IboxPro API
integration guide
V 1.5.4.0

Version control

Version	Date	Description
1.0.0	26.05.2015	Initial version
1.1.0	09.07.2015	History of payments API added. Some fixes in library structure.
1.1.1	27.07.2015	ReaderEvent added. WAITING_FOR_CARD_CANCELLED for PaymentController. Some changes in TransactionItem, TaxItem.
1.2	11.08.2015	Added: ReaderEvent, PAYMENT_CANCELED, ReaderEvent.EJECT_CARD. Removed: ReaderEvent. WAITING_FOR_CARD_CANCELED, PaymentError.PAYMENT_IN_PROGRESS. Signature of some methods in PaymentController was changed.
1.3	10.09.2015	Partial reversals/refunds added. Getting of transaction by its ID added. PaymentResultContext, ScheduleItem were added. Some fixes of the internal logic made.
1.3.1	11.09.2015	Transaction interruption made. Some fixes of internal logic
1.3.2	14.09.2015	Fixes of conduction of the PaymentController and card reader during reversal/refund of the payment and reversal of the transaction. PaymentContext.currencyName property added. PaymentContext property removed.
1.3.3.	25.09.2015	New events BAD_SWIPE and LOW_BATTERY of the card reader added.
1.3.4	30.09.2015	PaymentControllerListener.onTransactionStarted() method added.
1.3.5	30.05.2016	enum Currency added. Fixes of problems with rounding of sums.
1.3.7	22.07.2016	Partial reversals/refunds added. Automatic configuration of the card-reader added. One-factorial authorization added. Fixes of internal logic
1.3.8	28.07.2016	Partial reversal/refund possibility attributes added.
1.3.9	02.08.2016	Properties of TransactionItem class added.
1.4.0	09.08.2016	Internal logic fixes

1.4.1	11.08.2016	Cash payments added. Sending of the data of the fiscal registrar added
1.4.2	18.08.2016	Callback to receive level of charge of a card-reader in the PaymentControllerListener interface added
1.4.3	29.09.2016	Internal logic fixes
1.5.0	18.11.2016	Support of the new card-readers and NFC transactions added. Improvement of the internal logic
1.5.1	01.12.2016	Regular payment logic fixes. Rounding errors fixes. Improvement of QPOS card-readers working. Cancellation of the prepaid payments added.
1.5.2	11.01.2017	Fixes of the connection card-reader QPOS_MINI by the USB and NFC payments. Improvements of the internal logic
1.5.3	16.01.2017	Support of the WISEPAD2_PLUS card-reader added. PaymentController.printText() method added. PaymentController.PrintResult enum added.
1.5.3.1	23.01.2017	CardholderName and TerminalName fields in TransacionItem added
1.5.3.5	16.03.2017	Internal logic fixes
1.5.3.8	30.03.2017	Support of the NFC payments for the QPOS Mini card-readers. Improvements of the internal logic
1.5.3.9	05.04.2017	Working of PaymentControllerListener.onSelectApplication() call fixed.
1.5.4.0	12.04.2017	Support of the M17 card-readers added. Names of the card-readers changed. Support of the unused card-readers removed. Support of the autoconfiguration removed. General improvements.

Table of contents

Introduction

Android Permissions

Before the work with a library, add these lines into AndroidManifest.xml file of the application:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.ACTION_HEADSET_PLUG" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Package ibox.pro.sdk.external

Class PaymentController

This class is main in the library. It contains methods that creates transactions and adds external parameters into them. In addition, it encapsulates working with card-readers. Class contains sets of different **enum** parameters that are necessary for making payment.

It is necessary to set users Email and Password before making payment. They needed to authenticate with **setCredentials** method and to set the type of reader with **setReaderType** method.

It is necessary to call **enable** method before the work with card-reader. In addition, it is necessary to call **disable** method after work with card-reader. Calling of disable method or turning off card-reader causes an interruption of the current payment processing.
For proper working, it is necessary to call **enable/disable** methods of instance of the class in call of parent **Activity** of the following methods: **onCreate, onDestroy, onSaveInstanceState**. There is a 5-second timeout for the extraction of the card on failure of the transaction.

PaymentControllerListener can be send to the instance of the class through the method **setPaymentControllerListener**, for the event handling of the card-reader and/or transaction process of implementation.

If the sum of the payment would be added with after-dot capacity bigger that an after-dot capacity of the current currency, number of symbols after dot would be cut without rounding.

