Payrix Mobile

Software Development Kit (SDK)

iOS Supplementary
Developer Guide

Version 3.0.21
Revision Date: 05.20.2024

TABLE OF CONTENTS

Overview

This Payrix Mobile SDK for iOS current enhancement now allows another medium of authentication since the current Login session would be deprecated soon. This document explains how to implement this new function or authentication called TxnSession. The demo app provides assistance on how to perform this new enhancement.

1. **Authentication:** To initiate Authentication, developers must utilize the TxnSessionConfig Builder class, ensuring precise parameter configuration. The callback function, didReceiveTxnKeyResult, delivers success status, a PayCoreTxnSession object, and server messages.

2. **Payment Transaction Using TxnSessionKey**: When conducting payment transactions, developers can choose between TxnSessionKey or PaySessionKey. Depending on the preference, useTxnSessionKey should be set accordingly, with paySessionKey set to nil if TxnSessionKey is used, and vice versa.

3. **Fetching Transaction Lists:** The method for fetching transaction lists requires parameters indicating TxnSessionKey or PaySessionKey usage. Developers should set payTxnSessionKey accordingly, ensuring consistency in parameter assignment.

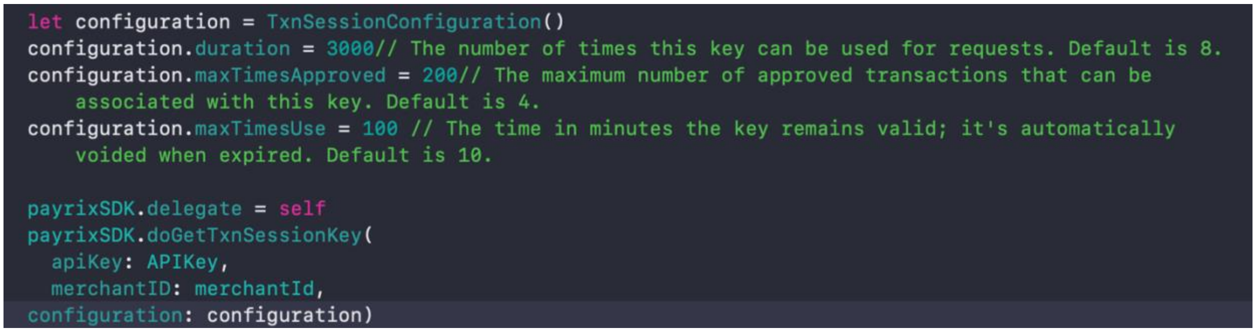
4. **Subsequent Transactions:** Similar to other methods, developers can specify useTxnSession and set paySessionKey or payTxnSessionKey based on requirements for subsequent transactions.

5. **Refund Eligibility Check:** To verify refund eligibility, developers can set useTxnSessionKey and configure paySessionKey and payTxnSessionKey accordingly.

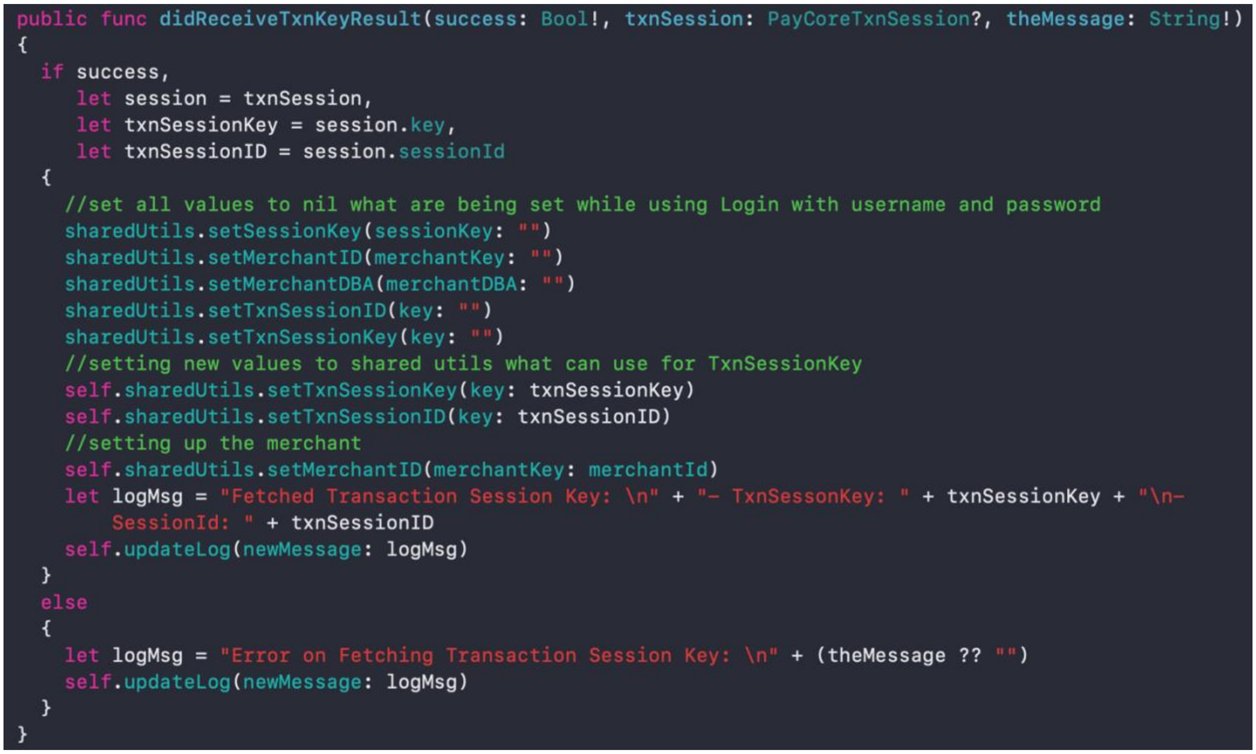
6. **Reverse-Auth (Reversal) / Refund Transaction** : Utilizing `useTxnSession`, developers can decide whether to employ TxnSessionKey. Depending on the chosen method, `payTxnSessionKey` or `paySessionKey` should be set to nil or assigned values as needed.

