

2can API

Руководство по интеграции

V 1.5.9.8

---

## История изменений

| Версия | Дата       | Описание   |
|--------|------------|--|
| 1.0.0  | 26.05.2015 | Исходная версия  |
| 1.1.0  | 09.07.2015 | Добавлен API истории платежей.<br>Изменения в структуре библиотеки.  |
| 1.1.1  | 27.07.2015 | Добавлен ReaderEvent.<br>WAITING_FOR_CARD_CANCELED для PaymentController. Изменения в TransactionItem, TaxItem   |
| 1.2    | 11.08.2015 | Добавлены: ReaderEvent.<br>PAYMENT_CANCELED,<br>ReaderEvent.EJECT_CARD.<br>Удалены: ReaderEvent.<br>WAITING_FOR_CARD_CANCELED,<br>PaymentError.PAYMENT_IN_PROGRESS.<br>Изменена сигнатура некоторых методов PaymentController.<br>В PaymentControllerListener добавлен метод onScheduleCreationFailed.<br>Правки внутренней логики |
| 1.3    | 10.09.2015 | Добавлена возможность отмены/возврата платежа, получение транзакции по ее ID,<br>PaymentResultContext, ScheduleItem.<br>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.<br>Добавлено свойство PaymentContext.currencyName.<br>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,<br>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  |

| Версия  | Дата       | Описание   |
|---------|------------|--|
| 1.5.3.9 | 05.04.2017 | Исправлена работа вызова PaymentControllerListener.onSelectApplication().  |
| 1.5.4.0 | 12.04.2017 | Добавлена поддержка ридеров M17<br>Изменены названия ридеров<br>Убрана поддержка неиспользуемых ридеров<br>Убрана поддержка автоконфигурации<br>Общие улучшения  |
| 1.5.4.1 | 18.04.2017 | Исправление падения для устройств, не оснащенных Bluetooth   |
| 1.5.4.3 | 30.06.2017 | Добавлена перегрузка PaymentController.submitFiscal() для отправки фискальных данных по стандарту Ф3-54  |
| 1.5.5.3 | 25.07.2017 | Добавлена возможность проведения отмены через EMV/NFC.<br>Добавлен метод auth() для проверки учетных данных и получения информации об учетной записи.<br>Общие улучшения.  |
| 1.5.6.1 | 22.09.2017 | Добавлена оплата по ссылке, обновлена процедура платежа, добавлен enum PaymentMethod, удален enum TransactionItem.InputType, добавлена установка банка(acquirer) для платежа, общие улучшенияsss   |
| 1.5.6.5 | 13.10.2017 | Исправлено проведение EMV/NFC транзакций для ридера QPOS.<br>Добавлено свойство PaymentController.ClientProductCode.<br>Обновлена сигнатура методов PaymentController.reversePayment().<br>Добавлены новые свойства для PaymentContext, TransactionItem и TransactionItem.Card |
| 1.5.6.8 | 31.10.2017 | PaymentControllerListener.OnTransactionStarted() теперь вызывается и для отмены/возврата платежа. Общие улучшения  |
| 1.5.6.9 | 01.11.2017 | Данные для отображения QR в ExternalPayment теперь могут приходить в количестве более одного кода  |

