How to use QRCoder static library

## This document will give you a high-level information and examples about how to use the QRCoder library. Please refer to the Integration document for getting started and adding the library to your project.

#### REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **VERSION** | **AUTHOR** | **UPDATE DESCRIPTION** |
| Aug 8, 2020 | 1.0 | Gautham Velappan | Initial Draft |
| Aug 17, 2020 | 1.1 | Gautham Velappan | Scanner View updates and bug fixes |
| Aug 25, 2020 | 1.2 | Gautham Velappan | Compression feature for input data |
|  |  |  |  |

General Usage

The library has 3 important classes for QR code handling,

1. Encode
2. Decode
3. ScannerView

Encode

This class gives you ability to generating QR code images

First, create an Encoder instance.

1. let qrCodeEncoder = QRCodeEncoder()

### Create a QR code for a string:

1. let string: String = String("John Smith aged 21")
3. qrCodeEncoder.encode(**for**: string, color: .red) { image, error **in**
4. **if** let error = error {
5. print(error.localizedDescription)
6. } **else** {
7. self.myImageView1?.image = image
8. }
9. }

### Create a QR code for a dictionary:

1. let dictionary: [String : Any] = [
2. "Name": "John Smith",
3. "Age": 21
4. ]
6. qrCodeEncoder.encode(**for**: dictionary, backgroundColor: .green, errorCorrection: .quartile) { image, error **in**
7. **if** let error = error {
8. print(error.localizedDescription)
9. } **else** {
10. self.myImageView2?.image = image
11. }
12. }

### Create a QR code for any data:

1. let array: [Any] = [
2. "John Smith", 21
3. ]
5. let data = **try**? JSONSerialization.data(withJSONObject: array, options: [])
7. qrCodeEncoder.encode(**for**: data!, size: CGSize(width: 4096, height: 4096)) { image, error **in**
8. **if** let error = error {
9. print(error.localizedDescription)
10. } **else** {
11. self.myImageView3?.image = image
12. }
13. }

Decode

This class gives you ability to fetch details of a provided QR code image

First, create a Decoder instance.

1. let qrCodeDecoder = QRCodeDecoder()

### Generate details of a provided image:

1. let image: UIImage = UIImage(named: "qrCode")!
3. qrCodeDecoder.decode(image: image) { messages, details, error **in**
4. **if** let error = error {
5. print(error.localizedDescription)
6. } **else** {
7. print("MessageString: \(messages)")
8. }
9. }

ScannerView

First, create a UIView for the camera layer in your storyboard or xib and update the custom *class* to **QRCoderScannerView** with **QRCoder** as the *Module*.

A picture containing screenshot

Description automatically generated

### Connect the IBOutlet and update the properties,

1. @IBOutlet weak **var** scannerView: QRCodeScannerView!

### Preferably in viewDidLoad,

1. scannerView.stopScannerAfterDecode = **true**
2. scannerView.delegate = self

### the QRCoderScannerViewDelegate methods will now provide the output for the scan results,

1. func scannerView(\_ scannerView: QRCodeScannerView, didScanQRCode message: String) {
2. print("QR code decoded message: \(message)")
3. }
5. func scannerView(\_ scannerView: QRCodeScannerView, didScanQRCodeDetails details: [[String: Any]]) {
6. print("Scan view details: \(details)")
7. }
8. func scannerView(\_ scannerView: QRCodeScannerView, didReceiveError error: Error) {
9. print("Scan view error: \(error.localizedDescription)")
10. }