

IboxPro API

Руководство по интеграции

V 1.5.3.8

История изменений

Версия	Дата	Описание
1.0.0	26.05.2015	Исходная версия
1.1.0	09.07.2015	Добавлен API истории платежей. Изменения в структуре библиотеки.
1.1.1	27.07.2015	Добавлен ReaderEvent. WAITING_FOR_CARD_CANCELED для PaymentController. Изменения в TransactionItem, TaxItem
1.2	11.08.2015	Добавлены: ReaderEvent. PAYMENT_CANCELED, ReaderEvent.EJECT_CARD. Удалены: ReaderEvent. WAITING_FOR_CARD_CANCELED, PaymentError.PAYMENT_IN_PROGRESS. Изменена сигнатура некоторых методов PaymentController. В PaymentControllerListener добавлен метод onScheduleCreationFailed. Правки внутренней логики
1.3	10.09.2015	Добавлена возможность отмены/возврата платежа, получение транзакции по ее ID, PaymentResultContext, ScheduleItem. Правки внутренней логики
1.3.1	11.09.2015	Прерывание транзакции. Правки внутренней логики
1.3.2	14.09.2015	Исправления в поведении PaymentController и считывателя карт при отмене/возврате платежа и отмене транзакции. Добавлено свойство PaymentContext.currencyName. Удалено свойство PaymentContext.Type
1.3.3	25.09.2015	Добавлены новые события BAD_SWIPE и LOW_BATTERY считывателя карт. Исправления внутренней логики
1.3.4	30.09.2015	Добавлен метод PaymentControllerListener.onTransactionStarted()
1.3.5	30.05.2016	Добавлен enum Currency, исправление ошибок округления

Версия	Дата	Описание
1.3.7	22.07.2016	Добавлены частичные отмены/возвраты, автоматическая конфигурация считывателей карт. Однофакторная авторизация. Исправления внутренней логики
1.3.8	28.07.2016	Добавлены признаки возможности частичной отмены/возврата транзакции
1.3.9	02.08.2016	Добавлены свойства класса TransactionItem
1.4.0	09.08.2016	Исправления внутренней логики
1.4.1	11.08.2016	Добавлены оплата наличными и отправка данных фискального регистратора
1.4.2	18.08.2016	Добавлен callback для получения уровня заряда считывателя карт в интерфейсе PaymentControllerListener
1.4.3	29.09.2016	Исправления внутренней логики
1.5.0	18.11.2016	Добавлена поддержка новых считывателей карт, добавлена поддержка NFC. Изменения в интерфейсе.
1.5.1	1.12.2016	Исправления логики регулярных платежей, исправление ошибок округления, улучшения работы ридеров QPOS. Добавлена отмена предоплат
1.5.2	11.01.2017	Ридер QPOS_MINI теперь поддерживает соединение по USB и оплату NFC. Улучшения внутренней логики.
1.5.3	16.01.2017	Добавлена поддержка ридера WISEPAD2_PLUS. Добавлен метод PaymentController.printText(), добавлен enum PaymentController.PrintResult.
1.5.3.1	23.01.2017	Добавлены поля CardholderName и TerminalName в TransactionItem
1.5.3.5	16.03.2017	Исправления внутренней логики

1.5.3.8	30.03.2017	Поддержка NFC для ридера QPOS Mini Улучшения внутренней логики
---------	------------	--

Содержание

Обзор	6
Разрешения Android	7
Пакет ibox.pro.sdk.external	8
Класс PaymentController	8
Интерфейс PaymentControllerListener	16
Класс PaymentContext	20
Класс RegularPaymentContext	21
Класс PaymentResultContext	22
Класс AbstractEntity	23
Пакет ibox.pro.sdk.external.entities	24
Класс TransactionItem	24
Класс TransactionItem.Card	26
Класс ScheduleItem	27
Класс APIResult	28
Класс APIGetHistoryResult	29
Пакет ibox.pro.sdk.external.ui	30
Класс SignatureView	30

Обзор

Разрешения Android

Перед началом работы с библиотекой в файл **AndroidManifest.xml** приложения необходимо добавить следующие строки:

```
<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" />
```

Пакет `ibox.pro.sdk.external`

Класс `PaymentController`

Данный класс является центральным в библиотеке. Содержит методы для создания транзакций и передачи в них дополнительных параметров а также инкапсулирует работу со считывателями карт. Класс содержит наборы различных параметров в виде **enum**, необходимых для проведения платежа.