| Версия  | Дата       | Описание   |
|---------|------------|--|
| 1.5.7.0 | 02.11.2017 | Сумма транзакций по ридеру ограничена 1 000 000. Сумма для возврата/отмены не может превышать остаток транзакции. Добавлена ошибка INVALID_AMOUNT  |
| 1.5.7.1 | 06.11.2017 | Добавлены транзакции «В обработке» для APIGetHistoryResult   |
| 1.5.7.3 | 09.11.2017 | Добавлена поддержка TLSv1.2  |
| 1.5.7.6 | 22.11.2017 | Добавлен ИНН пользователя в APIAuthResult  |
| 1.5.7.6 | 10.01.2018 | Исправление ошибок   |
| 1.5.8.1 | 15.01.2018 | Добавлена поддержка ридера C15   |
| 1.5.8.3 | 02.02.2018 | Добавлена поддержка привязанных карт   |
| 1.5.8.5 | 08.02.2018 | Добавлена поддержка автоконфигурации   |
| 1.5.8.6 | 13.02.2018 | Добавлены поля:<br>PaymentResultContext.AttachedCard<br>LinkedCard.PanMasked   |
| 1.5.8.7 | 16.02.2018 | Добавлен метод<br>PaymentController.balanceInquiry();<br>Добавлено поле<br>TransactionItem.Balance;<br>Исправление ошибок  |
| 1.5.8.8 | 23.02.2018 | Добавлена поддержка установки банка для методов<br>PaymentController.addLinkedCard() и<br>PaymentController.balanceInquiry().<br>Добавлено поле<br>Account.PaymentOptions.<br>Исправлен ошибочный запрос карты ридером при попытке возврата наличных, если ридер еще не был подключен ранее. |
| 1.5.8.9 | 27.02.2018 | Добавлено поле LinkedCard.Balance  |
| 1.5.9.0 | 06.03.2018 | Добавлен тип оплаты «Внешний POS-терминал», добавлена поддержка чиповых и NFC регулярных платежей. Исправление ошибок.   |

| Версия  | Дата       | Описание  |
|---------|------------|---|
| 1.5.9.1 | 07.03.2018 | Транзакции теперь могут быть отменены в режиме CNP, добавлены поля TransactionItem.CanCancelCNP, TransactionItem.CanCancelCNPpartial. Добавлен callback PaymentControllerListener.onSwitchedToCNP(). Исправлена ошибка отмена/возвраты транзакций, оплаченных способом «Внешний POS-терминал» |
| 1.5.9.2 | 13.03.2018 | Добавлен класс ReversePaymentContext, добавлено поле PaymentContext.ExtID, исправлена ошибка, когда APIResult.getErrorMessage() возвращал строку "null" вместо пустой строки  |
| 1.5.9.3 | 15.03.2018 | Добавлено свойство PaymentController.RepeatOnError, исправление ошибок  |
| 1.5.9.5 | 27.03.2018 | Добавлен поиск транзакций по RRN, добавлен метод PaymentController.submitEmailNotification(). Добавлены новые поля в Account, TransactionItem и LinkedCard. Исправлена ошибка при частичном возврате для уже частично отмененного платежа.  |
| 1.5.9.7 | 10.04.2018 | Добавлен класс FiscalInfo. Улучшена работа ридеров QPOS. Исправление ошибок.  |
| 1.5.9.8 | 12.04.2018 | Добавлено очищение контекста операции после успешного выполнения и при вызове метода PaymentController.disable()  |
| 1.5.9.9 | 17.04.2018 | Исправлена ошибка отсутствия запроса карты при возврате true из PaymentControllerListener.onCancellationTimeout() при PaymentController.RepeatOnError=true  |

## Содержание

|  |    |
|--|----|
| История изменений .....                                  | 2  |
| Обзор .....  | 8  |
| Разрешения Android .....                                 | 9  |
| Пакет <code>ibox.pro.sdk.external</code> .....           | 10 |
| Класс <code>PaymentController</code> .....               | 10 |
| Интерфейс <code>PaymentControllerListener</code> .....   | 24 |
| Класс <code>PaymentContext</code> .....                  | 27 |
| Класс <code>RegularPaymentContext</code> .....           | 28 |
| Класс <code>ReversePaymentContext</code> .....           | 29 |
| Класс <code>PaymentResultContext</code> .....            | 30 |
| Класс <code>AbstractEntity</code> .....                  | 31 |
| Пакет <code>ibox.pro.sdk.external.entities</code> .....  | 32 |
| Класс <code>TransactionItem</code> .....                 | 32 |
| Класс <code>TransactionItem.Card</code> .....            | 34 |
| Класс <code>TransactionItem.ExternalPayment</code> ..... | 35 |
| Класс <code>TransactionItem.FiscalInfo</code> .....      | 36 |
| Класс <code>ScheduleItem</code> .....                    | 37 |
| Класс <code>Account</code> .....                         | 38 |
| Класс <code>Account.PaymentOption</code> .....           | 39 |
| Класс <code>LinkedCard</code> .....                      | 40 |
| Класс <code>APIResult</code> .....                       | 41 |
| Класс <code>APIGetHistoryResult</code> .....             | 42 |
| Класс <code>APIAuthResult</code> .....                   | 43 |
| Класс <code>APIReadLinkedCardsResult</code> .....        | 44 |
| Пакет <code>ibox.pro.sdk.external.ui</code> .....        | 45 |
| Класс <code>SignatureView</code> .....                   | 45 |
| Приложение 1: Печать слипа .....                         | 46 |