Authentication

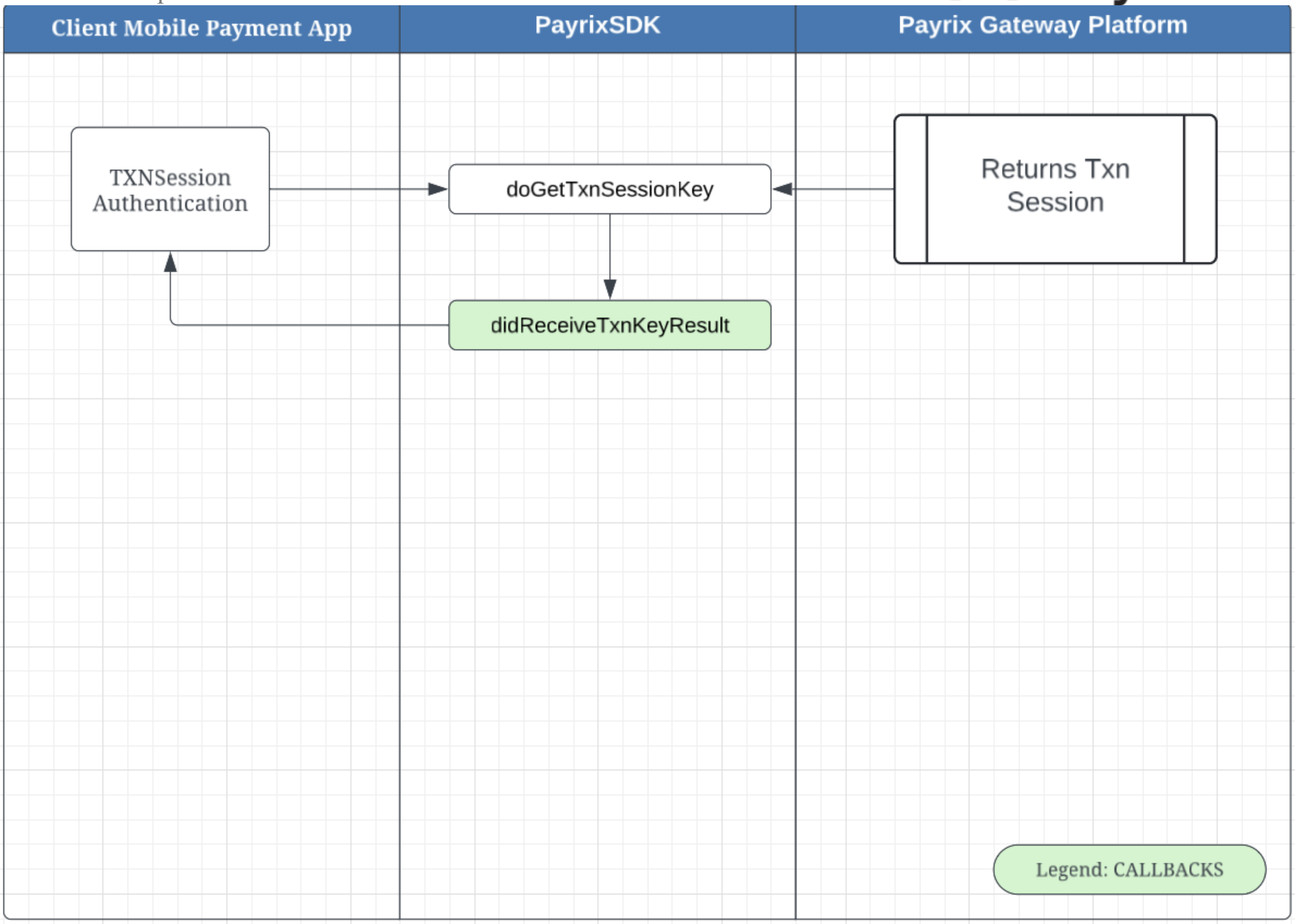
The flow is very similar to Login session, To initiate Authentication, developers must utilize the TxnSessionConfiguration class. The callback function, `didReceiveTxnKeyResult`, delivers success status, a PayCoreTxnSession object, and server messages.



```
let configuration = TxnSessionConfiguration()
configuration.duration = 3000// The number of times this key can be used for requests. Default is 8.
configuration.maxTimesApproved = 200// The maximum number of approved transactions that can be
    associated with this key. Default is 4.
configuration.maxTimesUse = 100 // The time in minutes the key remains valid; it's automatically
    voided when expired. Default is 10.

payrixSDK.delegate = self
payrixSDK.doGetTxnSessionKey(
  apiKey: APIKey,
  merchantID: merchantId,
configuration: configuration)
```