Parameter sets:

ReaderType

Set of supported types of card-readers

Type	Description
P15	Card-reader ”Chip&Pin”, P15
P16PLUS	Card-reader ”Chip&Pin”, P16 Plus
M17	
P17	Card-reader “Chip&Pin NFC”, P17

ReaderEvent

Set of possible events that could be sent by the card-reader

Type	Description
CONNECTED	Card-reader connected
DISCONNECTED	Card-reader disconnected
START_INIT	Beginning of initialization

INIT_SUCCESSFULLY	Initialization completed successfully
INIT_FAILED	An error of initialization
EJECT_CARD_TIMEOUT	Not in use
SWIPE_CARD	Swipe of the magnet stripe detected
EMV_TRANSACTION_STARTED	EMV transaction began
NFC_TRANSACTION_STARTED	NFC transaction began
WAITING_FOR_CARD	Waiting for the swipe of the magnet stripe or insert of the EMV-card
PAYMENT_CANCELLED	Payment cancelled by the user
EJECT_CARD	Card can be removed (caused by the error during the transaction progress)
BAD_SWIPE	Failed to read data from the magnet stripe
LOW_BATTERY	Card-reader battery charge level is less than 10%

PaymentInputType

Set of possible types of payment

Type	Description
SWIPE	Payment by swipe of the magnet stripe
CHIP	Payment by EMV
NFC	NFC payment
CASH	Cash payment

PaymentError

Set of possible errors, that could appear in payment progress.

Type	Description
CONNECTION_ERROR	Failed to connect to the server
SERVER_ERROR	Failed to complete transaction
TRANSACTION_NULL_OR_EMPTY	Failed to create transaction
NO_SUCH_TRANSACTION	Transaction can’t be found or it is not unique
EMV_ERROR	General EMV error
EMV_TERMINATED	Transaction interrupted
EMV_DECLINED	Transaction rejected
EMV_CANCEL	Transaction reversed
EMV_CARD_ERROR	Card error
EMV_DEVICE_ERROR	Card-reader error
EMV_CARD_NOT_SUPPORTED	Card is not supported
EMV_ZERO_TRAN_EMV	Zero sum transaction attempted
EMV_NOT_ALLOWED	EMV transaction not permitted
NFC_NOT_ALLOWED	NFC transaction not permitted

RegularRepeatType

Set of possible types of regular payments

Type	Description
Never	Payment will be done only once
Weekly	Weekly payment
Monthly	Monthly payment
Quarterly	Quarterly payment
Annual	Annual payment
ArbitraryDays	Payment will be done in specified days

RegularEndType

Set of possible methods of the end of the regular payment accomplishment

Type	Description
BY_QUANTITY	Accomplishment by the number of repeats
BY_DAY	Accomplishment in the set day

ReverseAction

Set of possible methods for payment cancellation

Type	Description
CANCEL	Payment reversal
RETURN	Payment refund

Currency

Currency that can be used in payment

Type	Description
RUB	Rubles of the Russian Federation
VND	Dongs of the Vietnam

PrintResult

Set of possible results of print accomplishment

Type	Description
SUCCESS	Print successful
NO_PAPPER	Out of paper
WRONG_CMD	Wrong command
OVERHEAT	Overheat of the print-head
TIMEOUT	Waiting timeout exceed
PRINTER_ERROR	Printer error

Class methods:

getInstance

Signature	PaymentController getInstance()
Input parameters	No
Return value	Instance of the class
Description	Getting instance of the class method

onCreate

Signature	void onCreate(Context context, Bundle savedInstanceState)
Input parameters	**context** – context of the application **savedInstanceState** – transmitted from the parent method Activity
Return value	No
Description	Should be called when self-titled method of the parent **Activity** called

onDestroy

Signature	Void onDestroy()
Input parameters	No
Return value	No
Description	Should be called when self-titled method of the parent **Activity** called

onSaveInstanceState

Signature	void onSaveInstanceState(Context context, Bundle savedInstanceState)
Input parameters	savedInstanceState - transmitted from the parent method Activity
Return value	No
Description	Should be called when self-titled method of the parent **Activity** called

enable

Signature	void enable()
Input parameters	No
Return value	No
Description	Begins work with card-reader

disable

Signature	void disable()
Input parameters	No

Return value	No
Description	Ends work with card-reader

isConnected

Signature	Boolean isConnected()
Input parameters	No
Return value	true, if card-reader is connected
Description	Uses for checking readiness of card-reader