## 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                              |
|---------|---------------------------------------|
| P15     | Считыватель карт «Chip&Pin», P15      |
| P16PLUS | Считыватель карт «Chip&Pin», P16 Plus |
| M17     |                                       |
| P17     | Считыватель карт «Chip&Pin NFC», 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 |

| Тип                     | Описание  |
|-------------------------|---|
| 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                                      |
| PREPAID    | Предоплата                                      |
| CREDIT     | Оплата кредитом                                 |
| CASH       | Оплата наличными                                |
| OTHER      | Оплата по ссылке                                |
| OUTER_CARD | Оплата внешним POS-терминалом                   |

### 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                       |
| EMV_ZERO_TRAN_EMV         | Попытка провести транзакцию на нулевую сумму  |
| EMV_NOT_ALLOWED           | Чиповая транзакция не разрешена               |

| Тип             | Описание   |
|-----------------|--|
| NFC_NOT_ALLOWED | NFC транзакция не разрешена                        |
| INVALID_AMOUNT  | Сумма возврата/отмены превышает остаток транзакции |

### 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 таймута ожидания ответа |
| PRINTER_ERROR | Ошибка принтера                    |

## PaymentMethod

Набор возможных способов выполнения платежа

| Тип         | Описание               |
|-------------|------------------------|
| CARD        | Платежной картой       |
| CREDIT      | Кредитом               |
| CASH        | Наличные               |
| OTHER       | По ссылке              |
| LINKED_CARD | Привязанной картой     |
| OUTER_CARD  | Внешним POS-терминалом |

## 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<br>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 | <b>true</b> , 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<br>password – пароль пользователя      |
| Возвращаемое значение | Нет   |
| Описание              | Задаёт данные пользователя, необходимые для проведения транзакций |

## auth

|                       |   |
|-----------------------|---|
| Сигнатура             | APIAuthResult auth(Context context)   |
| Входные параметры     | context – контекст приложения   |
| Возвращаемое значение | APIAuthResult   |
| Описание              | Проверяет правильность введенных учетных данных и возвращает информацию об учетной записи |

### setReaderType

|                       |   |
|-----------------------|---|
| Сигнатура             | <code>void setReaderType(Context context, ReaderType readerType, String address, String config) throws IllegalStateException</code>   |
| Входные параметры     | context – контекст приложения<br>readerType – тип считывателя карт<br>address – MAC-адресс Bluetooth-считывателя карт. Для подключения USB передавать константу <code>PaymentController.USB_MODE_KEY</code><br>config – 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     | context – контекст приложения<br>paymentContext – данные платежа  |
| Возвращаемое значение | Нет   |
| Описание              | Начинает выполнение платежа. При неверных параметрах платежа или 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, String receiptPhone, String receiptEmail) throws PaymentException</code>   |
| Входные параметры | context – контекст приложения<br>transactionID – ID транзакции отменяемого платежа<br>action – Тип отмены<br>amountToReverse – сумма, на которую будет выполнена отмена. Для полной отмены передавать <b>null</b><br>currency – валюта, используемая для отмены/возврата |



|                       |  |
|-----------------------|--|
|                       | receiptPhone – номер телефона для отправки чека<br>receiptEmail – email для отправки чека    |
| Возвращаемое значение | Нет  |
| Описание              | Устаревший. См. reversePayment(Context context, ReversePaymentContext reversePaymentContext) |

#### reversePayment

|                       |   |
|-----------------------|---|
| Сигнатура             | void reversePayment(Context context, ReversePaymentContext reversePaymentContext) throws PaymentException                                       |
| Входные параметры     | context – контекст приложения<br>reversePaymentContext – данные для проведения отмены/возврата  |
| Возвращаемое значение | Нет   |
| Описание              | Начинает выполнение отмены платежа. При попытке начать новый платеж/отмену платежа до окончания будет сгенерировано исключение PaymentException |