Перед проведением транзакций необходимо установить пользовательские Email и пароль, необходимые для аутентификации с помощью метода **setCredentials** и установить тип считывателя карт с помощью метода **setReaderType**.

Перед началом работы со считывателем карт необходимо вызвать метод **enable**, а после завершения работы – метод **disable**. Вызов метода **disable** или отключение считывателя карт прервет обработку текущей транзакции

Для корректной работы также необходимо вызывать одноименные методы экземпляра класса при вызове в родительской **Activity** следующих методов: **onCreate**, **onDestroy**, **onSaveInstanceState**. Таймаут для извлечения карты при сбое транзакции – 5 секунд.

Для обработки событий считывателя карт и/или процесса выполнения транзакции экземпляру класса может быть передан **PaymentControllerListener**, с помощью метода **setPaymentControllerListener**.

При указании суммы платежа, разрядность десятичной части которой превышает разрядность десятичной части валюты, количество знаков после запятой такой суммы будет урезано **без округления**.

Наборы параметров:

`ReaderType`

Набор поддерживаемых типов считывателей карт

Тип	Описание
EMV_SWIPE	Считыватель карт «Chip&Sign», C15
EMV_SWIPE_BT	
WISEPAD	Считыватель карт «Chip&Pin», P15
WISEPAD2	
WISEPAD2_PLUS	
CHIPPER2X	
QPOS	
QPOS_MINI	P17

ReaderEvent

Набор возможных событий, которые могут быть переданы считывателем карт

Тип	Описание
CONNECTED	Считыватель карт был подключен
DISCONNECTED	Считыватель карт был отключен
START_INIT	Начало инициализации
INIT_SUCCESSFULLY	Инициализация завершена успешно
INIT_FAILED	Произошла ошибка инициализации
EJECT_CARD_TIMEOUT	Не используется
SWIPE_CARD	Обнаружено проведение магнитной полосой
EMV_TRANSACTION_STARTED	Начата чиповая транзакция
NFC_TRANSACTION_STARTED	Начата NFC транзакция
WAITING_FOR_CARD	Ожидание проведения магнитной полосой или вставки чиповой карты
PAYMENT_CANCELED	Платеж отменен пользователем
EJECT_CARD	Пользователь может извлечь карту (возникает при ошибке проведения транзакции)
BAD_SWIPE	Не удалось считать данные магнитной ленты
LOW_BATTERY	Уровень заряда батареи считывателя карт менее 10%

PaymentInputType

Набор возможных типов оплаты

Тип	Описание
SWIPE	Оплата с помощью проката карты магнитной лентой
CHIP	Оплата с помощью чипа на карте
NFC	Оплата NFC
CASH	Оплата наличными

PaymentError

Набор возможных ошибок, которые могут возникнуть в процессе выполнения платежа

Тип	Описание
CONNECTION_ERROR	Ошибка соединения с сервером
SERVER_ERROR	Ошибка выполнения транзакции
TRANSACTION_NULL_OR_EMPTY	Ошибка создания транзакции
NO_SUCH_TRANSACTION	Транзакция не была найдена, либо не уникальна
EMV_ERROR	Общая ошибка EMV
EMV_TERMINATED	Транзакция прервана
EMV_DECLINED	Транзакция отклонена
EMV_CANCEL	Транзакция отменена
EMV_CARD_ERROR	Ошибка карты
EMV_DEVICE_ERROR	Ошибка ридера
EMV_CARD_NOT_SUPPORTED	Карта не поддерживается

Тип	Описание
EMV_ZERO_TRAN_EMV	Попытка провести транзакцию на нулевую сумму
EMV_NOT_ALLOWED	Чиповая транзакция не разрешена
NFC_NOT_ALLOWED	NFC транзакция не разрешена