getBluetoothDevices

Signature	ArrayList<BluetoothDevice> getBluetoothDevices(Context context)
Input parameters	**context** – context of the application
Return value	**ArrayList** of paired devices
Description	Uses for getting set of Bluetooth-devices available to connect

setPaymentControllerListener

Signature	void setPaymentControllerListener(PaymentControllerListener listener)
Input parameters	**Listener** – event handler
Return value	No
Description	Sets new event handler of making payment

setCredentials

Signature	void setCredentials(String email, String password)
Input parameters	**email** – user email **password** – user password
Return value	No
Description	Sets user data that are needed to make transactions

setReaderType

Signature	void setReaderType(Context context, ReaderType readerType, String address, String config) throws IllegalStateException
Input parameters	**context** – application context **readerType** – type of card-reader **address** – MAC-adress of Bluetooth card-reader. Use config parameters of the card-

	reader called **PaymentController.USB_MODE_KEY** **config** to connect by USB.
Return value	No
Description	Changes the type of the current card-reader.

getReaderType

Signature	ReaderType getReaderType()
Input parameters	No
Return value	Current type of card-reader
Description	Returns current type of card-reader

startPayment

Signature	void startPayment(Context context, PaymentContext paymentContext) throws PaymentException
Input parameters	**context** – application context **paymentContext** – payment data
Return value	No
Description	Begins execution of the payment. **PaymentException** exception will be generated on try to start new payment/reversal

eversePayment

Signature	void reversePayment(Context context, String transactionID, ReverseAction action, Double amountToReverse, Currency currency) throws PaymentException
Input parameters	**context** – application context **transactionID** – transaction ID of reversal payment **action** – type of reversal **amountToReverse** – sum of reversal. Should send **NULL** for full reversal
Return value	No
Description	Begins execution of payment reversal. **PaymentException** will be generated on try to start new payment/reversal

adjust

Signature	APIResult adjust(Context context, int regularID, byte [] signature)
Input parameters	**context** – application context **transactionID** – ID of transaction that is

	needed an extra data to send to. **receiptPhone** – telephone number to send receipt to.
Return value	Data sending result
Description	Uses for single payment signature and receipt sending

adjust

Signature	APIResult adjust(Context context, int regularID, byte [] signature)
Input parameters	**context** – application context **regularID** – ID of transaction that is needed an extra data to send to. **signature** – an image with a signature of a payer in it
Return value	Data sending result
Description	Uses for a regular payment signature and receipt sending

adjustReverse

Signature	APIResult adjustReverse(Context String transactionID, String receiptPhone, String receiptEmail, byte [] signature)
Input parameters	**context** – application context **regularID** – ID of transaction that is needed an extra data to send to. **receiptPhone** – phone number for receipt sending **receiptEmail** – email-address for receipt sending **signature** - an image with a signature of a payer in it
Return value	Data sending result
Description	Uses for a payment reversal signature and receipt sending

isPaymentInProgress

Signature	boolean isPaymentInProgress()
Input parameters	No
Return value	**true**, if payment is not finished yet
Description	Uses for a controllers state check

getHistory

Signature	APIGetHistoryResult getHistory(Context context, int page)
Input parameters	**context** – application context

	page – page number
Return value	**APIGetHistoryResult** object that includes set of transactions
Description	Allows to get payment history in a page view

getTransactionByID

Signature	APIGetHistoryResult getTransactionByID(Context context, String transactionID)
Input parameters	**context** – application context **transactionID** – ID of requested transaction
Return value	**APIGetHistoryResult** object that includes requested transaction
Description	Allows to get transaction by ID

setSingleStepEMV

Signature	void setSingleStepEMV(boolean singleStepEMV)
Input parameters	**singleStemEMV** – one-factor authorization attribute
Return value	No
Description	Allows to make payments with one-factor authorization

setSingleStepEMV

Signature	boolean isSingleStepEMV()
Input parameters	No
Return value	One-factor authorization mode attribute
Description	Returns one-factor authorization mode attribute

submitFiscal

Signature	APIResult submitFiscal(Context context, String transactionID, int printerID, int docID, int CVC, int shift)
Input parameters	**context** – application context **transationID** – ID of transaction which fiscal data need to be sent **printerID** – fiscal printer ID **docID** – end-to-end document number **CVC** – document CVC **shift** – number of operating shift
Return value	One-factor authorization mode attribute
Description	Returns one-factor authorization mode attribute

printText

Signature	PrintResult printText(String text, Layout.Alignment alignment) throws IllegalStateException
Input parameters	**text** – text to print **alignment** – alignment of the text
Return value	Printing result
Description	The command is workin with card-reader **WISEPAD2_PLUS** only, otherwise **IllegalStateException** would be generated

Interface PaymentControllerListener

Class **PaymentController** callback-interface.