#### adjust

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

#### adjust

|                       |   |
|-----------------------|---|
| Сигнатура             | APIResult adjust(Context context, int regularID, byte [] signature)   |
| Входные параметры     | context – контекст приложения<br>regularID – ID транзакции, для которой требуется отправить дополнительные данные<br>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<br>transactionID – ID транзакции, для которой требуется отправить дополнительные данные<br>receiptPhone – номер телефона для отправки чека<br>receiptEmail – email для отправки чека<br>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 | <b>true</b> , если выполнение платежа не завершено |
| Описание              | Используется для проверки состояния контроллера    |

### getHistory

|                       |   |
|-----------------------|---|
| Сигнатура             | APIGetHistoryResult getHistory(Context context, int page) |
| Входные параметры     | context – контекст приложения<br>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<br>transactionID – ID запрашиваемой транзакции  |
| Возвращаемое значение | Объект APIGetHistoryResult, содержащий запрашиваемую транзакцию               |
| Описание              | Позволяет получить данные транзакции по ее ID                                 |

### getTransactionByRRN

|                   |  |
|-------------------|--|
| Сигнатура         | APIGetHistoryResult getTransactionByRRN(Context context, String rrn) |
| Входные параметры | context – контекст приложения<br>rrn – Retrieval Reference Number    |

|                       |   |
|-----------------------|---|
| Возвращаемое значение | Объект APIGetHistoryResult, содержащий запрашиваемую транзакцию |
| Описание              | Позволяет получить данные транзакции по ее RRN                  |

#### 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<br>transactionID – ID транзакции, для которой требуется отправить фискальные данные<br>printerID – ID фискального регистратора<br>docID – сквозной номер документа<br>CVC – КПК документа<br>shift – номер операционной смены |
| Возвращаемое значение | Результат отправки данных   |
| Описание              | Выполняет отправку данных фискализации  |

#### submitFiscal

|                   |   |
|-------------------|---|
| Сигнатура         | APIResult submitFiscal(Context context, String transactionID, String printerID, int shiftSN, int docSN, String fdn, String fdm, String fs, Date fdt)  |
| Входные параметры | context – контекст приложения<br>transactionID – ID транзакции, для которой требуется отправить фискальные данные<br>printerID – ID фискального регистратора<br>shift – номер операционной смены<br>docID – сквозной номер документа<br>fdn – фискальный номер документа<br>fdm – фискальный признак документа<br>fs – фискальное хранилище |

|                       |   |
|-----------------------|---|
|                       | fdt – дата и время проведения фискальной операции               |
| Возвращаемое значение | Результат отправки данных                                       |
| Описание              | Выполняет отставку данных фискализации согласно стандарту Ф3-54 |

#### printText

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

#### setClientProductCode

|                       |   |
|-----------------------|---|
| Сигнатура             | void setClientProductCode(String clientProductCode) |
| Входные параметры     | Код клиентского приложения                          |
| Возвращаемое значение | Нет   |
| Описание              | Устанавливает код клиентского приложения            |

#### getClientProductCode

|                       |   |
|-----------------------|---|
| Сигнатура             | String getClientProductCode()                       |
| Входные параметры     | Нет   |
| Возвращаемое значение | Установленный код клиентского приложения            |
| Описание              | Возвращает устанавливает код клиентского приложения |

#### addLinkedCard

|                       |   |
|-----------------------|---|
| Сигнатура             | void addLinkedCard(Context context, PaymentController.Currency currency, String acquirerCode)<br>throws PaymentException  |
| Входные параметры     | context – контекст приложения<br>currency – валюта карты<br>acquirerCode – код банка  |
| Возвращаемое значение | нет   |
| Описание              | Начинает выполнение процедуры привязки карты. При попытке привязать карту до окончания выполняющегося платежа(отмены) будет сгенерировано исключение PaymentException. Асинхронный, см.<br><b>PaymentControllerListener</b> |

