

IboxPro API

Руководство по интеграции .NET

V 1.7

История изменений

Версия	Дата	Описание
1.0.0	12.04.2016	Исходная версия
1.0.1	20.04.2016	Исправлено Transaction.Card, вызов события PaymentCancelled. Добавлено событие ReverseEvent.CantReverse
1.0.2	30.05.2016	Добавлен enum Currency, исправлены ошибки округления
1.0.3	04.08.2016	Добавлены частичные отмены/возвраты. Актуализирована процедура EMV платежей, добавлена однофакторная авторизация. Добавлены свойства класса Transaction. Общие улучшения
1.0.4	11.08.2016	Добавлена оплата наличными
1.1.0	09.12.2016	Изменен фреймворк на .NET4.5. Добавлены считыватели карт Wisepad_2 и QPos_mini.
1.2.0	21.03.2017	Добавлена генерация слип-чека Общие улучшения
1.3.0	10.05.2017	Поддержка NFC для ридера P17 Изменены названия ридеров
1.4.1	19.08.2017	Добавлен метод PaymentController.Auth, разрешено проведение отмены/возврата через чип/NFC. Теперь после ошибки платежа ридер не запрашивает карту повторно. Общие улучшения
1.4.2	29.08.2017	Добавлена оплата кредитом. Общие улучшения
1.4.3	13.09.2017	Добавлен метод PaymentController.GetTransactionById
1.4.4	25.09.2017	Добавлена оплата по ссылке, обновлена процедура платежа, добавлен enum PaymentMethod, добавлена установка банка(acquirer) для платежа, общие улучшения
1.4.6	13.10.2017	Исправлено проведение EMV/NFC транзакций для ридера QPOS.

		Добавлено свойство <code>PaymentController.ClientProductCode</code> . Обновлена сигнатура метода <code>PaymentController.StartReverse()</code> . Добавлены новые свойства для <code>PaymentContext</code> , <code>Transaction</code> и <code>Card</code>
1.4.7	1.11.2017	Данные для отображения QR в <code>ExternalPayment</code> теперь могут приходить в количестве более одного кода
1.4.9	07.11.2017	Добавлены транзакции «В обработке» для <code>APIGetHistoryResult</code>
1.4.10	09.11.2017	Добавлена поддержка TLSv1.2
1.4.16	31.01.2018	Исправления работы ридеров Исправлена работа метода <code>PaymentController.GetTransactionById</code> Добавлен делегат <code>PaymentController.CancellationTimeoutDelegate</code> (вызывается при попытке выполнения отмены платежа по истечению доступного для отмены таймаута)
1.4.17	22.02.2018	Исправлен вызов событий <code>ReaderEvent.StartInit</code> , <code>ReaderEvent.InitSuccessfully</code>
1.4.18	09.03.2018	Добавлен тип оплаты «Внешний POS-терминал», добавлена поддержка чиповых и NFC регулярных платежей, добавлена поддержка привязанных карт. Исправление ошибок.
1.4.19	13.03.2018	Добавлен класс <code>ReversePaymentContext</code> , добавлено поле <code>PaymentContext.ExtID</code>
1.4.20	10.04.2018	Добавлен класс <code>FiscalInfo</code> .
1.4.21	12.04.2018	Добавлено очищение контекста операции после успешного выполнения и при вызове метода <code>PaymentController.Disable()</code>
1.4.22	26.04.2018	Добавлено событие <code>ReaderEvent.CardTimeout</code> . Общие улучшения.

1.4.24	18.06.2018	В класс PaymentContext и ReversePaymentContext добавлено поле PaymentController.AuxData. Добавлено поле PaymentContext .AmountCashGot. Добавлены классы Purchase, Tax и TaxContribution. Добавлены поля для печати данных фискального чека в Transaction и Account
1.4.25	02.07.2018	Исправлены пустые поля в FiscalInfo, изменена логика получения TaxID в Account(Больше не связан с ИНН клиента)
1.4.26	04.07.2018	Общие улучшения
1.4.27	16.10.2018	Обновление ридеров QPOS
1.4.28	25.01.2019	Добавлена расширенная поддержка пользовательских продуктов: PaymentProduct и PaymentProductField. Подробнее о заполнении продуктов – см. описание PaymentContext.Prepare() и соответствующих классов. Добавлен метод PaymentContext.Prepare(). Добавлена константа для НДС20% в Purchase. Исправления работы ридера P17.
1.5.29	05.02.2019	Добавлена поддержка тегов для товаров согласно ФФД 1.05 (см. Purchase). Добавлен метод для отправки фискальных данных PaymentController.SubmitFiscal(). Добавлен метод PaymentController.Fiscalize(). Добавлен методы запроса состояния платежа PaymentController.TryGetPaymentStatus() и PaymentController.TryGetFiscalInfo().Добавлен класс APITryGetPaymentStatusResult. Добавлены методы поиска платежа PaymentController.GetTransactionByInvoice() и