RegularRepeatType

Набор возможных типов регулярного платежа

Тип	Описание
Never	Платеж будет выполнен один раз
Weekly	Еженедельный платеж
Monthly	Ежемесячный платеж
Quarterly	Ежеквартальный платеж
Annual	Ежегодный платеж
ArbitraryDays	Платеж будет выполняться в заданные дни

RegularEndType

Набор возможных способов окончания выполнения регулярного платежа

Тип	Описание
BY_QUANTITY	Окончание по количеству повторов
BY_DAY	Окончание в заданный день

ReverseAction

Набор возможных способов отмены платежа

Тип	Описание
CANCEL	Отмена платежа
RETURN	Возврат платежа

Currency

Валюты, которыми можно провести оплату

Тип	Описание
RUB	Российский рубль
VND	Вьетнамский донг

PrintResult

Набор возможных результатов выполнения печати

Тип	Описание
SUCCESS	Печать успешно завершена
NO_PAPER	Нет бумаги
WRONG_CMD	Неправильная команда
OVERHEAT	Перегрев печатающей головки
TIMEOUT	Превышение таймаута ожидания ответа
PRINTER_ERROR	Ошибка принтера

Методы класса:

getInstance

Сигнатура	PaymentController getInstance()
Входные параметры	Нет
Возвращаемое значение	Экземпляр класса
Описание	Метод для получения экземпляра класса

onCreate

Сигнатура	void onCreate(Context context, Bundle savedInstanceState)
Входные параметры	context – контекст приложения savedInstanceState – передается из метода родительской Activity
Возвращаемое значение	Нет
Описание	Должен быть вызван при вызове одноименного метода родительской Activity

onDestroy

Сигнатура	void onDestroy()
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Должен быть вызван при вызове одноименного метода родительской Activity

onSaveInstanceState

Сигнатура	void onSaveInstanceState(Context context, Bundle savedInstanceState)
Входные параметры	savedInstanceState – передается из метода родительской Activity
Возвращаемое значение	Нет
Описание	Должен быть вызван при вызове одноименного метода родительской Activity

enable

Сигнатура	void enable()
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Начинает работу со считывателем карт

disable

Сигнатура	void disable()
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Завершает работу со считывателем карт

isConnected

Сигнатура	boolean isConnected()
Входные параметры	Нет
Возвращаемое значение	true , если считыватель карт подключен
Описание	Используется для проверки готовности считывателя карт

getBluetoothDevices

Сигнатура	ArrayList<BluetoothDevice> getBluetoothDevices(Context context)
Входные параметры	context – контекст приложения
Возвращаемое значение	ArrayList сопряженных устройств
Описание	Используется для получения набора доступных для соединения Bluetooth-устройств

setPaymentControllerListener

Сигнатура	void setPaymentControllerListener(PaymentControllerListener listener)
Входные параметры	listener – обработчик событий
Возвращаемое значение	Нет
Описание	Задаёт новый обработчик событий проведения платежа

setCredentials

Сигнатура	void setCredentials(String email, String password)
Входные параметры	email – email пользователя password – пароль пользователя
Возвращаемое значение	Нет
Описание	Задаёт данные пользователя, необходимые для проведения транзакций

setReaderType

Сигнатура	<code>void setReaderType(Context context, ReaderType readerType, String address, String config) throws IllegalStateException</code>
Входные параметры	<code>context</code> – контекст приложения <code>readerType</code> – тип считывателя карт <code>address</code> – MAC-адресс Bluetooth-считывателя карт. Для подключения USB передавать константу <code>PaymentController.USB_MODE_KEY</code> <code>config</code> – параметры конфигурации считывателя карт
Возвращаемое значение	Нет
Описание	Изменяет тип текущего считывателя карт. При попытке изменить тип считывателя во время проведения платежа будет сгенерировано исключение <code>IllegalStateException</code>

getReaderType

Сигнатура	<code>ReaderType getReaderType()</code>
Входные параметры	Нет
Возвращаемое значение	Текущий тип считывателя карт
Описание	Возвращает текущий тип считывателя карт