### addLinkedCard

|                       |  |
|-----------------------|--|
| Сигнатура             | void addLinkedCard(Context context, PaymentController.Currency currency) throws PaymentException |
| Входные параметры     |  |
| Возвращаемое значение |  |
| Описание              | Устаревший.<br>См. addLinkedCard (Context context, PaymentController.Currency currency, null)    |

### removeLinkedCard

|                       |  |
|-----------------------|--|
| Сигнатура             | APIResult removeLinkedCard(Context context, int cardID)        |
| Входные параметры     | context – контекст приложения<br>cardID – ID удаляемой карты   |
| Возвращаемое значение | Результат операции   |
| Описание              | Удаляет привязанную карту для текущего аккаунта.<br>Синхронный |

### getLinkedCards

|                       |  |
|-----------------------|--|
| Сигнатура             | APIReadLinkedCardsResult getLinkedCards(Context context)             |
| Входные параметры     | context – контекст приложения  |
| Возвращаемое значение | Объект APIReadLinkedCardsResult, содержащий набор привязанных карт   |
| Описание              | Запрашивает набор привязанных карт для текущего аккаунта. Синхронный |

### 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 |

### balanceInquiry

|                       |  |
|-----------------------|--|
| Сигнатура             | void balanceInquiry(Context context, PaymentController.Currency currency, String acquirerCode) throws PaymentException |
| Входные параметры     | context – контекст приложения<br>currency – валюта карты<br>acquirerCode – код банка                                   |
| Возвращаемое значение | нет  |

|          |   |
|----------|---|
| Описание | Начинает выполнение процедуры запроса баланса карты. При попытке проверить баланс до окончания выполняющегося платежа(отмены) будет сгенерировано исключение <code>PaymentException</code> . Асинхронный, см. <b><code>PaymentControllerListener</code></b> |
|----------|---|

#### `balanceInquiry`

|                       |  |
|-----------------------|--|
| Сигнатура             | <code>void balanceInquiry(Context context, PaymentController.Currency currency) throws PaymentException</code> |
| Входные параметры     |  |
| Возвращаемое значение |  |
| Описание              | Устаревший.<br>См. <code>balanceInquiry(Context context, PaymentController.Currency currency, null)</code>     |

#### `setRepeatOnError`

|                       |  |
|-----------------------|--|
| Сигнатура             | <code>void setRepeatOnError(boolean repeatOnError)</code>  |
| Входные параметры     | <code>repeatOnError</code> – признак необходимости повторного запроса карты при ошибке проведения транзакции   |
| Возвращаемое значение | Нет  |
| Описание              | Позволяет включить/выключить режим повторного запроса карты считывателем карт при ошибке проведения транзакции |

#### `getRepeatOnError`

|                       |  |
|-----------------------|--|
| Сигнатура             | <code>boolean getRepeatOnError ()</code>   |
| Входные параметры     | Нет  |
| Возвращаемое значение | Признак необходимости повторного запроса карты при ошибке проведения транзакции            |
| Описание              | Возвращает признак необходимости повторного запроса карты при ошибке проведения транзакции |

#### `submitEmailNotification`

|                       |  |
|-----------------------|--|
| Сигнатура             | <code>APIResult submitEmailNotification(Context context, String email, String subj, String body, Map&lt;String, byte[]&gt; images)</code>  |
| Входные параметры     | <code>context</code> – контекст приложения<br><code>email</code> – адрес для отправки письма<br><code>subj</code> – тема письма<br><code>body</code> – html содержимое письма<br><code>images</code> – набор изображений |
| Возвращаемое значение | Результат отправки запроса   |

|          |   |
|----------|---|
| Описание | <p>Позволяет отправить уведомление на email пользователя.</p> <p>Изображения в письме должны размещаться по своему коду указанном в атрибуте src как параметр cid:код_изображения (пример &lt;img src="cid:test_image"&gt;).</p> <p>Пары код – изображения указываются в images</p> |
|----------|---|

## Интерфейс PaymentControllerListener

Callback-интерфейс для класса **PaymentController**.