		PaymentController.GetTransactionByRRN().
1.5.31	25.04.2019	Обновлен flow платежа/отмены платежа. Добавлена ошибка ErrorEvent.PaymentCanceled, удалено событие ReaderEvent.PaymentCanceled. Исправлен метод PaymentController.CancelPayment(). Добавлено поле Purchase.TitleAmount. Исправлена ошибка, возникавшая при создании регулярного платежа с дробной суммой.
1.7	25.06.2019	Общие улучшения

Содержание

История изменений	2
Обзор	8
Пространство имен Ibox.Pro.SDK.External	9
Перечисления(enums)	9
Структуры(structs)	10
Класс PaymentController	11
Пространство имен Ibox.Pro.SDK.External.Context	20
Перечисления(enums)	20
Класс PaymentContext	21
Класс RegularPaymentContext	23
Класс ReversePaymentContext	25
Класс PaymentResultContext	26
Пространство имен Ibox.Pro.SDK.External.Entry	27
Перечисления(enums)	27
Класс Transaction	29
Класс Transaction.Product	31
Класс Transaction.ProductField	32
Класс Account	33
Класс Card	34
Класс Schedule	36
Класс LinkedCard	37
Класс ExternalPayment	38
Класс FiscallInfo	39
Класс Purchase	40
Класс Tax	41
Класс TaxContribution	42
Класс PaymentProduct	43
Класс PaymentProductField	44
Класс PreparedField	45
Пространство имен Ibox.Pro.SDK.External.Result	46
Класс APIResult	46
Класс APIGetHistoryResult	47
Класс APIAuthResult	48
Класс APIReadLinkedCardsResult	49

Класс APIPrepareResult	50
Класс APITryGetPaymentStatusResult	51
Приложение 1: Печать слипа	52
Приложение 2: Пример чека	53

Обзор

Пространство имен Ibox.Pro.SDK.External

Перечисления(enums)

ReaderType

Набор поддерживаемых типов считывателей карт

Тип	Описание
P15	Считыватель карт «Chip&Pin», P15
P16	Считыватель карт «Chip&Pin», P16
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
Startinit	Начало инициализации
InitSuccessfully	Инициализация завершена успешно
InitFailed	Произошла ошибка инициализации
EjectCardTimeout	Не используется
SwipeCard	Обнаружено проведение магнитной полосой
TransactionStarted	Начата чиповая транзакция
WaitingForCard	Ожидание проведения магнитной полосой или вставки чиповой карты
EjectCard	Пользователь может извлечь карту (возникает при ошибке проведения транзакции)
BadSwipe	Не удалось считать данные магнитной ленты
LowBattery	Уровень заряда батареи считывателя карт менее 10%
CardTimeout	Превышен таймаут ожидания карты

PaymentError

Набор возможных ошибок, которые могут возникнуть в процессе выполнения платежа

Тип	Описание
ConnectionError	Ошибка соединения с сервером
ServerError	Ошибка выполнения транзакции
PaymentRuntimeError	Ошибка выполнения платежа
TransactionNullOrEmpty	Ошибка создания транзакции
EmvError	
EmvTerminated	
EmvDeclined	
EmvCancel	

Тип	Описание
EmvCardError	
EmvCardBlocked	
EmvDeviceError	
EmvCardNotSupported	
EmvZeroTranEmv	
EmvNotAllowed	Чиповая транзакция не разрешена
SwipeNotAllowed	Проведение транзакции магнитной лентой не разрешено
NfcNotAllowed	Проведение NFC транзакции не разрешено
PaymentCanceled	Платеж отменен пользователем

ReverseEvent

События изменения состояния транзакции отмены(возврата)