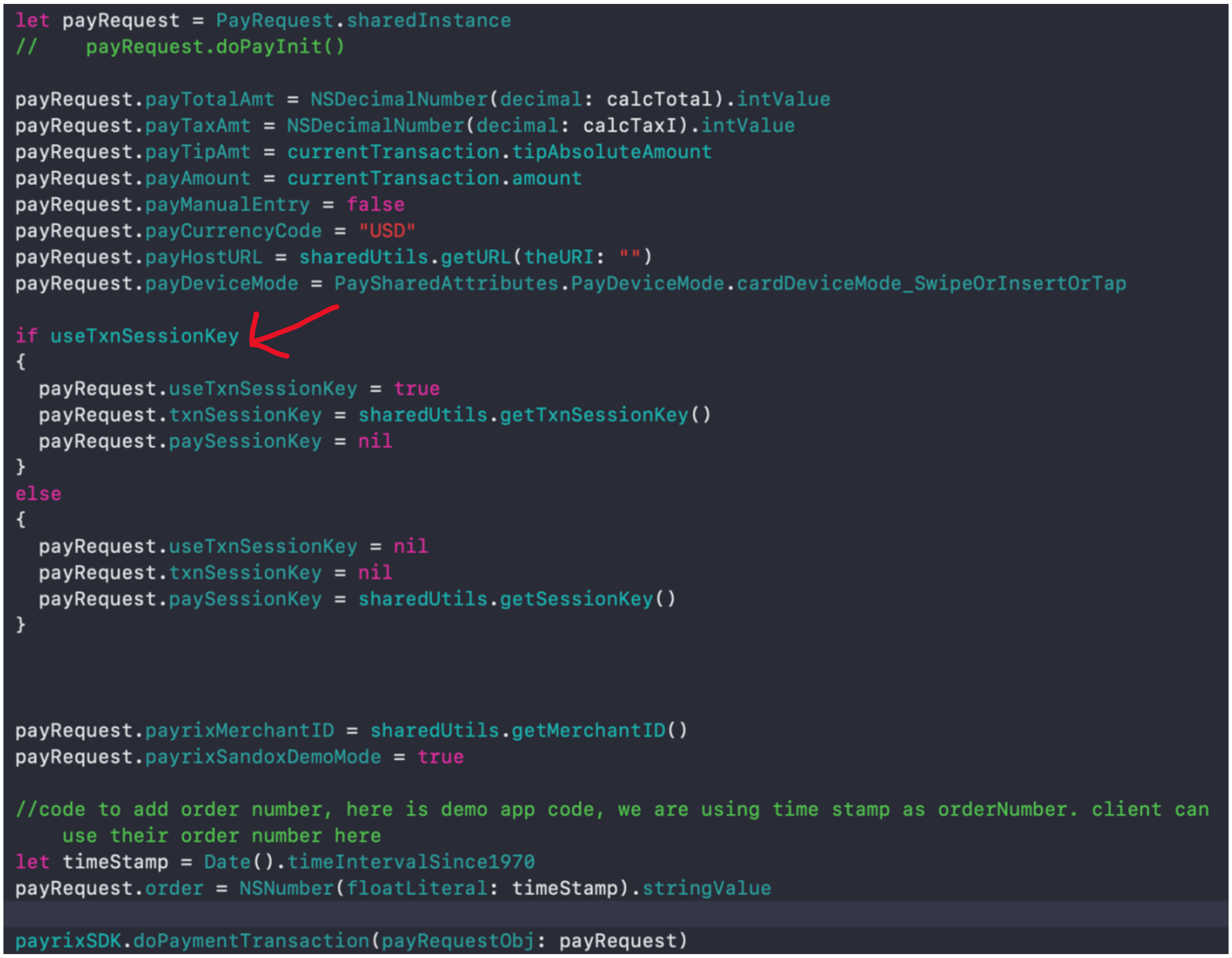


```
public func didReceiveTxnKeyResult(success: Bool!, txnSession: PayCoreTxnSession?, theMessage: String!)
{
  if success,
     let session = txnSession,
     let txnSessionKey = session.key,
     let txnSessionID = session.sessionId
  {
    //set all values to nil what are being set while using Login with username and password
    sharedUtils.setSessionKey(sessionKey: "")
    sharedUtils.setMerchantID(merchantKey: "")
    sharedUtils.setMerchantDBA(merchantDBA: "")
    sharedUtils.setTxnSessionID(key: "")
    sharedUtils.setTxnSessionKey(key: "")
    //setting new values to shared utils what can use for TxnSessionKey
    self.sharedUtils.setTxnSessionKey(key: txnSessionKey)
    self.sharedUtils.setTxnSessionID(key: txnSessionID)
    //setting up the merchant
    self.sharedUtils.setMerchantID(merchantKey: merchantId)
    let logMsg = "Fetched Transaction Session Key: \n" + "- TxnSessonKey: " + txnSessionKey + "\n-
        SessionId: " + txnSessionID
    self.updateLog(newMessage: logMsg)
  }
  else
  {
    let logMsg = "Error on Fetching Transaction Session Key: \n" + (theMessage ?? "")
    self.updateLog(newMessage: logMsg)
  }
}
```