startPayment

Сигнатура	<code>void startPayment(Context context, PaymentContext paymentContext) throws PaymentException</code>
Входные параметры	<code>context</code> – контекст приложения <code>paymentContext</code> – данные платежа
Возвращаемое значение	Нет
Описание	Начинает выполнение платежа. При попытке начать новый платеж/отмену платежа до окончания будет сгенерировано исключение <code>PaymentException</code>

reversePayment

Сигнатура	<code>void reversePayment(Context context, String transactionID, ReverseAction action, Double amountToReverse, Currency currency) throws PaymentException</code>
Входные параметры	context – контекст приложения transactionID – ID транзакции отменяемого платежа action – Тип отмены amountToReverse – сумма, на которую будет выполнена отмена. Для полной отмены передавать null currency – валюта, используемая для отмены/возврата
Возвращаемое значение	Нет
Описание	Начинает выполнение отмены платежа. При попытке начать новый платеж/отмену платежа до окончания будет сгенерировано исключение <code>PaymentException</code>

adjust

Сигнатура	<code>APIResult adjust(Context context, String transactionID, String receiptPhone, String receiptEmail, byte [] signature)</code>
Входные параметры	context – контекст приложения transactionID – ID транзакции, для которой требуется отправить дополнительные данные receiptPhone – номер телефона для отправки чека receiptEmail – email для отправки чека signature – изображение с подписью плательщика
Возвращаемое значение	Результат отправки данных
Описание	Используется для отправки подписи и чека для транзакции единичного платежа

adjust

Сигнатура	<code>APIResult adjust(Context context, int regularID, byte [] signature)</code>
Входные параметры	context – контекст приложения regularID – ID транзакции, для которой требуется отправить дополнительные данные signature – изображение с подписью плательщика
Возвращаемое значение	Результат отправки данных
Описание	Используется для отправки подписи и чека для транзакции регулярного платежа

adjustReverse

Сигнатура	APIResult adjustReverse(Context String transactionID, String receiptPhone, String receiptEmail, byte [] signature)
Входные параметры	context – контекст приложения transactionID – ID транзакции, для которой требуется отправить дополнительные данные receiptPhone – номер телефона для отправки чека receiptEmail – email для отправки чека signature – изображение с подписью плательщика
Возвращаемое значение	Результат отправки данных
Описание	Используется для отправки подписи и чека для транзакции отмены платежа

isPaymentInProgress

Сигнатура	boolean isPaymentInProgress()
Входные параметры	Нет
Возвращаемое значение	true , если выполнение платежа не завершено
Описание	Используется для проверки состояния контроллера

getHistory

Сигнатура	APIGetHistoryResult getHistory(Context context, int page)
Входные параметры	context – контекст приложения page – номер страницы
Возвращаемое значение	Объект APIGetHistoryResult, содержащий набор транзакций
Описание	Позволяет получить историю транзакций в постраничном виде

getTransactionByID

Сигнатура	APIGetHistoryResult getTransactionByID(Context context, String transactionID)
Входные параметры	context – контекст приложения transactionID – ID запрашиваемой транзакции
Возвращаемое значение	Объект APIGetHistoryResult, содержащий запрашиваемую транзакцию
Описание	Позволяет получить данные транзакции по ее ID

startAutoConfig

Сигнатура	String startAutoConfig() throws IllegalStateException
Входные параметры	Нет
Возвращаемое значение	Строка конфигурации считывателя карт
Описание	Позволяет получить конфигурацию считывателя карт для дальнейшей работы с ним

setSingleStepEMV

Сигнатура	void setSingleStepEMV(boolean singleStepEMV)
Входные параметры	singleStepEMV – признак однофакторной авторизации
Возвращаемое значение	Нет
Описание	Позволяет проводить платежи с однофакторной авторизацией

getSingleStepEMV

Сигнатура	boolean isSingleStepEMV()
Входные параметры	Нет
Возвращаемое значение	Признак режима однофакторной авторизации
Описание	Возвращает признак режима однофакторной авторизации

submitFiscal

Сигнатура	APIResult submitFiscal(Context context, String transactionID, int printerID, int docID, int CVC, int shift)
Входные параметры	context – контекст приложения transactionID – ID транзакции, для которой требуется отправить фискальные данные printerID – ID фискального регистратора docID – сквозной номер документа CVC – КПК документа shift – номер операционной смены
Возвращаемое значение	Признак режима однофакторной авторизации
Описание	Возвращает признак режима однофакторной авторизации

printText

Сигнатура	PrintResult printText(String text, Layout.Alignment alignment) throws IllegalStateException
Входные параметры	text – Текст для печати alignment – Выравнивание текста
Возвращаемое значение	Результат печати
Описание	Команда работает только для ридера WISEPAD2_PLUS, иначе будет сгенерировано исключение IllegalStateException

Интерфейс PaymentControllerListener

Callback-интерфейс для класса **PaymentController**.