### Методы интерфейса:

#### onTransactionStarted

|                       |  |
|-----------------------|--|
| Сигнатура             | void onTransactionStarted(String transactionID)  |
| Входные параметры     | transactionID – ID выполняемой транзакции для простого платежа. ID отменяемой транзакции для возврата/отмены платежа |
| Возвращаемое значение | Нет  |
| Описание              | Метод будет вызван перед выполнением транзакции  |

#### onFinished

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

#### onError

|                       |   |
|-----------------------|---|
| Сигнатура             | void onError(PaymentError error, String errorMessage)   |
| Входные параметры     | error – тип ошибки<br>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             | 250olean 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><br>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             | 250olean onScheduleCreationFailed(PaymentError error, String errorMessage)                                |
| Входные параметры     | error – тип ошибки<br>errorMessage – сообщение об ошибке. Используется только когда error == SERVER_ERROR |
| Возвращаемое значение | <b>true</b> , если необходимо повторить попытку создания расписания                                       |
| Описание              | Метод будет вызван в случае возникновения ошибки при создании расписания регулярного платежа              |

### onCancellationTimeout

|                       |  |
|-----------------------|--|
| Сигнатура             | 250olean onCancellationTimeout()   |
| Входные параметры     | Нет  |
| Возвращаемое значение | <b>true</b> , для выполнения возврата платежа  |
| Описание              | Метод будет вызван в случае попытки выполнения отмены платежа по истечению доступного для отмены таймута |

### 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 |

### 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          |

### onSelectInputType

|                       |  |
|-----------------------|--|
| Сигнатура             | PaymentController.PaymentInputType<br>onSelectInputType(List<PaymentController.PaymentInputType><br>allowedInputTypes) |
| Входные параметры     | allowedInputTypes – перечень доступных типов ввода для проведения отмены/возврата                                      |
| Возвращаемое значение | Выбранный тип ввода  |
| Описание              | Метод будет вызван при проведении отмены/возврата, если необходимо выбрать способ проведения транзакции                |

### onAutoconfigUpdate

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

### onAutoconfigFinished

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

### onSwitchedToCNP

|                       |  |
|-----------------------|--|
| Сигнатура             | void onSwitchedToCNP()   |
| Входные параметры     | Нет  |
| Возвращаемое значение | Нет  |
| Описание              | Метод будет вызываться при выполнении отмены транзакции в режиме CNP |

## 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(устаревший) |
| credit        | Признак оплаты кредитом(устаревший)  |
| nfc           | Признак проведения транзакции NFC    |
| method        | Способ оплаты                        |
| acquirerCode  | Код банка                            |
| receiptEmail  | Email для отправки чека              |
| receiptPhone  | Телефон для отправки чека            |
| linkedCardID  | ID привязанной карты                 |
| extID         | ID клиентского приложения            |

### 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   |

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 ReversePaymentContext

JavaBean контейнер данных, необходимых для выполнения отмены/возврата платежа.

### Свойства класса:

| Название      | Описание   |
|---------------|--|
| transactionID | ID транзакции отменяемого платежа  |
| action        | Тип отмены   |
| currency      | валюта, используемая для отмены/возврата   |
| returnAmount  | сумма, на которую будет выполнена отмена. Для полной отмены установить <b>null</b> |
| auxData       | Перечень товаров в установленном формате   |
| extID         | ID клиентского приложения  |
| receiptEmail  | Email для отправки чека  |
| receiptPhone  | Телефон для отправки чека  |

### Методы класса:

#### reset

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

## 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 <b>TransactionItem</b>         |
| scheduleItem      | Данные о транзакции регулярного платежа в представлении <b>ScheduleItem</b>               |
| requiresSignature | признак необходимости отправки подписи плательщика после оплаты                           |
| terminalName      | Терминал  |
| emvData           | Набор данных EMV(чиповой) транзакции в представлении <b>HashMap&lt;String, String&gt;</b> |
| attachedCard      | Данные о карте, привязанной при вызове <code>PaymentController.addLinkedCard()</code>     |