Client Mobile Payment App
PayrixSDK
Payrix Gateway Platform
TXNSession Authentication
doGetTxnSessionKey
Returns Txn Session
didReceiveTxnKeyResult
Legend: CALLBACKS

Payment Transaction

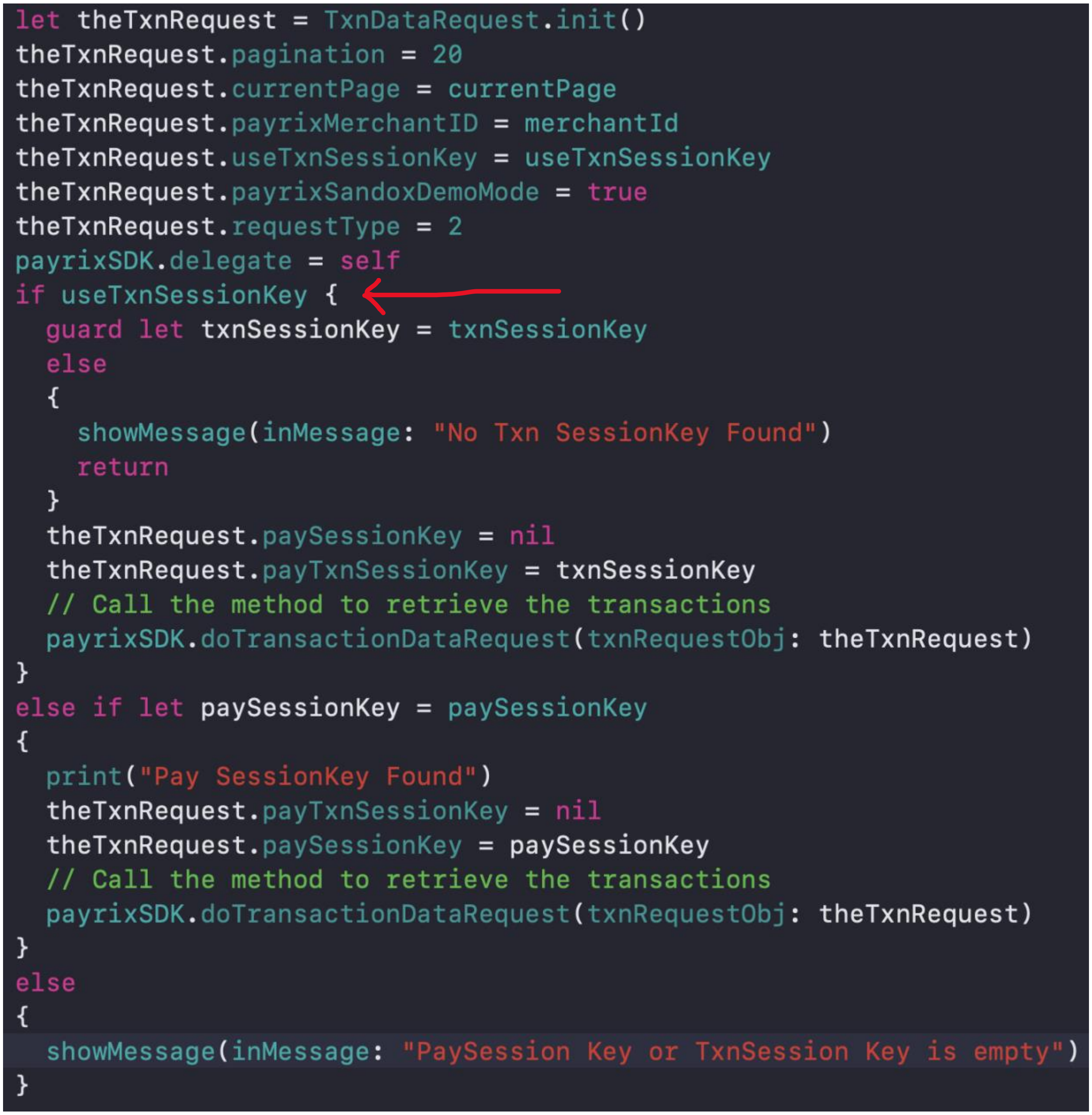
Tell PayRequest what type of auth to use. See below for how to implement this before calling `payrixSDK.doPaymentTransaction(payRequest);`



```
let payRequest = PayRequest.sharedInstance
//    payRequest.doPayInit()

payRequest.payTotalAmt = NSDecimalNumber(decimal: calcTotal).intValue
payRequest.payTaxAmt = NSDecimalNumber(decimal: calcTaxI).intValue
payRequest.payTipAmt = currentTransaction.tipAbsoluteAmount
payRequest.payAmount = currentTransaction.amount
payRequest.payManualEntry = false
payRequest.payCurrencyCode = "USD"
payRequest.payHostURL = sharedUtils.getURL(theURI: "")
payRequest.payDeviceMode = PaySharedAttributes.PayDeviceMode.cardDeviceMode_SwipeOrInsertOrTap

if useTxnSessionKey
{
  payRequest.useTxnSessionKey = true
  payRequest.txnSessionKey = sharedUtils.getTxnSessionKey()
  payRequest.paySessionKey = nil
}
else
{
  payRequest.useTxnSessionKey = nil
  payRequest.txnSessionKey = nil
  payRequest.paySessionKey = sharedUtils.getSessionKey()
}



payRequest.payrixMerchantID = sharedUtils.getMerchantID()
payRequest.payrixSandoxDemoMode = true

//code to add order number, here is demo app code, we are using time stamp as orderNumber. client can
    use their order number here
let timeStamp = Date().timeIntervalSince1970
payRequest.order = NSNumber(floatLiteral: timeStamp).stringValue

payrixSDK.doPaymentTransaction(payRequestObj: payRequest)
```