Тип	Описание
TransactionNotFound	Транзакция не была найдена, либо не уникальна
SwitchedToCNP	Отмена транзакции будет выполнена в режиме CNP
CantReverse	Для транзакции не может быть выполнена операция отмены(возврата)
InvalidInputType	Неправильный тип ввода

Currency

Валюты, которыми можно провести оплату

Тип	Описание
RUB	Российский рубль
VND	Вьетнамский донг

Структуры(structs)

Credentials

Данные авторизации

Тип	Описание
Email	Email пользователя
Password	Пароль пользователя

Класс PaymentController

Данный класс является центральным в библиотеке. Содержит методы для создания и отмены(возврата) транзакций, передачи в них дополнительных параметров а также инкапсулирует работу со считывателями карт. Также с помощью этого класса осуществляется передача дополнительных данных платежа и получение истории платежей.

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**. Иначе будет сгенерировано исключение **InvalidOperationException**. Также необходимо вызвать метод **Enable**. Вызов метода **Disable** приведет к отключению считывателя карт, но НЕ прервет обработку текущей транзакции. Если перед началом платежа не будут определены свойства **SelectApplicationDelegate**, **ConfirmScheduleDelegate** и **ScheduleCreationFailedDelegate** или не будет вызван метод **Enable** – будет сгенерировано исключение **InvalidOperationException**. При указании суммы платежа, разрядность десятичной части которой превышает разрядность десятичной части валюты, количество знаков после запятой такой суммы будет урезано **без округления**. Перед началом выполнения операции необходимо дождаться события **ReaderEvent.InitSuccessfully**.

После окончания транзакции, в зависимости от результата, будет инициировано событие **TransactionFinishedEvent** или **ErrorEvent**.

Свойства:

Название	Описание
Instance	Экземпляр класса
IsPaymentInProgress	true , если платеж все еще выполняется
SinglestepEMV	Признак однофакторной авторизации
SelectApplicationDelegate	Обработчик выбора приложения карты
ConfirmScheduleDelegate	Обработчик подтверждения регулярного платежа
ScheduleCreationFailedDelegate	Обработчик запроса повторной попытки создания расписания
CancellationTimeoutDelegate	Обработчик попытки отмены платежа по истечению доступного для отмены таймаута
Credentials	Параметры авторизации
ClientProductCode	Код клиентского приложения

События:

Название	Описание
TransactionStartedEvent	Начало выполнения транзакции
TransactionFinishedEvent	Успешное окончание выполнения транзакции
ReaderEvent	Изменилось состояние считывателя карт
ErrorEvent	При выполнении платежа произошла ошибка
ReverseEvent	Состояние транзакции отмены(возврата) изменилось

Делегаты:

SelectApplicationCallback

Сигнатура	Int SelectApplicationCallback(List<string> apps)
Входные параметры	apps – список названий приложений
Возвращаемое значение	Порядковый номер выбранного приложения (начиная с 0)
Описание	Вызывается при выполнении чиповой транзакции, если чиповая карта содержит более 1 приложения

ConfirmScheduleCallback

Сигнатура	bool ConfirmScheduleCallback(List<KeyValuePair<DateTime, decimal>> steps, decimal 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	Вызывается при создании регулярного платежа

ScheduleCreationFailedCallback

Сигнатура	bool ScheduleCreationFailedCallback(PaymentError error, string description = null)
Входные параметры	error – тип ошибки description – сообщение об ошибке. Используется только когда error == SERVER_ERROR
Возвращаемое значение	true , если необходимо повторить попытку создания расписания
Описание	Вызывается в случае возникновения ошибки при создании расписания регулярного платежа

SelectInputTypeCallback

Сигнатура	Entry.InputType SelectInputTypeCallback(List<Entry.InputType> allowedInputTypes)
Входные параметры	allowedInputTypes – разрешенные типы ввода
Возвращаемое значение	Выбранный тип ввода
Описание	Вызывается в случае необходимости выбора типа ввода перед проведением транзакции отмены/возврата

CancellationTimeoutCallback

Сигнатура	bool SelectInputTypeCallback()
Входные параметры	нет
Возвращаемое значение	true , для выполнения возврата платежа

Описание	Вызывается в случае попытки выполнения отмены платежа по истечению доступного для отмены таймаута
----------	---

TransactionStartedEventHandler

Сигнатура	void TransactionStartedEventHandler(string transactionID)
Входные параметры	transactionID – ID созданной транзакции
Возвращаемое значение	нет
Описание	Обработчик события TransactionStartedEvent

