IboxPro API
.NET integration guide
V 1.2.0

Version control

Version	Date	Description
1.0.0	12.04.2016	Initial version
1.0.1	20.04.2016	Fixes: - Transaction.Card, - PaymentCancelled event call. ReverseEvent.CantReverse event has been added
1.0.2	30.05.2016	enum Currency is added, rounding errors are fixed
1.0.3	04.08.2016	Partial reversals/refunds are added. EMV payments procedure is actualized, one-factor authorization is added. Transaction class properties are added
1.0.4	11.08.2016	Cash payment is added
1.1.0	09.12.2016	Framework is changed to .NET4.5. Card readers Wisepad_2 and QPos_mini are added.
1.2.0	21.03.2017	Receipt generation is added
1.3.0	10.05.2017	NFC support for reader P17 Names of readers have been changed

Table of contents

Introduction

Namespace Ibox.Pro.SDK.External

Enumerations (enums)

ReaderType

The set of supported reader types

Type	Description
P15	Card reader «Chip&Pin», P15
P16	Card reader «Chip&Sign», P16
P17	Card reader « Chip&Pin NFC », P17

ReaderEvent

The set of events that can be sent to card readers

Type	Description
Connected	Card reader is connected
Disconnected	Card reader is disconnected
Startinit	Start of initialization
InitSuccessfully	Initialization has been completed successfully
InitFailed	Initialization error
EjectCardTimeout	Not used
SwipeCard	Magnetic stripe transaction is detected
TransactionStarted	EMV transaction is started
WaitingForCard	Waiting for magnetic stripe or EMV contact transaction
PaymentCanceled	Payment is cancelled by user
EjectCard	Card can be ejected (generated by error during transaction processing)
BadSwipe	Can’t read magnetic stripe
LowBattery	Battery charge level of reader is less than 10%

PaymentError

The set of possible errors for payment processing

Type	Description
ConnectionError	Server connection error
ServerError	Transaction processing error
PaymentRuntimeError	Payment processing error
TransactionNullOrEmpty	Transaction generation error
EmvError	
EmvTerminated	
EmvDeclined	
Type	**Description**
EmvCancel	

EmvCardError	
EmvCardBlocked	
EmvDeviceError	
EmvCardNotSupported	
EmvZeroTranEmv	
EmvNotAllowed	EMV transaction is not allowed

ReverseEvent

Events of status change for Reversal (Refund) transaction

Type	**Description**
TransactionNotFound	Transaction is not found or not unique
SwitchedToCNP	Transaction reversal will be implemented in CNP mode
CantReverse	Reversal(Refund) is not allowable for this transaction

Currency

Payment currencies

Type	**Description**
RUB	Russian ruble
VND	Vietnamese dong

Class PaymentController

This is the main class of library. It contains methods for processing of Purchase/Reversal (Refund) transactions and for passing additional parameters, it also encapsulates all actions with card readers. Also this class is used to transfer additional payment data and receive payments history.

User's email and password must be defined before transaction processing, they are required for authentication with **SetCredentials** method, card reader type must also be defined before transaction processing with method **SetReaderType** (otherwise exception **InvalidOperationException** will be thrown**)**. Method **Enable** call is also mandatory. Call of method **Disable** will switch off card reader but **NOT** stop current transaction processing**.** If properties **SelectApplicationDelegate**, **ConfirmScheduleDelegate** and **ScheduleCreationFailedDelegate** are not defined before payment processing or method **Enable** is not called then program generates **InvalidOperationException exception**. If decimal part of transaction amount is wider than possible decimal part for currency of transaction then decimal part will be **truncated**.