Fetching Transaction List

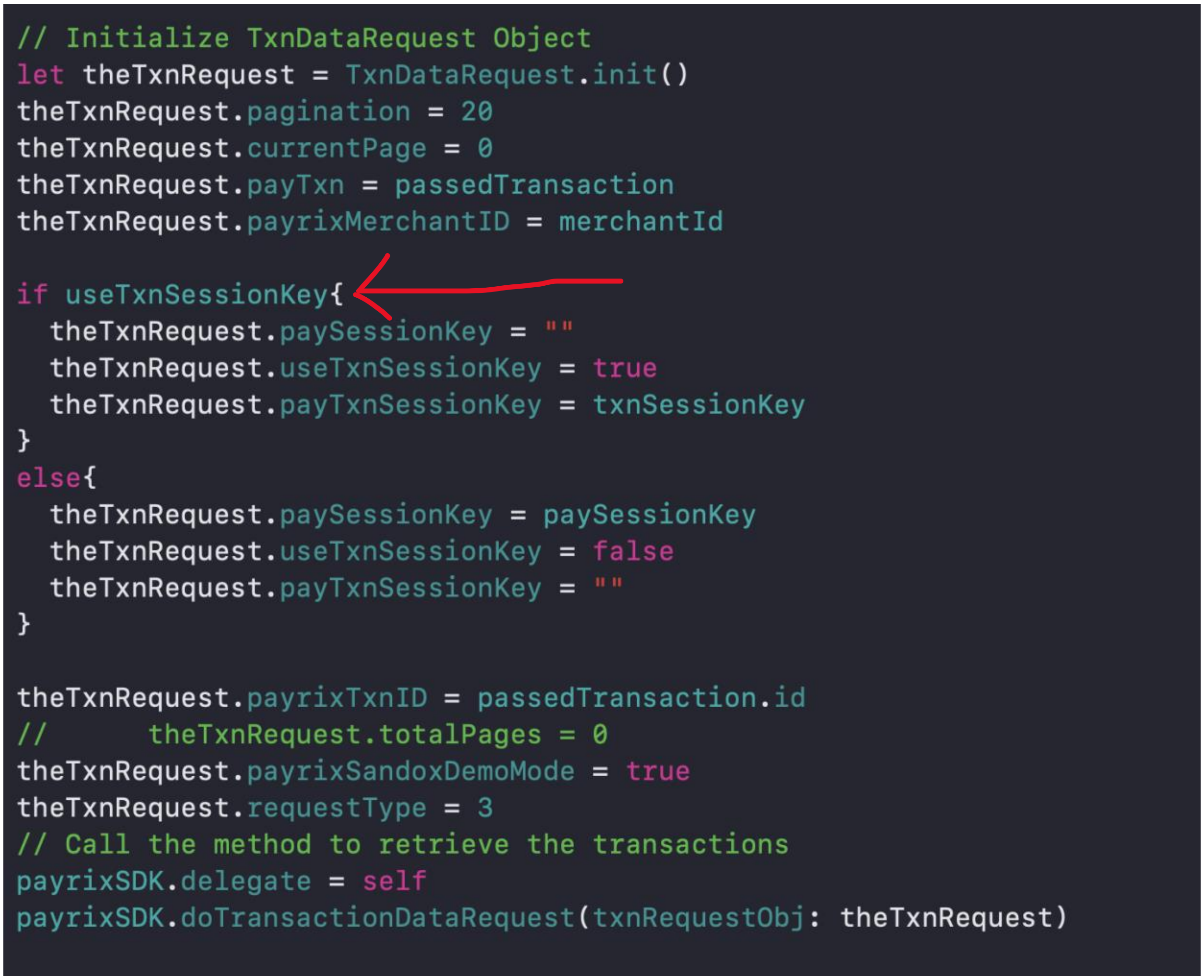
The method for fetching transaction lists requires parameters indicating TxnSessionKey or PaySessionKey usage. Developers should set `payTxnSessionKey` accordingly, ensuring consistency in parameter assignment.



```
let theTxnRequest = TxnDataRequest.init()
theTxnRequest.pagination = 20
theTxnRequest.currentPage = currentPage
theTxnRequest.payrixMerchantID = merchantId
theTxnRequest.useTxnSessionKey = useTxnSessionKey
theTxnRequest.payrixSandoxDemoMode = true
theTxnRequest.requestType = 2
payrixSDK.delegate = self
if useTxnSessionKey {
  guard let txnSessionKey = txnSessionKey
  else
  {
    showMessage(inMessage: "No Txn SessionKey Found")
    return
  }
  theTxnRequest.paySessionKey = nil
  theTxnRequest.payTxnSessionKey = txnSessionKey
  // Call the method to retrieve the transactions
  payrixSDK.doTransactionDataRequest(txnRequestObj: theTxnRequest)
}
else if let paySessionKey = paySessionKey
{
  print("Pay SessionKey Found")
  theTxnRequest.payTxnSessionKey = nil
  theTxnRequest.paySessionKey = paySessionKey
  // Call the method to retrieve the transactions
  payrixSDK.doTransactionDataRequest(txnRequestObj: theTxnRequest)
}
else
{
  showMessage(inMessage: "PaySession Key or TxnSession Key is empty")
}
```

