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.

```
let qrCodeEncoder = QRCodeEncoder()
```

Create a QR code for a string:

```
let string: String = String("John Smith aged 21")

qrCodeEncoder.encode(for: string, color: .red) { image, error in
    if let error = error {
        print(error.localizedDescription)
    } else {
        self.myImageView1?.image = image
    }
}
```

Create a QR code for a dictionary:

```
let dictionary: [String : Any] = [
    "Name": "John Smith",
    "Age": 21
]

qrCodeEncoder.encode(for: dictionary, backgroundColor: .green, errorCorrection: .quartile) { image, error in
    if let error = error {
        print(error.localizedDescription)
    } else {
        self.myImageView2?.image = image
    }
}
```

Create a QR code for any data:

```
let array: [Any] = [
    "John Smith", 21
]

let data = try? JSONSerialization.data(withJSONObject: array, options: [])

qrCodeEncoder.encode(for: data!, size: CGSize(width: 4096, height: 4096)) { image, erro
r in
    if let error = error {
        print(error.localizedDescription)
    } else {
        self.myImageView3?.image = image
    }
}
```

Decode

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

First, create a Decoder instance.

```
let qrCodeDecoder = QRCodeDecoder()
```

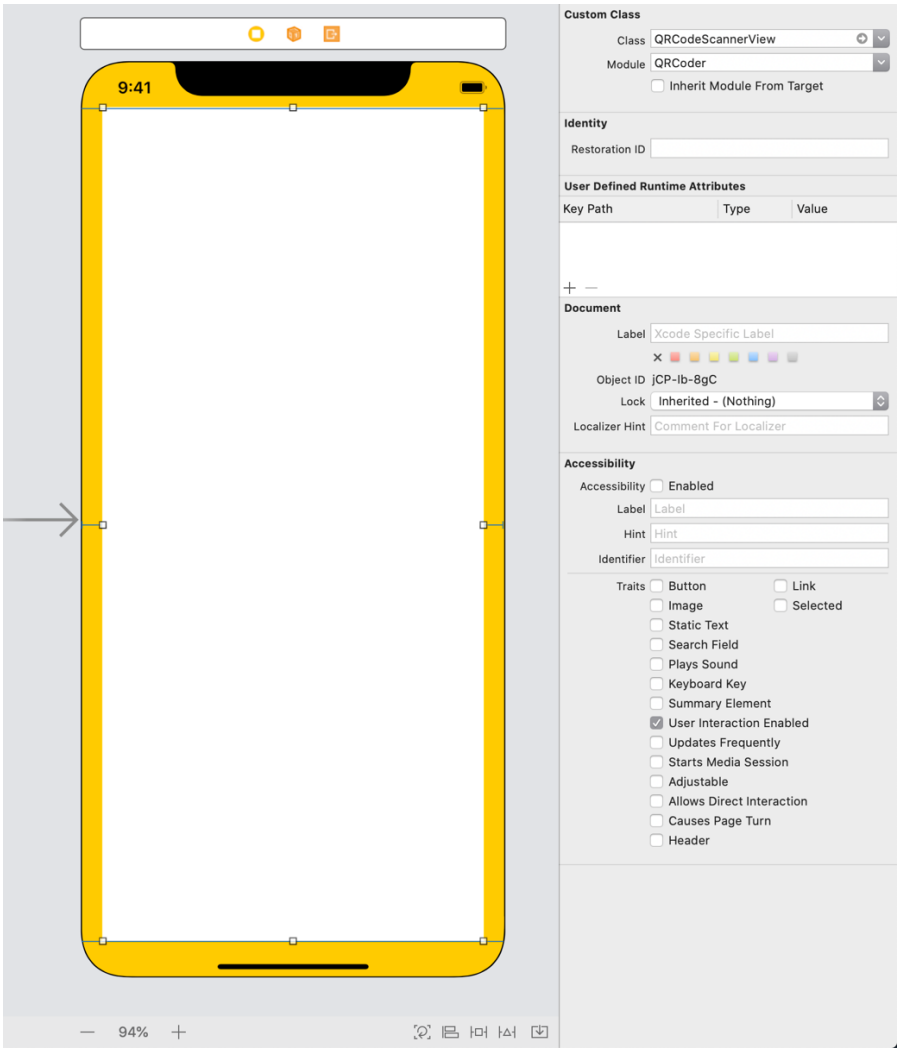
Generate details of a provided image:

```
let image: UIImage = UIImage(named: "qrCode")!

qrCodeDecoder.decode(image: image) { messages, details, error in
    if let error = error {
        print(error.localizedDescription)
    } else {
        print("MessageString: \(messages)")
    }
}
```

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*.



Connect the IBOutlet and update the properties,

```
@IBOutlet weak var scannerView: QRCodeScannerView!
```

Preferably in viewDidLoad,

```
scannerView.stopScannerAfterDecode = true
scannerView.delegate = self
```

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

```
func scannerView(_ scannerView: QRCodeScannerView, didScanQRCode message: String) {
    print("QR code decoded message: \(message)")
}

func scannerView(_ scannerView: QRCodeScannerView, didScanQRCodeDetails
details: [[String: Any]]) {
    print("Scan view details: \(details)")
}

func scannerView(_ scannerView: QRCodeScannerView, didReceiveError error: Error) {
    print("Scan view error: \(error.localizedDescription)")
}
```