses On Security, Intelligence And The Clou

Apple Launches iOS 10. What's N For The Developers!

Deep Linking: An Essential Tool fo Mobile App Marketers

MOST VIEWED POSTS

iPhone UITableView Tutorial: Grou Table

Android Studio 2.2 Preview

Swift Code Syntax : Important Ch sheet & references for Swift programmers

iPhone Open Source libraries

Say No to SPAM!

SUBSCRIBE US

Enter Email Address

SUBMIT

TAGS

- CoreMedia
- MobileCoreServices

8. Finally, Confirm that following two settings (TARGET->BUILD SETTINGS) are enabled, if not enable them

- Enable Modules (C and Objective-C)
- Link Frameworks Automatically

No Tags for this post.

Using Objective-C into Swift Project

The Card.io framework is written in Objective-C, and our project is written in Swift. These two don't work together without a proper configuration. Any Objective-C library, project or class can be used in a Swift project by setting up a bridging header.

You can create the bridging file by using following steps,

1. Add a new Header file into project, using File – > New -> File -> iOS -> Source -> Header File.

2. Name the class <your-project-name>-Bridging-Header.h and press continue to save the file.

3. Go to Build Settings -> Objective-C Bridging Header and set as '<your-project-name>-Bridging-Header.h'.

4. Add "#import "CardIO.h" in to your bridging header file.

Using Card.io

There are two ways to use the card.io in to the project, Choose as per your requirement

Ready Made (As a View Controller)

As the title suggest, it is ready to use controller and it is very quick and easy. What you have to do is, Just create a instance of a class 'CardIOPaymentViewController' and present it modally, And it will handle rest all.

Do It Yourself (As a View)

If you are the DIY kind of developer and want to more control over the transitions and presentation of the scanner, then you should use this approach. This will only provide the magic for the scanning the card, rest handle yourself. You have to create a instance of 'CardIOView' which will handle the scanning.

In this tutorial, I will cover the only the Ready Made approach, because most of us not really required the modifications in the scanner but we will cover DIY approach in the next part of this tutorial.

Integrating As a View Controller

- Open 'ReadyMadeViewController.swift' class file from our 'CardioDemoStarter' project. And add CardIOPaymentViewControllerDelegate to the class.

```
class ReadyMadeViewController: UIViewController,
CardIOPaymentViewControllerDelegate {

}
```

- The first time that you create either a CardIOPaymentViewController or a CardIOView, the card.io SDK must load resources, which can result in a noticeable delay. To avoid this delay you may optionally call CardIOUtilities.preload() in advance, so that this resource loading occurs in advance on a background thread.

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from
    CardIOUtilities.preload()
}
```

- Now create a actual scanner and start the scanning add the following code in the “scanButtonClicked” action.

```
/// Create CardIOPaymentViewController object with paymentDelegate to
let cardIOVC = CardIOPaymentViewController(paymentDelegate: self)
```

- Also there are few optional configurations in the scanner.

```
/// Set to true if you need to collect the cardholder name. Defaults t
cardIOVC.collectCardholderName = true



/// Hide the PayPal or card.io logo in the scan view. Defaults to fals
cardIOVC.hideCardIOLogo = true



/// Set the guide color, Defaults to nil; if nil, will use card.io gre
```

```
/// Set to false if you don't need to collect the CVV from the user. I
cardIOVC.collectCVV = true
```

- Finally present the scanner modally.

```
/// Present Card Scanner View modally.
presentViewController(cardIOVC, animated: true, completion: nil)
```

- Now, Implement the delegate method to get card info or cancellation.

```
    func userDidCancelPaymentViewController(paymentViewController: CardIOP
        labelResults.text = "User has canceled the scanner."
        paymentViewController.dismissViewControllerAnimated(true, completi
     }

    func userDidProvideCreditCardInfo(cardInfo: CardIOCreditCardInfo?,
    inPaymentViewController paymentViewController: CardIOPaymentViewContro

    if let info = cardInfo {
        let str = NSString(format: "Received card info.\n Cardholders Name
        redactedCardNumber, info.expiryMonth, info.expiryYear, info.cvv)
        l abelResults.text = str as String
    }
    paymentViewController.dismissViewControllerAnimated(true, completion:
}
```

Build and Run

You may download the complete project **‘CardIO_Complete_Project’** from here.

Happy Coding!

Written by – Shrimant Nikate.

iOS Developer – Mobisoft Infotech.

Like what you have read

Like Tweet Pin It Follow

RELATED ARTICLES You may like

In MGuide | December 18, 2014

In Apple - MaciPhone - iPad | February 25, 2012

In Blog | December 15, 2015

Implementing Background Fetch in iOS

55 million opportunities to make new iPad Apps

Mobile App Marketing – How to get better rankings and increase App Downloads?

Project Planner

Use our project planner to estimate the deliverables and pricing for your product needs.

GET A FREE QUOTE

Company

- Startup Solutions
- Enterprise Solutions
- Technology Stack
- Work
- Careers
- Partners
- Contact
- Sitemap

Solutions

- Catalog Pulse
- GeoMosquito
- Taxi Pulse
- IR Pulse
- Alumni Pulse
- Slate One
- Voter Pulse
- Ecommerce Pulse

Services

- Mobile Application Development
- Android Application Development
- iPhone Application Development
- Web App Development
- Hire Dedicated Developers
- Highway Safety Solutions
- Apple Watch Development
- Healthcare Mobility Solutions

Resources

- Blog
- Mobi Toons
- White Papers
- Newsletters
- M-Design
- M-Guide
- Media

Reach out to us!

Name

Email

Phone

Country

Message

SUBMIT

Awarded 8th Fastest growing company (Houston)

Best Android Development Company (USA)

Best iPhone Development Company (USA)

Accredited Business

Clutch TOP IPHONE APP DEVELOPERS 2015

★★★★★ Rated 4.5 / 5.0 by 150 + Clients for Best iOS and Android Mobile Application Development Services.

Discover Mobility by reaching us @ our offices in Pune (India), Houston (USA), Melbourne (Australia), Jurong West (Singapore)