Subsequent Transactions

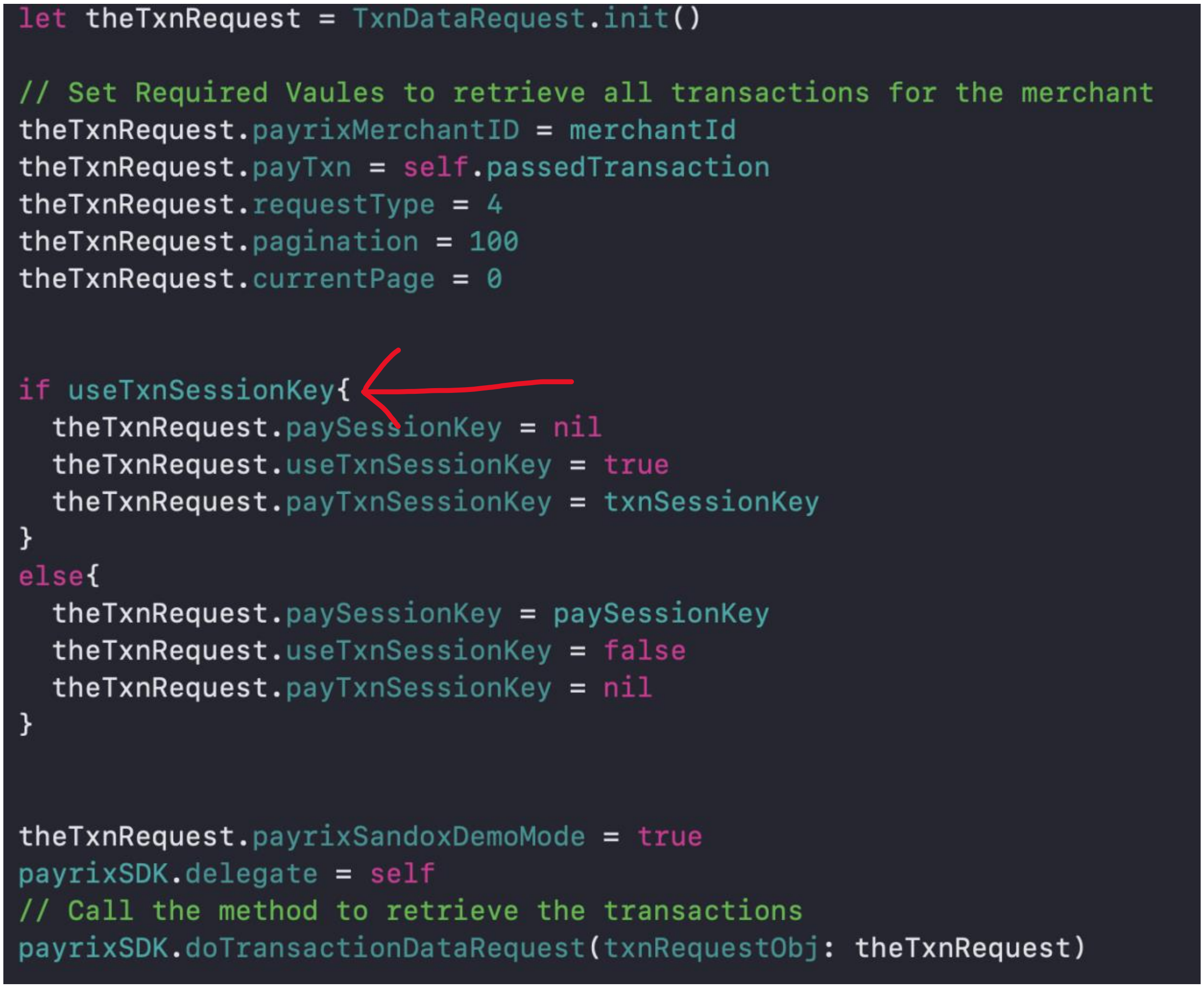
Similar to other methods, developers can specify `useTxnSession` and set `paySessionKey` or `payTxnSessionKey` based on requirements for subsequent transactions.



```
// Initialize TxnDataRequest Object
let theTxnRequest = TxnDataRequest.init()
theTxnRequest.pagination = 20
theTxnRequest.currentPage = 0
theTxnRequest.payTxn = passedTransaction
theTxnRequest.payrixMerchantID = merchantId

if useTxnSessionKey{
   theTxnRequest.paySessionKey = ""
   theTxnRequest.useTxnSessionKey = true
   theTxnRequest.payTxnSessionKey = txnSessionKey
}
else{
   theTxnRequest.paySessionKey = paySessionKey
   theTxnRequest.useTxnSessionKey = false
   theTxnRequest.payTxnSessionKey = ""
}

theTxnRequest.payrixTxnID = passedTransaction.id
//       theTxnRequest.totalPages = 0
theTxnRequest.payrixSandoxDemoMode = true
theTxnRequest.requestType = 3
// Call the method to retrieve the transactions
payrixSDK.delegate = self
payrixSDK.doTransactionDataRequest(txnRequestObj: theTxnRequest)
```