Methods of the interface:

onTransactionStarted

Signature	void onTransactionStarted(String transactionID)
Input parameters	**transactionID** – ID of the executable transaction
Return value	No
Description	Method would be called before payment processing. Calls only for a single(simple) payment.

onFinished

Signature	void onFinished(PaymentResultContext result)
Input parameters	**result** – completed transaction data
Return value	No
Description	Method would be called after successful payment or successful payment reversal.

onError

Signature	void onError(PaymentError error, String errorMessage)
Input parameters	**error** – error type **errorMessage** – error message. Uses only if error == SERVER_ERROR
Return value	No
Description	Method would be called if an error happens during the transaction progress.

onSelectApplication

Signature	int onSelectApplication(List<String> apps)
Input parameters	**apps** – list of application names
Return value	Serial number of chosen application (counting starts from 0)
Description	Method would be called during the EMV transaction if EMV-card has more than 1 application. Calling of the method happens not in a parent stream.

onConfirmSchedule

Signature	boolean onConfirmSchedule(List<Map.Entry<Date, Double>> steps, double totalAmount)
Input parameters	**steps** – list of scheduling steps that are consists of pairs <Write-off date, Write-off sum> **totalAmount** – all days total amount
Return value	Payer confirmation of correctness of scheduler attribute.
Description	Method would be called in regular payment progress. Calling of the method happens not in a parent stream.

onScheduleCreationFailed

Signature	boolean onScheduleCreationFailed(PaymentError error, String errorMessage)
Input parameters	**error** – error type **errorMessage** – error message. Uses only if error == SERVER_ERROR
Return value	true if it is necessary to repeat the schedule creation
Description	Method would be called if an error happens during the creation of the regular payment schedule

onCancellationTimeout

Signature	boolean onCancellationTimeout()
Input parameters	No
Return value	**true** for the payment refund
Description	Method would be called if there would be a try to refund a payment but the allowed time to refund is already over

onPinRequest

Signature	void onPinRequest()
Input parameters	No
Return value	No
Description	Method would be called by card-reader PIN request

onPinEntered

Signature	void onPinEntered()
Input parameters	No

Return value	No
Description	Method would be called after entering PIN-code

onBatteryState

Signature	void onBatteryState(double percent)
Input parameters	**percent** – card-reader charge level in percent
Return value	No
Description	Method would be called after card-reader initialization

Class PaymentContext

Necessary for a single payment JavaBean data container.

Class properties:

Name	Description
amount	Payment sum
currency	Payment currency
description	Payment description
transactionID	Not in use
image	Attached to the payment image
currency	Payment currency
cash	Cash payment attribute

Class methods:

reset

Signature	reset()
Input parameters	No
Return value	No
Description	Clears object fields

Class RegularPaymentContext

It is an extended version of the **PaymentContext** class that includes attributes that are necessary for a regular payment.
For execution of payment on the last day of a month, **dayOfWeek** property should have a value equal to **LAST_DAY_OF_MONTH** constant.

Class properties:

Name	Description
repeatType	Type of a regular payment
endType	Type of regular payment completion
startDate	Date of regular payment beginning
endDate	Date of regular payment ending (if it is by date)
repeatCount	Number of regular payments (if ending of it depends on number of repeats)
arbitraryDays	Payment specified days (if arbitrary payment type)
month	Payment month ([1,12] and [1,4] if repeatType == Quarterly)
day	Payment day ([1,31])
dayOfWeek	Payment day of week ([0,7] where 0 is Sunday)
hour	Payment hour
minute	Payment minute
receiptEmail	Receipt sending Email
receiptPhone	Receipt sending phone number

Set of necessary filled attributes depends on type of payment:

Payment type	Set of properties
Never	startDate
Weekly	startDate, (endDate or repeatCount)
Monthly	startDate, (endDate or repeatCount), day
Quarterly	startDate, (endDate or repeatCount), month, day
Annual	startDate, (endDate or repeatCount), month, day
ArbitraryDays	arbitraryDays

repeatType, endType, receiptType, recieptEmail, receiptPhone attributes are necessary for all types of regular payments.
hour and **minute** attributes are necessary for all types of regular payments.

Class PaymentResultContext

JavaBean container of received after successful payment of payment reversal data.

