

QR Generator Technical Documentation-iOS

QRGenerator

QR Generator framework allows users to generate a TLV from the map. Issuer will provide the details dictionary format. This dictionary data will be processed based on EMVCo specifications and output will be provided to the issuer. The output can be either the response TLV string or an appropriate error code.

What's in QRGeneratorSDK- $\{x.y.z\}$.zip :

Inside QRGeneratorSDK- $\{x.y.z\}$.zip, there will be QRGenerator - $\{x.y.z\}$ -DEBUG-SWIFT4.0.zip and QRGenerator - $\{x.y.z\}$ -PRODUCTION-SWIFT4.0.zip zip files:

QRGenerator- $\{x.y.z\}$ -DEBUG-SWIFT4.0.zip

The content of package are to be used during the development phase of the issuer application. This package contain binaries for universal (debug/development) architecture supporting simulator and device, but not eligible for app store submission.

This folder will have following data :

Name	Description
QRGenerator .framework	Contains the public interface for integrating with QRGenerator .framework
SampleApp	Sample App which use QRGenerator .framework
CHANGE_LOG	Change log file to show all changes on QRGenerator .framework for each version

QRGenerator- $\{x.y.z\}$ -PRODUCTION-SWIFT4.0.zip

The content of package are to be used in the release version of the issuers application. The binaries in this package supports device only architectures and eligible for app store submission.

This folder will have following data :

Name	Description
QRGenerator .framework	Contains the public interface for integrating with QRGenerator .framework
CHANGE_LOG	Change log file to show all changes on QRGenerator .framework for each version

How to integrate QRGenerator.framework:

To integrate QRGenerator .framework in your project you will need to drag and drop the .framework file to your project. Once the framework is imported you can import it in any file by calling the following line of code.

```
import QRGenerator
```

Now to launch the generator you can use the following line of codes

```

var qrData = [String: AnyObject]()
qrData[QRCodeTag.jcb1] = "77777777" as AnyObject
qrData[QRCodeTag.currencyCode] = "356" as AnyObject
qrData[QRCodeTag.pointOfInitiation] = "11" as AnyObject
qrData[QRCodeTag.postalCode] = "401104" as AnyObject
qrData[QRCodeTag.amex1] = "99999999" as AnyObject
qrData[QRCodeTag.countryCode] = "IN" as AnyObject
qrData[QRCodeTag.cityName] = "Mumbai" as AnyObject
qrData[QRCodeTag.payloadFormatIndicator] = "01" as AnyObject
qrData[QRCodeTag.merchantCategoryCode] = "6010" as AnyObject
qrData[QRCodeTag.discover1] = "72222222" as AnyObject
qrData[QRCodeTag.npciid1] = "91111111" as AnyObject
qrData[QRCodeTag.mVisaMerchantID] = "47613617" as AnyObject
qrData[QRCodeTag.unionPay1] = "66666666" as AnyObject
qrData[QRCodeTag.tag17] = "33333333" as AnyObject
qrData[QRCodeTag.masterCardPAN1] = "51111111" as AnyObject
qrData[QRCodeTag.merchantName] = "Corner Store" as AnyObject
qrData[QRCodeTag.additionalDataFieldTemplate] = [QRCodeTag.billID: "5001", QRCodeTag.mobileNumber: "7411566609"] as
AnyObject
QRCodeGenerator.generateQRCodeString(qrCodeDictionary: qrData) { (generatorResponse) in
    print("***** QR Generator positive scenario testcase *****")
    print("TLV -> \(String(describing: generatorResponse?.tlvString))")
}

```

QRGenerator.framework API Detail:

This section consists of other important information related to the use and support of the SDK. As some of the information may be subjected to regular updates such as supported handset devices and IDE as well as list of software limitations, it will be advisable to refer to the information on the Visa Developers Portal for the latest update if needed.

API Class and Methods

The following describes the methods used in QRGenerator with their use cases, input & output parameters.

USE CASE	ENTRY POINT METHOD	INPUT PARAMETERS	OUTPUT PARAMETERS
Generate QR code data string from map	QRGenerator .generateQRCodeString(qrCodeDictionary: qrCodeDic)	The raw qrCodeDic which is a map based representation of QR data	QRGeneratorResponse object – Model object containing tlvString and qrCodeError which throws string/list of error codes

Class - QRCodeGenerator

The QRCodeGenerator class provides sync and async methods to parse QR dictionary, all the available methods are class level. To generate QR code, call class method generateQRCodeString: with required parameters. If the parsing of map is successful, then the response object will provide the QR TLV string as output. In case if the parsing of map fails, the response object will provide the array of error codes. Error codes will help to describe the errors because of which the generation is failed.

Method - generateQRCodeString

- Generate TLV string and return QRGeneratorResponse object with parsed data and error object.
- Parameter qrCodeDictionary: QR code data in dictionary format which need to be parsed to generate TLV string.
- Return QRGeneratorResponse object with parsed data.
- QRGeneratorResponse has 2 variables, tlvString of type String which has parsed QR data and qrCodeError of type QRCodeError which has error information.
- If qrCodeError key is not in output means generating is successfully done and tlvString will have the QR TLV.
- If qrCodeError key is there in output means generator has failed with some error and qrCodeError key will provide json array of error code

API Objects

QRCodeTag Class: This class will provide variables for tags.