Refund Eligibility Check

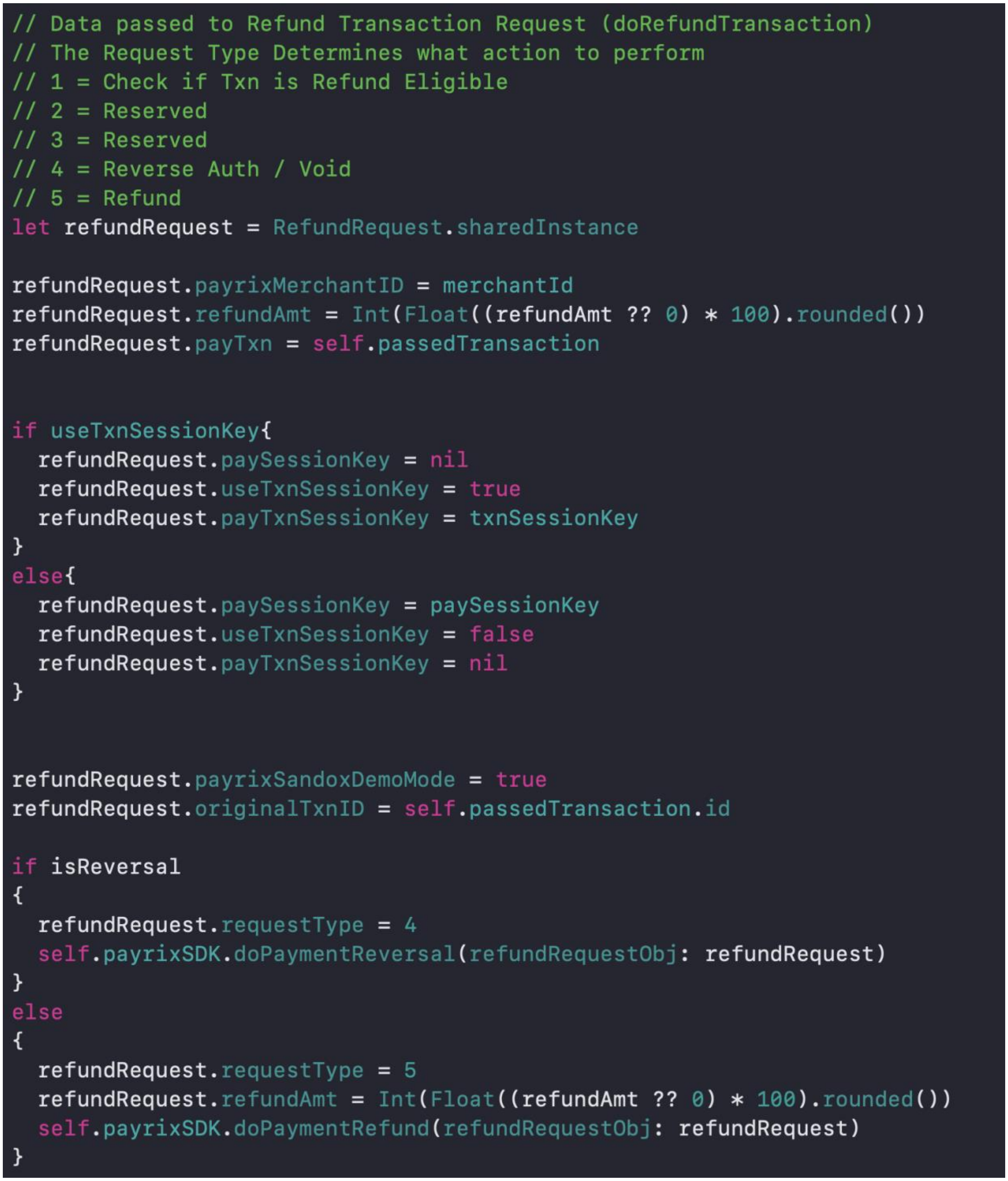
To verify refund eligibility, developers can set `useTxnSessionKey` and configure `paySessionKey` and `payTxnSessionKey` accordingly.



```
let theTxnRequest = TxnDataRequest.init()

// Set Required Vaules to retrieve all transactions for the merchant
theTxnRequest.payrixMerchantID = merchantId
theTxnRequest.payTxn = self.passedTransaction
theTxnRequest.requestType = 4
theTxnRequest.pagination = 100
theTxnRequest.currentPage = 0


if useTxnSessionKey{
  theTxnRequest.paySessionKey = nil
  theTxnRequest.useTxnSessionKey = true
  theTxnRequest.payTxnSessionKey = txnSessionKey
}
else{
  theTxnRequest.paySessionKey = paySessionKey
  theTxnRequest.useTxnSessionKey = false
  theTxnRequest.payTxnSessionKey = nil
}


theTxnRequest.payrixSandoxDemoMode = true
payrixSDK.delegate = self
// Call the method to retrieve the transactions
payrixSDK.doTransactionDataRequest(txnRequestObj: theTxnRequest)
```