Class properties:

Name	Description
transactionItem	Payment/reversal data in the view of **TransactionItem**
scheduleItem	Regular payment data in the view of **ScheduleItem**
requiresSignature	Attribute of necessity to send signature of payer after payment
terminalName	Terminal
emvData	Set of EMV transaction data in the view of **HashMap<String, String>**

Class AbstractEntity

Abstract wrap-class for a data array in the view of JSON. Realizes Serializable method.

getJSON

Signature	JSONObject getJSON()
Input parameters	No
Return value	Set of data in JSON view
Description	Returns JSON set of data

Package ibox.sdk.external.entities

Class TransactionItem

Derived **AbstractEntity** class. Is an object representation of transaction. Contains a set of defining properties. Contains Card and Formant nested classes.

Class properties:

Name	Description
ID	Transaction ID
Date	Date and time of transaction according to GMT of device
Description	Transaction description
Invoice	Receipt number
ApprovalCode	Approval code
ScheduleID	Regular payment ID
ScheduleStepID	Recurrent payment write-off ID
Amount	Transaction sum
AmountEff	Transaction balance
InputType	Payment method in the view of InputType
Operation	Operation name
Latitude	Transaction location geographical latitude
Longitude	Transaction location geographical longitude
HasPhoto	Attached image attribute
PhotoUrl	Attached image URL
HasSignature	Attached signature attribute
SignatureUrl	Attached signature URL
StateDisplay	Transaction state description
Card	Payment card data in the view of TransactionItem.Card
CanCancel	Transaction reversal possibility attribute
CanReturn	Transaction refund possibility attribute
CanCancelPartial	Transaction partial reversal possibility attribute
CanReturnPartial	Transaction partial refund possibility attribute
DisplayMode	Transaction display type in the view of DisplayMode
SubstateDisplay	Transaction substate display
CardholderName	Cardholder name
TerminalName	Terminal

Set of arguments:

InputType

Set of possible payment methods

Type	Description
SWIPE	Magnet stripe swipe payment
CHIP	EMV-chip payment
NFC	NFC payment
CASH	Cash payment

DisplayMode

Transaction display type

Type	Description
DECLINED	Declined transaction
SUCCESS	Successful transaction
REVERSE	Reverse/refund transaction
REVERSED	Payment reversed/payment refunded
NONFINANCIAL	

Class methods:

isNotCanceled

Signature	Boolean isNotCanceled()
Input parameters	No
Return value	Payment reversed or refunded attribute
Description	Returns an attribute of payment reverse/refund.

Class TransactionItem.Card

TransactionItem nested class, **AbstractEntity** derived class. Contains payment card data.

Class properties:

Name	Description
Iin	Type of card or “cash”
Bin	Interior bank identifier
Exp	Card expiry date
PanMasked	First and last 4 numbers of a card number, separated by a “*” symbol
PanEnding	Last 4 numbers of a card

Class ScheduleItem

AbstractEntity derived class. Data object representation of a regular payment.

Class properties:

Name	Description
ID	Regular payment ID
Card	Payment card data in the view of **TransactionItem.Card**

Class APIResult

AbstractEntity derived class. Server response containing primitive entity.

Class methods:

getErrorCode

Signature	int getErrorCode()
Input parameters	No
Return value	Error code
Description	Returns an error code: 0 – if there is no error message -1 – if server response is not received or response format is incorrect

getErrorMessage

Signature	String getErrorMessage()
Input parameters	No
Return value	Error message
Description	Returns error message

isValid

Signature	boolean isValid()
Input parameters	No
Return value	Attribute of response that it has no error message and its format is correct
Description	Returns an attribute of response which has no error message and its format is correct

Class APIGetHistoryResult

APIResult derived class. Contains a set of transactions that received after history request.

Class methods:

getTransactions:

Signature	ArrayList<TransactionItem> getTransactions()
Input parameters	No
Return value	Transactions ArrayList
Description	Returns contained in response set of transactions

Package ibox.pro.sdk.external.ui

Class SignatureView

Is a **View**, that allows client to make a signature with a stylus or finger moving on the device screen.

Class properties:

Name	Description
color	Brush color

Class methods:

erase

Signature	erase()
Input parameters	No
Return value	No
Description	Erases signature field

getBitmap

Signature	Bitmap getBitmap()
Input parameters	No
Return value	Bitmap signature submission
Description	Returns bitmap signature submission

getBitmapByteArray

Signature	byte [] getBitmapByteArray()
Input parameters	No
Return value	byte [] signature submission
Description	Returns byte [] signature submission