TransactionFinishedEventHandler

Сигнатура	void TransactionStartedEventHandler(PaymentResultContext result)
Входные параметры	result – данные проведенной транзакции в представлении PaymentResultContext
Возвращаемое значение	нет
Описание	Обработчик события TransactionFinishedEvent

ReaderEventHandler

Сигнатура	void ReaderEventHandler(ReaderEvent readerEvent)
Входные параметры	readerEvent – событие считывателя карт
Возвращаемое значение	нет
Описание	Обработчик события ReaderEvent

ErrorEventHandler

Сигнатура	void ErrorHandler(PaymentError error, string description = null)
Входные параметры	error – возникнувшая ошибка description – описание
Возвращаемое значение	нет
Описание	Обработчик события ErrorEvent

ReverseEventHandler

Сигнатура	void ReverseEventHandler(ReverseEvent reverseEvent, string description = null)
Входные параметры	event – событие description – описание
Возвращаемое значение	нет
Описание	Обработчик события ReverseEvent

Методы класса:

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

SetReaderType

Сигнатура	void SetReaderType(ReaderType readerType, string readerBTPort = null)
Входные параметры	readerType – тип считывателя карт readerBTPort – название COM порта, через который подключен Bluetooth-ридер.
Возвращаемое значение	Нет
Описание	Изменяет тип текущего считывателя карт. При попытке изменить тип считывателя во время проведения платежа будет сгенерировано исключение InvalidOperationException

GetReaderType

Сигнатура	ReaderType? GetReaderType()
Входные параметры	Нет
Возвращаемое значение	Текущий тип считывателя карт
Описание	Возвращает текущий тип считывателя карт

StartPayment

Сигнатура	void StartPayment(PaymentContext paymentContext)
Входные параметры	paymentContext – данные платежа
Возвращаемое значение	Нет
Описание	Начинает выполнение платежа. Перед вызовом необходимо установить данные пользователя, свойства SelectApplicationDelegate , ConfirmScheduleDelegate , ScheduleCreationFailedDelegate и убедиться, что тип считывателя карт задан, а сам считыватель подключен, иначе будет сгенерировано исключение InvalidOperationException . При попытке начать новый

	платеж/отмену платежа до окончания предыдущего будет сгенерировано исключение InvalidOperationException .
--	--

StartReverse

Сигнатура	void StartReverse(string transactionID, ReverseMode mode, decimal? amountToReverse = null, string receiptEmail = null, string receiptPhone = null)
Входные параметры	transactionID – ID транзакции отменяемого платежа mode – тип отмены amountToReverse – сумма, на которую будет выполнена отмена/возврат. Для полной отмены/возврата передавать null receiptEmail – email для отправки чека receiptPhone – номер телефона для отправки чека
Возвращаемое значение	Нет
Описание	Устаевший. См. StartReverse(ReversePaymentContext reversePaymentContext)

StartReverse

Сигнатура	void StartReverse(ReversePaymentContext reversePaymentContext)
Входные параметры	reversePaymentContext – данные для проведения отмены/возврата
Возвращаемое значение	Нет
Описание	Начинает выполнение отмены платежа. Перед вызовом необходимо установить данные пользователя и убедиться, что тип считывателя карт задан, а сам считыватель подключен, иначе будет сгенерировано исключение InvalidOperationException . При попытке начать новый платеж/отмену платежа до окончания будет сгенерировано исключение InvalidOperationException .

Adjust

Сигнатура	APIResult Adjust(string transactionID, string email, string phone)
Входные параметры	transactionID – ID транзакции, для которой требуется отправить дополнительные данные email – email для отправки чека phone – номер телефона для отправки чека
Возвращаемое значение	Результат отправки данных
Описание	Используется для отправки подписи и чека для транзакции единичного платежа

AdjustRegular

Сигнатура	APIResult AdjustRegular(string transactionID, string email, string phone)
Входные параметры	transactionID – ID транзакции, для которой требуется отправить дополнительные данные email – email для отправки чека phone – номер телефона для отправки чека
Возвращаемое значение	Результат отправки данных
Описание	Используется для отправки подписи и чека для регулярного платежа