Reverse-Auth (Reversal) / Refund Transaction:

Use payTxnSessionKey or paySessionKey based on requirements.



```
// Data passed to Refund Transaction Request (doRefundTransaction)
// The Request Type Determines what action to perform
// 1 = Check if Txn is Refund Eligible
// 2 = Reserved
// 3 = Reserved
// 4 = Reverse Auth / Void
// 5 = Refund
let refundRequest = RefundRequest.sharedInstance

refundRequest.payrixMerchantID = merchantId
refundRequest.refundAmt = Int(Float((refundAmt ?? 0) * 100).rounded())
refundRequest.payTxn = self.passedTransaction


if useTxnSessionKey{
  refundRequest.paySessionKey = nil
  refundRequest.useTxnSessionKey = true
  refundRequest.payTxnSessionKey = txnSessionKey
}
else{
  refundRequest.paySessionKey = paySessionKey
  refundRequest.useTxnSessionKey = false
  refundRequest.payTxnSessionKey = nil
}


refundRequest.payrixSandoxDemoMode = true
refundRequest.originalTxnID = self.passedTransaction.id

if isReversal
{
  refundRequest.requestType = 4
  self.payrixSDK.doPaymentReversal(refundRequestObj: refundRequest)
}
else
{
  refundRequest.requestType = 5
  refundRequest.refundAmt = Int(Float((refundAmt ?? 0) * 100).rounded())
  self.payrixSDK.doPaymentRefund(refundRequestObj: refundRequest)
}
```

Object Classes

Important Note:

The following objects are subject to change and new ones added before this document may be updated. The actual SDK and the inline documentation is the best reference. Payrix is committed to maintaining this document as well

PayRequest

Element Name	Data Type	Comment
payTotalAmt	Double	The Total amount (includes Tax and Tip)
payTaxAmt	Double	A Tax amount in decimal form (6.75)
payTipAmt	Double	A Tip amount in decimal form (1.05)
payCurrencyCode	String	A Universal Currency Code (Ex: USD)
payHostURL	String	Payrix Host URL
payDeviceMode	PaySharedAttributes. PayDeviceMode	Swipe, Insert, Tap, or some combination (See definition for full list)
paySessionKey	String	A valid session key (or API Key) from a successful credentials login
payrixMerchantID	String	A valid Payrix Merchant ID

payrixSandoxDemoMode	Bool	An Indicator: True = Sandbox Demo mode;
		False = Live - Production Mode

payManualEntry	Bool	An Indicator: True = Manual Card Data Entry;
		False = Card Reader Capture (Default)

useTxnSessionKey	Bool	True = Use TxnSessionKey, False = Means it should fallback to Login Session
txnSessionKey	String	A valid txn session key
Following Fields Required for Manual Entry Only		
payAmount	Double	The Amount (Cost) without Tax or Tip
payTaxPercent	Double	A Tax Rate (Percent)
payTipPercent	Int	A Tip Percent Amount
payCardHolder	String	The Card Holder Name
payCCNumber	String	The Card Number
payCardType	PaySharedAttributes .CCType	The Card Type (Brand)
payCardCVV	String	The CVV Security Code
payCardExp	String	The Card Expiration Date
payOrigin	PaySharedAttributes .PayTxnOrigin	The Transaction Origin: ECommerce, CreditCardTerminal, etc
payAddress1	String	
payAddress2	String	
payCity	String	
payStateProvince	String	
payPostalCodeZip	String	The Card Holder Billing Zip Code

TxnDataRequest

Element Name	Data Type	Comment
requestType	Integer	Request Type
payHostURL	String	Payrix Host URL
paySessionKey	String	A valid session key from a successful credentials login
useTxnSessionKey	Boolean	Txn session key retrieve after a successful auth
payTxnSessionKey	String	Txn Session Key
payrixMerchantID	String	A valid Payrix Merchant ID
payrixSandoxDemoMode	Boolean	An Indicator: True = Sandbox Demo mode; False = Live - Production Mode
payrixTxnID	String	The Txn ID being requested
payTxn	PayCoreTxn	The Txn being requested
txns	Txn	The transaction to be processed
pagination	Integer	The number of transactions per page
currentPage	Integer	The current page retrieved
totalPages	Integer	The total pages retrieved
morePages	Boolean	More pages available to retrieve

TxnSessionConfig

Method Name	Arguments	Comment
setDuration	Int duration	How many minutes this key should live
setMaxTimesApproved	Int maxTimesApproved	It’s related to how many approved transactions related it can be used
setMaxTimesUse	Int maxTimesUse	How many times a request can be made with this key

Text in red are the newly added variables

Error Message

During transaction you might notice the below error:

An Unexpected Error Occurred: Msg: Error: We couldn't recognize this auth, check and try again.

This simply means the Txn Session as expired, so you will have to generate another by calling TxnSessionConfig class.