## 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   |
| 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 <b>TransactionItem.Card</b> |
| 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 <b>DisplayMode</b>                                   |
| SubstateDisplay  | Описание подсостояния транзакции  |
| CardholderName   | Владелец платежной карты  |
| TerminalName     | Терминал  |



| Название            | Описание  |
|---------------------|---|
| ExternalPayment     | Данные оплаты по ссылке в представлении <b>TransactionItem .ExternalPayment</b> |
| CancelReturnTypes   | Список типов ввода, доступных для проведения отмены/возврата                    |
| ReceiptEmail        | Email для отправки чека   |
| ReceiptPhone        | Телефон для отправки чека   |
| Balance             | Баланс карты  |
| CanCancelCNP        | Транзакция может быть отменена в режиме CNP                                     |
| CanCancelCNPpartial | Транзакция может быть частично отменена в режиме CNP                            |
| RRN                 | Retrieval Reference Number  |
| SignatureRequired   | Признак необходимости отправки подписи  |
| FiscalInfo          | Фискальные данные платежа в представлении <b>TransactionItem .FiscalInfo</b>    |

#### Наборы параметров:

##### 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  |
|---------------|---|
| lin           | 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                                    |
| BankName      | Название банка  |
| BankCountryID |   |

## Класс TransactionItem.ExternalPayment

Вложенный класс **ExternalPayment**, дочерний класс **AbstractEntity**. Содержит данные о платеже по ссылке.

Свойства класса:

| Название | Описание  |
|----------|---|
| Type     | Способ отображения данных   |
| Link     | Ссылка для оплаты   |
| QR       | Набор пар вида <Название, Содержимое> для отображения в виде QR-кодов |

Наборы параметров:

Type

Способ отображения платежа

| Тип  | Описание          |
|------|-------------------|
| LINK | Отображать ссылку |
| QR   | Отображать QR-код |

## Класс TransactionItem.FiscalInfo

Дочерний класс **AbstractEntity**. Содержит фискальные данные платежа.

Свойства класса:

| Название              | Описание                  |
|-----------------------|---------------------------|
| CVC                   | КПК                       |
| FiscalDocumentNumber  | ФД                        |
| FiscalStorageNumber   | ФН                        |
| FiscalMark            | ФП                        |
| FiscalDeviceID        | ЗН ККТ                    |
| FiscalDeviceRegNumber | РН ККТ                    |
| FiscalDocSN           | Номер чека                |
| FiscalPrinterShift    | Номер смены               |
| FiscalDatetime        | Дата и время фискализации |
| FiscalStatus          | Статус фискализации       |

Наборы параметров:

### FiscalStatus

Статус фискализации

| Тип     | Описание            |
|---------|---------------------|
| NONE    | Не установлен       |
| CREATED | Фискализируется     |
| SUCCESS | Фискализована       |
| FAILURE | Ошибка фискализации |

## 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 <b>TransactionItem.Card</b> |

## Класс Account

Дочерний класс **AbstractEntity**. Содержит информацию об учетной записи

Свойства класса:

| Название           | Описание   |
|--------------------|--|
| SingleStepAuth     | Признак доступности одношаговой авторизации                                      |
| Name               | Имя агента   |
| BranchName         | Название филиала   |
| BranchAddress      | Адрес филиала  |
| BranchPhone        | Телефон филиала  |
| ClientName         | Название компании  |
| ClientLegalName    | Юридическое название компании  |
| ClientLegalAddress | Юридический адрес компании   |
| ClientRealAddress  | Фактический адрес компании   |
| ClientPhone        | Телефон компании   |
| ClientWeb          | Сайт компании  |
| BankName           | Название банка   |
| TerminalName       | Номер терминала  |
| AcquirersByMethods | Сгруппированные по доступным методам ввода пары типа <Код банка, Название банка> |
| ID                 | ID аккаунта  |
| PaymentOptions     | Набор разрешенных типов оплаты   |
| LinkedCards        | Набор привязанных карт   |
| Email              | Email аккаунта   |

## Класс Account.PaymentOption

Вложенный класс **Account**, дочерний класс **AbstractEntity**. Содержит информацию о типе оплаты, поддерживаемой банком.

Свойства класса:

| Название     | Описание       |
|--------------|----------------|
| InputType    | Тип оплаты     |
| Code         | Код банка      |
| AcquirerName | Название банка |

## Класс LinkedCard

Дочерний класс **AbstractEntity**. Содержит данные о привязанной платежной карте.