AdjustReverse

Сигнатура	APIResult AdjustReverse(string transactionID, string email, string phone)
Входные параметры	transactionID – ID транзакции, для которой требуется отправить дополнительные данные email – email для отправки чека phone – номер телефона для отправки чека
Возвращаемое значение	Результат отправки данных
Описание	Используется для отправки подписи и чека для транзакции отмены(возврата) платежа

GetHistory

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

GetTransactionByID

Сигнатура	APIGetHistoryResult GetTransactionByID(string transactionID)
Входные параметры	transactionID – ID целевой транзакции
Возвращаемое значение	Объект APIGetHistoryResult, содержащий набор транзакций
Описание	Позволяет получить данные транзакции по ее ID

Auth

Сигнатура	APIAuthResult Auth()
Входные параметры	Нет
Возвращаемое значение	APIAuthResult

Описание	Проверяет правильность введенных учетных данных и возвращает информацию об учетной записи
----------	---

AddLinkedCard

Сигнатура	void AddLinkedCard(Currency currency, String acquirerCode = null)
Входные параметры	currency – валюта карты acquirerCode – код банка
Возвращаемое значение	Нет
Описание	Начинает выполнение привязки карты. Перед вызовом необходимо установить данные пользователя и убедиться, что тип считывателя карт задан, а сам считыватель подключен, иначе будет сгенерировано исключение InvalidOperationException . При попытке начать привязку карты до окончания текущего платежа или привязки карты будет сгенерировано исключение InvalidOperationException . Асинхронный, см. StartPayment().

RemoveLinkedCard

Сигнатура	APIResult RemoveLinkedCard(int cardID)
Входные параметры	cardID – ID удаляемой карты
Возвращаемое значение	Результат операции
Описание	Удаляет привязанную карту для текущего аккаунта. Синхронный

ReadLinkedCards

Сигнатура	APIReadLinkedCardsResult ReadLinkedCards()
Входные параметры	Нет
Возвращаемое значение	Объект APIReadLinkedCardsResult, содержащий набор привязанных карт
Описание	Запрашивает набор привязанных карт для текущего аккаунта. Перед вызовом необходимо установить данные пользователя и убедиться, что ранее были получены данные аккаунта с помощью метода Auth(), иначе будет сгенерировано исключение InvalidOperationException . Синхронный.

Prepare

Сигнатура	APIPrepareResult ReadLinkedCards(string productCode, Dictionary<string, string> fields)
Входные параметры	context – контекст приложения productCode – код продукта, для которого требуется заполнить поля fields – значения preparable-полей продукта, по которым должны заполниться остальные поля
Возвращаемое значение	Результат операции
Описание	Выполняет заполнение полей для пользовательского продукта, имеющего свойство preparable, согласно данным, указанным в его preparable-полях. Рекомендуется заполнять значениями только preparable-полей. Синхронный.

SubmitFiscal

Сигнатура	APIResult SubmitFiscal(string transactionID, string printerID, int shift, int docSN, string fdn, string fdm, string fs, DateTime fdt)
Входные параметры	transactionID – ID транзакции, для которой требуется отправить фискальные данные printerID – ID фискального регистратора shift – номер операционной смены docID – сквозной номер документа fdn – фискальный номер документа fdm – фискальный признак документа fs – фискальное хранилище fdt – дата и время проведения фискальной операции
Возвращаемое значение	Результат отправки данных
Описание	Выполняет отправку данных фискализации согласно стандарту Ф3-54. Синхронный.

Fiscalize

Сигнатура	APITryGetPaymentStatusResult Fiscalize(string transactionID)
Входные параметры	transactionID – ID транзакции
Возвращаемое значение	Результат запроса, содержащий в себе TransactionItem с обновленными данными
Описание	Иницирует повторный запрос серверной фискализации. Синхронный.

TryGetPaymentStatus

Сигнатура	APITryGetPaymentStatusResult TryGetPaymentStatus(string transactionID)
Входные параметры	transactionID – ID транзакции
Возвращаемое значение	Результат запроса, содержащий в себе TransactionItem с обновленными данными
Описание	Позволяет выполнить запрос состояния оплаты транзакции. Используется для платежей «по ссылке». Синхронный.

TryGetFiscalInfo

Сигнатура	APITryGetPaymentStatusResult TryGetFiscalInfo(string transactionID)
Входные параметры	transactionID – ID транзакции
Возвращаемое значение	Результат запроса, содержащий в себе TransactionItem с обновленными данными
Описание	Позволяет выполнить запрос фискального состояния и фискальных данных транзакции. Синхронный.