Variable Name
@objc public static let payloadFormatIndicator = "00"
@objc public static let pointOfInitiation = "01"
@objc public static let mVisaMerchantID = "02"
@objc public static let masterCardPAN1 = "04"
@objc public static let masterCardPAN2 = "05"
@objc public static let npciid1 = "06"
@objc public static let npciid2 = "07"
@objc public static let tag08 = "08"
@objc public static let discover1 = "09"
@objc public static let discover2 = "10"
@objc public static let amex1 = "11"
@objc public static let amex2 = "12"
@objc public static let jcb1 = "13"
@objc public static let jcb2 = "14"
@objc public static let unionPay1 = "15"
@objc public static let unionPay2 = "16"
@objc public static let tag17 = "17"
@objc public static let tag18 = "18"
@objc public static let tag19 = "19"
@objc public static let tag20 = "20"
@objc public static let tag21 = "21"
@objc public static let tag22 = "22"
@objc public static let tag23 = "23"
@objc public static let tag24 = "24"
@objc public static let tag25 = "25"
@objc public static let tag26 = "26"
@objc public static let tag27 = "27"
@objc public static let tag28 = "28"
@objc public static let tag29 = "29"
@objc public static let tag30 = "30"
@objc public static let tag31 = "31"
@objc public static let tag32 = "32"
@objc public static let tag33 = "33"
@objc public static let tag34 = "34"
@objc public static let tag35 = "35"
@objc public static let tag36 = "36"
@objc public static let tag37 = "37"
@objc public static let tag38 = "38"
@objc public static let tag39 = "39"
@objc public static let tag40 = "40"
@objc public static let tag41 = "41"
@objc public static let tag42 = "42"
@objc public static let tag43 = "43"
@objc public static let tag44 = "44"
@objc public static let tag45 = "45"
@objc public static let tag46 = "46"
@objc public static let tag47 = "47"
@objc public static let tag48 = "48"
@objc public static let tag49 = "49"
@objc public static let tag50 = "50"
@objc public static let tag51 = "51"
@objc public static let merchantCategoryCode = "52"
@objc public static let currencyCode = "53"
@objc public static let transactionAmount = "54"
@objc public static let tipAndFeeIndicator = "55"

```
@objc public static let convenienceFeeAmount = "56"  
@objc public static let convenienceFeePercentage = "57"  
@objc public static let countryCode = "58"  
@objc public static let merchantName = "59"  
@objc public static let cityName = "60"  
@objc public static let postalCode = "61"  
@objc public static let additionalDataFieldTemplate = "62"  
@objc public static let billID = "6201"  
@objc public static let mobileNumber = "6202"  
@objc public static let storeID = "6203"  
@objc public static let loyaltyNumber = "6204"  
@objc public static let referenceID = "6205"  
@objc public static let consumerID = "6206"  
@objc public static let terminalID = "6207"  
@objc public static let purpose = "6208"  
@objc public static let additionalConsumerDataRequest = "6209"  
@objc public static let tagSixtyTwo10 = "6210"  
@objc public static let addDataMasterCard1 = "6211"  
@objc public static let addDataMasterCard2 = "6212"  
@objc public static let addDataNpci1 = "6213"  
@objc public static let addDataNpci2 = "6214"  
@objc public static let crc = "63"  
@objc public static let merchantInformationLanguageTemplate = "64"  
@objc public static let tag65 = "65"  
@objc public static let tag66 = "66"  
@objc public static let tag67 = "67"  
@objc public static let tag68 = "68"  
@objc public static let tag69 = "69"  
@objc public static let tag70 = "70"  
@objc public static let tag71 = "71"  
@objc public static let tag72 = "72"  
@objc public static let tag73 = "73"  
@objc public static let tag74 = "74"  
@objc public static let tag75 = "75"  
@objc public static let tag76 = "76"  
@objc public static let tag77 = "77"  
@objc public static let tag78 = "78"  
@objc public static let tag79 = "79"  
@objc public static let tag80 = "80"  
@objc public static let tag81 = "81"  
@objc public static let tag82 = "82"  
@objc public static let tag83 = "83"  
@objc public static let tag84 = "84"  
@objc public static let tag85 = "85"  
@objc public static let tag86 = "86"  
@objc public static let tag87 = "87"  
@objc public static let tag88 = "88"  
@objc public static let tag89 = "89"  
@objc public static let tag90 = "90"  
@objc public static let tag91 = "91"  
@objc public static let tag92 = "92"  
@objc public static let tag93 = "93"  
@objc public static let tag94 = "94"  
@objc public static let tag95 = "95"  
@objc public static let tag96 = "96"
```

```
@objc public static let tag97 = "97"  
@objc public static let tag98 = "98"  
@objc public static let tag99 = "99"
```