Properties:

Name	Description
Instance	Class instance
IsPaymentInProgress	**true** if payment is in progress
SinglestepEMV	Sign of one-factor authorization
SelectApplicationDelegate	Card application selection handler
ConfirmScheduleDelegate	Recurrent payment confirmation handler
ScheduleCreationFailedDelegate	Schedule creation handler for retry request

Events:

Name	Description
TransactionStartedEvent	The beginning of transaction processing
TransactionFinishedEvent	Transaction is completed successfully
ReaderEvent	Card reader state is changed
ErrorEvent	Payment processing error
ReverseEvent	State of Reversal (Refund) transaction has been changed

Delegates:

SelectApplicationCallback

Signature	Int SelectApplicationCallback(List<string> apps)
Input parameters	apps – list of application names
Returned value	Index number of selected application (starts from 0)
Description	Is called for EMV transaction with multiple-applications card

ConfirmScheduleCallback

Signature	bool ConfirmScheduleCallback(List<KeyValuePair<DateTime, decimal>> steps, decimal totalAmount)
Input parameters	steps – the list of steps for schedule execution, consists of pairs <Charge date, charge amount> totalAmount – total amount for all days
Returned value	Approval of payer about schedule validity
Description	Is called during recurrent payment creation

ScheduleCreationFailedCallback

Signature	bool ScheduleCreationFailedCallback(PaymentError error, string description = null)
Input parameters	error – error type description – error message. Is used only for cases where error == SERVER_ERROR
Returned value	**true** to repeat an effort of schedule creation
Description	Is called when error during schedule creation for recurrent payment happens

TransactionStartedEventHandler

Signature	void TransactionStartedEventHandler(string transactionID)
Input parameters	transactionID – ID of transaction
Returned value	No returned value
Description	Event processor for **TransactionStartedEvent**

TransactionFinishedEventHandler

Signature	void TransactionStartedEventHandler(PaymentResultContext result)
Input parameters	result – transaction data in **PaymentResultContext** view
Returned value	No returned value
Description	**TransactionFinishedEvent** event handler

ReaderEventHandler

Signature	void ReaderEventHandler(ReaderEvent readerEvent)
Input parameters	readerEvent – event of card reader

Returned value	No returned value
Description	**ReaderEvent** event handler

ErrorEventHandler

Signature	void ErrorEventHandler(PaymentError error, string description = null)
Input parameters	error – thrown error description – error description
Returned value	No returned value
Description	**ErrorEvent** event handler

ReverseEventHandler

Signature	void ReverseEventHandler(ReverseEvent reverseEvent, string description = null)
Input parameters	event description
Returned value	No returned value
Description	**ReverseEvent** event handler

Class methods:

Enable

Signature	void Enable()
Input parameters	No input parameters
Returned value	No returned value
Description	Starts work with card reader

Disable

Signature	void Disable()
Input parameters	No input parameters
Returned value	No returned value
Description	Finalizes work with card reader

SetCredentials

Signature	void SetCredentials(string email, string password)
Input parameters	email – email of user password – password of user
Returned value	No returned value
Description	Sets user's credentials

SetReaderType

Signature	void SetReaderType(ReaderType readerType, string readerBTPort = null)

Input parameters	readerType – type of card reader readerBTPort – COM port name that is used for reader connection.
Returned value	No returned value
Description	Changes type of current card reader. **InvalidOperationException** will be thrown if there will be an effort to change reader type during transaction processing

GetReaderType

Signature	ReaderType? GetReaderType()
Input parameters	No input parameters
Returned value	Type of current card reader
Description	Returns current type of card reader

StartPayment

Signature	void StartPayment(PaymentContext paymentContext)
Input parameters	paymentContext – payment data
Returned value	No returned value
Description	Starts payment. It's required to define user data and properties **SelectApplicationDelegate**, **ConfirmScheduleDelegate**, **ScheduleCreationFailedDelegate** and make sure that reader type is defined and reader is connected before method call, otherwise **InvalidOperationException** exception will be thrown. An attempt to start new payment (reversal) during the current payment processing will cause an error with **InvalidOperationException** exception.

StartReverse

Signature	void StartReverse(string transactionID, ReverseMode mode, decimal? amountToReverse)
Input parameters	transactionID – ID of cancelled payment transaction mode – cancellation type amountToRverse – amount to be reversed. For full reversal/refund must be **null**
Returned value	No returned value
Description	Starts payment reversal. It's required to define user data and make sure that reader type is defined and reader is connected before method call, otherwise **InvalidOperationException** exception will be thrown. An attempt to start new payment (reversal) during the current payment processing will cause an error with **InvalidOperationException** exception.