Методы интерфейса:

onTransactionStarted

Сигнатура	void onTransactionStarted(String transactionID)
Входные параметры	transactionID – ID выполняемой транзакции
Возвращаемое значение	Нет
Описание	Метод будет вызван перед выполнением платежа. Вызывается только для простого(единичного) платежа

onFinished

Сигнатура	void onFinished(PaymentResultContext result)
Входные параметры	result – данные о проведенной транзакции
Возвращаемое значение	Нет
Описание	Метод будет вызван при успешном проведении платежа или успешной отмене платежа

onError

Сигнатура	void onError(PaymentError error, String errorMessage)
Входные параметры	error – тип ошибки errorMessage – сообщение об ошибке. Используется только когда error == SERVER_ERROR
Возвращаемое значение	Нет
Описание	Метод будет вызван при возникновении ошибки во время попытки проведения транзакции

onSelectApplication

Сигнатура	int onSelectApplication(List<String> apps)
Входные параметры	apps – список названий приложений
Возвращаемое значение	Порядковый номер выбранного приложения (начиная с 0)
Описание	Метод будет вызван при выполнении чиповой транзакции, если чиповая карта содержит более 1 приложения. Вызов метода происходит не в родительском потоке.

onConfirmSchedule

Сигнатура	boolean onConfirmSchedule(List<Map.Entry<Date, Double>> steps, double totalAmount)
Входные параметры	steps – список шагов выполнения расписания, состоящий из пар типа <Дата списания, Сумма к списанию> totalAmount – итоговая сумма по всем дням
Возвращаемое значение	Признак того, что плательщик подтверждает правильность расписания
Описание	Метод будет вызван при создании регулярного платежа. Вызов метода происходит не в родительском потоке.

onScheduleCreationFailed

Сигнатура	boolean onScheduleCreationFailed(PaymentError error, String errorMessage)
Входные параметры	error – тип ошибки errorMessage – сообщение об ошибке. Используется только когда error == SERVER_ERROR
Возвращаемое значение	true , если необходимо повторить попытку создания расписания
Описание	Метод будет вызван в случае возникновения ошибки при создании расписания регулярного платежа

onCancellationTimeout

Сигнатура	boolean onCancellationTimeout()
Входные параметры	Нет
Возвращаемое значение	true , для выполнения возврата платежа
Описание	Метод будет вызван в случае попытки выполнения отмены платежа по истечению доступного для отмены таймаута

onPinRequest

Сигнатура	void onPinRequest()
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Метод будет вызван при запросе PIN-кода карты считывателем карт

onPinEntered

Сигнатура	void onPinEntered()
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Метод будет вызван после ввода PIN-кода карты

onAutoconfigUpdate

Сигнатура	void onAutoConfigUpdate(double percent)
Входные параметры	percent – индикатор прогресса, в процентах
Возвращаемое значение	Нет
Описание	Метод будет вызван при обновлении прогресса во время выполнения автоконфигурации считывателя карт

onAutoconfigFinished

Сигнатура	void onAutoConfigFinished(boolean success, String config, boolean isDefault)
Входные параметры	success – true , если автоконфигурация выполнена успешно config – строка конфигурации isDefault – true , если были использованы настройки по умолчанию
Возвращаемое значение	Нет
Описание	Метод будет вызван при завершении автоконфигурации считывателя карт

onBatteryState

Сигнатура	void onBatteryState(double percent)
Входные параметры	percent – уровень заряда считывателя карт, в процентах
Возвращаемое значение	Нет
Описание	Метод будет вызван после инициализации ридера

Класс PaymentContext

JavaBean контейнер данных, необходимых для выполнения разового платежа.