Свойства класса:

| Название  | Описание  |
|-----------|---|
| ID        | ID привязанной карты  |
| State     | Состояние   |
| isDeleted | Признак того, что карта более не привязана                        |
| Alias     | Имя карты для отображения   |
| PanEnding | Последние 4 цифры номера карты                                    |
| PanMasked | Первые и последние 4 цифры номера карты, разделенные символом “*” |
| Balance   | Баланс карты, используется только для операции привязки карты     |
| Bin       | Внутренний идентификатор банка                                    |

Наборы параметров:

### State

Состояние карты

| Тип      | Описание                    |
|----------|-----------------------------|
| DISABLED | Карта недоступна для работы |
| ENABLED  | Операции по карте разрешены |



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

#### getInProgressTransactions

|                       |  |
|-----------------------|--|
| Сигнатура             | ArrayList<TransactionItem> getInProgressTransactions() |
| Входные параметры     | Нет  |
| Возвращаемое значение | ArrayList транзакций                                   |
| Описание              | Возвращает набор транзакций, со статусом «В обработке» |

## Класс **APIAuthResult**

Дочерний класс **APIResult**.

**Методы класса:**

### **getAccount**

|                       |  |
|-----------------------|--|
| Сигнатура             | Account getAccount()                     |
| Входные параметры     | Нет                                      |
| Возвращаемое значение | Объект Account                           |
| Описание              | Возвращает информацию об учетной записи. |

### **getTaxID**

|                       |                             |
|-----------------------|-----------------------------|
| Сигнатура             | String getTaxID()           |
| Входные параметры     | Нет                         |
| Возвращаемое значение | ИНН пользователя            |
| Описание              | Возвращает ИНН пользователя |

## Класс `APIReadLinkedCardsResult`

Дочерний класс **`APIResult`**.

**Методы класса:**

`getLinkedCards`

|                       |   |
|-----------------------|---|
| Сигнатура             | <code>ArrayList&lt;LinkedCard&gt; getLinkedCards()</code> |
| Входные параметры     | Нет   |
| Возвращаемое значение | <code>ArrayList</code> привязанных карт                   |
| Описание              | Возвращает набор привязанных карт, 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 |

## Приложение 1: Печать слипа

Данные для слипа приходят в событии onFinished

Реквизиты клиента можно получить при помощи метода PaymentController.auth().

### Поля слипа:

| Название                   | Описание  |
|----------------------------|---|
| Банк                       | Account.getBankName()   |
| Название компании          | Account. getClientName ()   |
| Название юридического лица | Account. getClientLegalName ()  |
| Телефон компании           | Account. getClientPhone ()  |
| WEB-сайт компании          | Account. getClientWeb ()  |
| Дата и время операции      | paymentResultContext.getTransactionItem().getDate()   |
| Номер терминала            | paymentResultContext.getTransactionItem().getTerminalName()   |
| Номер чека                 | paymentResultContext.getTransactionItem().getInvoice()  |
| Код подтверждения          | result.TransactionItem.AcquirerApprovalCode   |
| Номер и тип карты          | paymentResultContext.getTransactionItem().getCard().getlin(), paymentResultContext.getTransactionItem().getCard().getPanMasked () |
| EMV тэги транзакции        | paymentResultContext.getEmvData(), печатаются в виде ключ-значение  |
| Тип операции               | paymentResultContext.getTransactionItem().getOperation()  |
| Сумма операции             | paymentResultContext.getTransactionItem().getAmount()   |
| Комиссия                   | 0.00 р.   |
| Статус                     | Успешно   |
| Подпись клиента            | Место для подписи в случае, если paymentResultContext.isRequiresSignature()==true, в ином случае «Подтверждено вводом PIN»        |

### Пример слипа:

ВТБ 24  
Тестовый клиент  
ООО "Тестовый клиент"  
+7 916 111 2233  
www.testclient.com  
Дата и время операции: 21.03.2017 15:47:34  
Терминал: II040001  
Чек: RM7ZEDMAAE7L  
Код подтверждения: SIMULATION  
Карта: mastercard \*\*\*\* 5631  
AID: A0000000041010  
TSI: 6800  
TVR: 8020008000  
Операция: Purchase  
Итого: 33 р  
Комиссия: 0.00 р  
Статус: Успешно  
Подтверждено вводом PIN кода.s