Adjust

Signature	APIResult Adjust(string transactionID, string email, string phone)
Input parameters	transactionID – transaction ID to send additional data email – email to send receipt phone – phone number to send receipt
Returned value	Result
Description	Is used to send receipt and signature for single payment

AdjustRegular

Signature	APIResult AdjustRegular(string transactionID, string email, string phone)
Input parameters	transactionID – transaction ID to send additional data email – email to send receipt phone – phone number to send receipt
Returned value	Result
Description	Is used to send receipt and signature for recurrent payment

AdjustReverse

Signature	APIResult AdjustReverse(string transactionID, string email, string phone)
Input parameters	transactionID – transaction ID to send additional data email – email to send receipt phone – phone number to send receipt
Returned value	Result
Description	Is used to send receipt and signature for reversal (refund)

GetHistory

Signature	APIGetHistoryResult GetHistory(int page)
Input parameters	page – page number
Returned value	Object APIGetHistoryResult containing set of transactions
Description	Provides transactions history by pages

Namespace Ibox.Pro.SDK.External.Context

Enumerations(enums)

RepeatType

The set of acceptable types for recurrent payment

Type	Description
DelayedOnce	Payment will be implemented only once
Weekly	Weekly payment
Monthly	Monthly payment
Quarterly	Quarterly payment
Annual	Annual payment
ArbitraryDays	Payment will be implemented in predefined days

EndType

The set of acceptable modes to end recurrent payment

Type	Description
ByQuantity	End by number of repetitions
AtDay	End in defined day

ReverseMode

The set of acceptable types for payment reversal

Type	Description
Cancel	Payment reversal
Return	Payment refund
CancelCNP	Cancellation in CNP mode

InputType

The set of acceptable payment types

Type	Description
Swipe	Payment with magnetic stripe
Chip	Payment with chip
Cash	Cash payment

Class PaymentContext

Container of data required for single payment processing.

Class properties:

Name	Description
Amount	Transaction amount
Currency	Transaction currency
Description	Transaction description
Image	Image attached to transaction
PaymentProductCode	Payment product code
PaymentProductTextDictionary	The set of values for text fields of product, format: <Field code, value>
PaymentProductImageDictionary	The set of images for graphic fields of product, format: <Field code, Image>
Cash	Cash sign

Class methods:

Clear

Signature	Clear()
Input parameters	No input parameters
Returned value	No returned value
Description	Clears object fields

Class RegularPaymentContext

Extension of class **PaymentContext** additionally contains properties for recurrent payment creation. For payment processing in the last day of month it's required to set property **DayOfWeek** with constant value **LAST_DAY_OF_MONTH**.

Class properties:

Name	Description
PaymentRepeatType	Recurrent payment type
PaymentEndType	Method to end recurrent payment processing
StartDate	Start date for recurrent payment
EndDate	End date for recurrent payment (if the end is configured for date)
RepeatCount	Recurrent payment executions limit (if executions count is defined for recurrent payment)
ArbitraryDays	Days defined for payment implementation (for payment type with defined dates)
Month	Month for payment, [1,12]
MonthOfQuarter	Month for quarterly payment, [1,4]
Day	Day for payment, [1,31]
DayOfWeek	Day of week for payment, [0,7], 0 – Sunday)
Hour	Hour for payment
Minute	Minute for payment
ReceiptEmail	Email for receipt sending
ReceiptPhone	Phone number for receipt sending

The set of mandatory properties depends on payment type:

Payment type	Set of properties
Never	StartDate
Weekly	StartDate, (EndDate или RepeatCount)
Monthly	StartDate, (EndDate или RepeatCount), Day
Quarterly	StartDate, (EndDate или RepeatCount), MonthOfQuarter, Day
Annual	StartDate, (EndDate или RepeatCount), Month, Day
ArbitraryDays	ArbitraryDays

Parameters RepeatType, EndType, ReceiptEmail, ReceiptPhone are mandatory for all types of recurrent payments.

Parameters Hour, Minute are optional for all types of recurrent payments.

Class PaymentResultContext

Container of data received for successful payment or reversal (refund).