Свойства класса:

Название	Описание
amount	Сумма платежа
currency	Валюта платежа
description	Описание платежа
transactionID	Не используется
image	Изображение, прикрепленное к платежу
currency	Валюта платежа
cash	Признак оплаты наличными

Методы класса:

reset

Сигнатура	reset()
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Выполняет очистку полей объекта

Класс RegularPaymentContext

Расширение класса **PaymentContext**, дополнительно содержащее свойства, необходимые для создания регулярного платежа. Для выполнения платежа в последний день месяца свойство **dayOfWeek** должно иметь значение, равное константе **LAST_DAY_OF_MONTH**.

Свойства класса:

Название	Описание
repeatType	Тип регулярного платежа
endType	Способ завершения выполнения регулярного платежа
startDate	Дата начала выполнения регулярного платежа
endDate	Дата окончания выполнения регулярного платежа (если окончание по дате)
repeatCount	Количество выполнений регулярного платежа (если окончание по количеству повторов)
arbitraryDays	Дни, заданные для выполнения платежа (если тип платежа – по заданным датам)
month	Месяц для выполнения платежа ([1,12] и [1,4] при repeatType == Quarterly)
day	День для выполнения платежа ([1,31])
dayOfWeek	День недели для выполнения платежа ([0,7], 0 – Воскресенье)
hour	Час выполнения платежа
minute	Минута выполнения платежа
receiptEmail	Email для отправки чека
receiptPhone	Телефон для отправки чека

Набор необходимых заполненных свойств зависит от типа платежа:

Тип платежа	Набор свойств
Never	startDate
Weekly	startDate, (endDate или repeatCount)
Monthly	startDate, (endDate или repeatCount), day
Quarterly	startDate, (endDate или repeatCount), month, day
Annual	startDate, (endDate или repeatCount), month, day
ArbitraryDays	arbitraryDays

Параметры repeatType, endType, receiptEmail, receiptPhone являются обязательными для всех типов регулярных платежей.

Параметры hour, minute являются необязательными для всех типов регулярных платежей.

Класс PaymentResultContext

JavaBean контейнер данных, полученных при успешном проведении платежа или отмене платежа.

Свойства класса:

Название	Описание
transactionItem	Данные о транзакции платежа/отмены платежа в представлении TransactionItem
scheduleItem	Данные о транзакции регулярного платежа в представлении ScheduleItem
requiresSignature	признак необходимости отправки подписи плательщика после оплаты
terminalName	Терминал
emvData	Набор данных EMV(чиповой) транзакции в представлении HashMap<String, String>

Класс AbstractEntity

Абстрактный класс-обертка для массива данных в представлении JSON. Реализует интерфейс **Serializable**.

Методы класса:

getJSON

Сигнатура	JSONObject getJSON()
Входные параметры	Нет
Возвращаемое значение	JSON представление набора данных
Описание	Возвращает JSON представление набора данных

Пакет `ibox.pro.sdk.external.entities`

Класс `TransactionItem`

Дочерний класс **AbstractEntity**. Является объектным представлением транзакции. Содержит набор свойств, определяющих ее. Имеет вложенные классы **Card** и **Format**.

Свойства класса:

Название	Описание
ID	ID транзакции
Date	Время и дата выполнения транзакции, согласно GMT устройства
Description	Описание транзакции
Invoice	Номер чека
ApprovalCode	Код подтверждения
ScheduleID	ID регулярного платежа
ScheduleStepID	ID списания для рекуррентного платежа
Amount	Сумма транзакции
AmountEff	Баланс транзакции
InputType	Способ оплаты в представлении InputType
Operation	Название операции
Latitude	Географическая широта места выполнения транзакции
Longitude	Географическая долгота места выполнения транзакции
HasPhoto	Признак наличия приложенного изображения
PhotoUrl	URL приложенного изображения
HasSignature	Признак наличия приложенной подписи
SignatureUrl	URL приложенной подписи
StateDisplay	Описание состояния транзакции
Card	Данные карты, которая была использована для оплаты, в представлении TransactionItem.Card
CanCancel	Признак возможности проведения отмены платежа
CanReturn	Признак возможности проведения возврата платежа
CanCancelPartial	Признак возможности проведения частичной отмены платежа
CanReturnPartial	Признак возможности проведения частичного возврата платежа
DisplayMode	Тип отображения транзакции в представлении DisplayMode
SubstateDisplay	Описание подсостояния транзакции
CardholderName	Владелец платежной карты
TerminalName	Терминал