GetTransactionByInvoice

Сигнатура	APIGetHistoryResult getTransactionByInvoice(string invoice)
Входные параметры	invoice – номер чека
Возвращаемое значение	Объект APIGetHistoryResult, содержащий запрашиваемую транзакцию
Описание	Позволяет получить данные транзакции по номеру чека

GetTransactionByRRN

Сигнатура	APIGetHistoryResult getTransactionByRRN(string rrn)
Входные параметры	rrn – Retrieval Reference Number
Возвращаемое значение	Объект APIGetHistoryResult, содержащий запрашиваемую транзакцию
Описание	Позволяет получить данные транзакции по ее RRN

CancelPayment

Сигнатура	void CancelPayment()
Входные параметры	Нет
Возвращаемое значение	Нет
Описание	Позволяет прервать выполнение транзакции, если карта еще не была вставлена.

Пространство имен Ibox.Pro.SDK.External.Context

Перечисления(enums)

RepeatType

Набор возможных типов регулярного платежа

Тип	Описание
DelayedOnce	Платеж будет выполнен один раз
Weekly	Еженедельный платеж
Monthly	Ежемесячный платеж
Quarterly	Ежеквартальный платеж
Annual	Ежегодный платеж
ArbitraryDays	Платеж будет выполняться в заданные дни

EndType

Набор возможных способов окончания выполнения регулярного платежа

Тип	Описание
ByQuantity	Окончание по количеству повторов
AtDay	Окончание в заданный день

ReverseMode

Набор возможных способов отмены платежа

Тип	Описание
Cancel	Отмена платежа
Return	Возврат платежа
CancelCNP	Отмена в режиме CNP

PaymentMethod

Набор возможных способов проведения платежа

Тип	Описание
Card	Оплата картой
Credit	Оплата кредитом
Cash	Оплата наличными
Other	Оплата по ссылке
LinkedCard	Оплата привязанной картой
OuterCard	Оплата внешним POS-терминалом

Класс PaymentContext

Контейнер данных, необходимых для выполнения разового платежа. Обратите внимание, что разница Summ(AuxData) - Amount суммы перечня товаров в поле AuxData и суммы платежа Amount будет учтена как предоплата. При Amount > Summ(AuxData) будет сгенерировано исключение **InvalidOperationException**.

Свойства класса:

Название	Описание
Amount	Сумма платежа
Currency	Валюта платежа
Description	Описание платежа
Image	Изображение, прикрепленное к платежу
PaymentProductCode	Код продукта в платеже
PaymentProductTextDictionary	Набор значений текстовых полей продукта в формате <Код поля, значение>
PaymentProductImageDictionary	Набор изображений для графических полей продукта в формате <Код поля, изображение>
Cash	Признак оплаты наличными (устаревший)
Credit	Признак оплаты в кредит (устаревший)
AcquirerCode	Код банка
Method	Способ проведения платежа
ReceiptEmail	Email для отправки чека
ReceiptPhone	Телефон для отправки чека
LinkedCardID	ID привязанной карты
ExtID	ID клиентского приложения
AuxData	Перечень товаров в установленном формате
AmountCashGot	Принято наличными

Методы класса:

Clear

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

PutPurchase

Сигнатура	bool PutPurchase(Purchase purchase)
Входные параметры	purchase – данные о товаре
Возвращаемое значение	true , если товар был успешно добавлен
Описание	Используйте этот метод для добавления данных о товаре в стандартном формате Purchase

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**. При создании регулярного платежа с продуктом, для которого применимы только управляемые регулярные платежи (**RecurrentMode == MANAGED**), необходимо установить поле **Managed true**. Остальные поля должны быть заполнены параметрами, необходимыми для старта платежа. В явном виде:

```
regPaymentContext.Managed = true;  
regPaymentContext.PaymentRepeatType = RepeatType.DelayedOnce;  
regPaymentContext.PaymentEndType = EndType.ByQuantity;  
regPaymentContext.RepeatCount = 1;  
regPaymentContext.StartDate = DateTime.Now;  
regPaymentContext.EndDate = DateTime.Now;  
regPaymentContext.ArbitraryDays = null;  
regPaymentContext.Day = 0;  
regPaymentContext.DayOfWeek = 0;  
regPaymentContext.Hour = 0;  
regPaymentContext.Minute = 0;  
regPaymentContext.Month = 0;
```