Class properties:

Name	Description
TransactionItem	Data of payment/reversal transaction in **TransactionItem** view
ScheduleItem	Data of recurrent payment transaction in **ScheduleItem** view
RequiresSignature	Attribute of necessity for signature
TerminalName	Terminal
EmvData	Set of transaction EMV data in **HashMap<String, String>**

Namespace Ibox.Pro.SDK.External.Entry

Enumerations(enums)

InputType

The set of acceptable payment types

Type	Description
SWIPE	Payment with magnetic stripe
CHIP	Payment with chip of card
CASH	Cash payment

DisplayMode

Type of transaction display

Type	Description
Declined	Declined transaction
Success	Successful transaction
Reverse	Reversal (Refund)
Reversed	Payment is reversed (refunded)
NonFinancial	

Class Transaction

Transaction presentation in the terms of objects, contains the set of transaction properties.

Class properties:

Name	Description
ID	Transaction ID
Date	Transaction date and time according to GMT of device
Description	Transaction description
Invoice	Receipt number
Terminal	Terminal name
AcquirerApprovalCode	Transaction approval code
ScheduleID	Recurrent payment ID
ScheduleStepID	Charge ID for recurrent payment
Amount	Transaction amount
InputType	Payment method in **InputType**
Operation	Operation name
Latitude	Transaction latitude
Longitude	Transaction longitude
HasPhoto	Attribute of attached image presence
PhotoUrl	URL of attached image
RequiresSignature	Attribute of necessity for signature. False for PIN based transaction
HasSignature	Attribute of attached signature presence
SignatureUrl	URL of attached signature
StateDisplay	Transaction state description
CardholderName	Cardholder name
Card	Card data that was used for payment in view **Card**
EMVData	List of EMV tags to be printed in receipt
CanCancel	Ability to make reversal
CanReturn	Ability to make refund
CanCancelPartial	Ability to make partial reversal
CanReturnPartial	Ability to make partial refund
Canceled	Sign of reversal/refund for purchase
DisplayMode	Type of display mode for transaction in **DisplayMode**
SubstateDisplay	Substate transaction description

Class Card

Contains card data.

Class properties:

Name	**Description**
IIN	Type “card” or “cash”(for cash transaction)
BIN	Bank identification number
EXP	Card expiration date
PANMasked	First and last 4 digits of card number separated by symbol “*”
PANEnding	Last 4 digits of card number

Class methods:

IsCash

Signature	IsCash()
Input parameters	No input parameters
Returned value	**true** for cash payment
Description	Attribute of cash payment

Class Schedule

Object presentation for recurrent payment

Class properties:

Name	Description
ID	Recurrent payment ID
Card	Card data used in view **TransactionItem.Card** for payment

Namespace Ibox.Pro.SDK.External.Result

Class APIResult

Primitive entity, contains reply from server

Class properties:

Name	Description
ErrorCode	Error code. 0 – if reply doesn't contain error messages.
ErrorMessage	Error message

Class APIGetHistoryResult

Child class of **APIResult**. Contains the set of transactions received in reply for history request.

Class properties:

Name	Description
Transactions	Set of transactions contained in reply

Appendix 1: Receipt printing

Receipt data can be received in event onPaymentFinished
Requisites of client have to be taken form settings of source application.

Fields of receipt:

Name	Description
Bank	Requisites of client
Company name	
Legal entity name	
Company phone number	
Company WEB-site	
Operation date and time	result.TransactionItem.Date
Terminal name	result.TransactionItem.TerminalName
Receipt number	result.TransactionItem.Invoice
Approval code	result.TransactionItem.AcquirerApprovalCode
Card number and type	result.TransactionItem.Card.IIN, result.TransactionItem.Card.PANMasked
EMV tags of transaction	result.TransactionItem.EMVData, printed as key-value
Transaction type	result.TransactionItem.Operation
Transaction amount	result.TransactionItem.Amount
Fee	0.00 rur.
Status	Success
Signature	Area for signature if result.TransactionItem.RequiresSignature==true, otherwise « PIN is entered»

Receipt example:

```
VTB 24
Test client
JSC "Test client"
+7 916 111 2233
www.testclient.com
Transaction date and time: 21.03.2017 15:47:34
Terminal: II040001
Receipt: RM7ZEDMAAE7L
Approval code: SIMULATION
Card: mastercard  **** 5631
AID: A0000000041010
TSI: 6800
TVR: 8020008000
Transaction: Purchase
Total: 33 p
Fee: 0.00 p
Status: Success
PIN is entered
```