Наборы параметров:

InputType

Набор возможных способов оплаты

Тип	Описание
SWIPE	Оплата с помощью проката карты магнитной лентой
CHIP	Оплата с помощью чипа на карте
NFC	Оплата NFC
CASH	Оплата наличными

DisplayMode

Тип отображения транзакции

Тип	Описание
DECLINED	Отклоненная транзакция
SUCCESS	Успешная транзакция
REVERSE	Транзакция отмены/возврата
REVERSED	Платеж отменен/выполнен возврат
NONFINANCIAL	

Методы класса:

isNotCanceled

Сигнатура	Boolean isNotCanceled()
Входные параметры	Нет
Возвращаемое значение	Признак того, что для платежа выполнена отмена или возврат
Описание	Возвращает признак отмены/возврата транзакции

Класс TransactionItem.Card

Вложенный класс **TransactionItem**, дочерний класс **AbstractEntity**. Содержит данные о платежной карте.

Свойства класса:

Название	Описание
Iin	Тип карты или "cash"(в случае оплаты наличными)
Bin	Внутренний идентификатор банка
Exp	Срок действия карты
PanMasked	Первые и последние 4 цифры номера карты, разделенные символом "*"
PanEnding	Последние 4 цифры номера карты

Класс ScheduleItem

Дочерний класс **AbstractEntity**. Является объектным представлением данных о регулярном платеже

Свойства класса:

Название	Описание
ID	ID регулярного платежа
Card	Данные карты, которая была использована для оплаты, в представлении TransactionItem.Card

Класс APIResult

Дочерний класс **AbstractEntity**. Является примитивной сущностью, содержащую ответ от сервера

Методы класса:

getErrorCode

Сигнатура	Int getErrorCode()
Входные параметры	Нет
Возвращаемое значение	Код ошибки
Описание	Возвращает код ошибки. 0 – если ответ не содержит сообщений об ошибке, -1 – если ответ от сервера не получен, или формат ответа неправильный

getErrorMessage

Сигнатура	String getErrorMessage()
Входные параметры	Нет
Возвращаемое значение	Сообщение об ошибке
Описание	Возвращает сообщение об ошибке

isValid

Сигнатура	boolean isValid()
Входные параметры	Нет
Возвращаемое значение	Признак того, что ответ не содержит сообщений об ошибке и его формат правильный
Описание	Возвращает признак того, что ответ не содержит сообщений об ошибке и его формат правильный

Класс **APIGetHistoryResult**

Дочерний класс **APIResult**. Содержит набор транзакций, полученных в ответ на запрос истории.

Методы класса:

`getTransactions`

Сигнатура	<code>ArrayList<TransactionItem> getTransactions()</code>
Входные параметры	Нет
Возвращаемое значение	ArrayList транзакций
Описание	Возвращает набор транзакций, содержащихся в ответе.

Пакет `ibox.pro.sdk.external.ui`

Класс `SignatureView`

Является View, предоставляющим возможность выполнения подписи клиента с помощью передвижений пальца или стилуса по экрану.

Свойства класса:

Название	Описание
<code>color</code>	Цвет кисти

Методы класса:

`erase`

Сигнатура	<code>erase()</code>
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Очищает поле для подписи

`getBitmap`

Сигнатура	<code>Bitmap getBitmap()</code>
Входные параметры	Нет
Возвращаемое значение	Bitmap представление подписи
Описание	Возвращает Bitmap представление подписи

`getBitmapByteArray`

Сигнатура	<code>byte [] getBitampByteArray()</code>
Входные параметры	Нет
Возвращаемое значение	<code>byte []</code> представление подписи
Описание	Возвращает <code>byte []</code> представление подписи