Свойства класса:

Название	Описание
PaymentRepeatType	Тип регулярного платежа
PaymentEndType	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]
MonthOfQuarter	Месяц для выполнения ежеквартального платежа, [1,4]
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
Managed	Признак того, что настройки регулярного платежа будут определены хостом

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), MonthOfQuarter, 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

Контейнер данных, необходимых для выполнения отмены(возврата). Обратите внимание, что проверка корректности содержимого AuxData[Purchases] остается за конечным пользователем. Разница Summ(AuxData) - Amount суммы перечня товаров в поле AuxData и суммы отмены(возврата) Amount будет учтена как предоплата

Свойства класса:

Название	Описание
TransactionID	ID транзакции отменяемого платежа
Mode	Тип операции
AmountToReverse	сумма, на которую будет выполнена отмена. Для полной отмены установить null
ExtID	ID клиентского приложения
ReceiptEmail	Email для отправки чека
ReceiptPhone	Телефон для отправки чека
AuxData	Перечень товаров для отмены(возврата) в установленном формате

Методы класса:

Clear

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

PutPurchase

Сигнатура	bool PutPurchase(Purchase purchase)
Входные параметры	purchase – данные о товаре
Возвращаемое значение	true , если товар был успешно добавлен
Описание	Используйте этот метод для добавления данных о товаре в стандартном формате Purchase

Класс PaymentResultContext

Контейнер данных, полученных при успешном проведении платежа или отмене(возврате) платежа или при успешной привязке карты.

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>
AttachedCard	Данные привязанной карты в представлении LinkedCard .

Пространство имен Ibox.Pro.SDK.External.Entry

Перечисления(enums)

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
OuterCard	Оплата внешним POS-терминалом
Prepaid	Предоплата
Cash	Оплата наличными
Credit	Оплата в кредит
Other	Оплата по ссылке

DisplayMode

Тип отображения транзакции

Тип	Описание
Declined	Отклоненная транзакция
Success	Успешная транзакция
Reverse	Транзакция отмены/возврата
Reversed	Платеж отменен/выполнен возврат
NonFinancial	

FiscalStatus

Статус фискализации

Тип	Описание
None	Не установлен
Created	Фискализируется
Success	Фискализирована
Failure	Ошибка фискализации

TaxMode

Режим начисления налогов

Тип	Описание
ForEach	Начисление налогов по позициям
ForTotal	Начисление налогов на весь чек

ProductItemState

Состояние пользовательского продукта или его полей

Тип	Описание
Disabled	Продукт(поле) отключе
Enabled	Продукт(поле) доступен

Класс Transaction

Объектное представление транзакции. Содержит набор свойств, определяющих ее. Имеет вложенные классы.

Свойства класса:

Название	Описание
ID	ID транзакции
Date	Время и дата выполнения транзакции, согласно GMT устройства
Description	Описание транзакции
Invoice	Номер чека
Terminal	Номер терминала
AcquirerApprovalCode	Код подтверждения транзакции
ScheduleID	ID регулярного платежа
ScheduleStepID	ID списания для рекуррентного платежа
Amount	Сумма 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
RequiresSignature	Признак требования подписи. False, если оплата была подтверждена вводом PIN кода
HasSignature	Признак наличия приложенной подписи
SignatureUrl	URL приложенной подписи
StateDisplay	Описание состояния транзакции
CardholderName	Имя держателя карты
Card	Данные карты, которая была использована для оплаты, в представлении Card
EMVData	Список EMV тэгов карты, необходимых для печати в слипе
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
Canceled	Признак того, что для платежа выполнена отмена или возврат
DisplayMode	Тип отображения транзакции в представлении DisplayMode

Название	Описание
SubstateDisplay	Описание подсостояния транзакции
CancelInputTypes	Доступные типы ввода для проведения отмены/возврата
CurrencyID	ID валюты операции
ExternalPayment	Данные оплаты по ссылке в представлении ExternalPayment
ReceiptEmail	Email для отправки чека
ReceiptPhone	Телефон для отправки чека
CanCancelCNP	Транзакция может быть отменена в режиме CNP
CanCancelCNPpartial	Транзакция может быть частично отменена в режиме CNP
AmountCashGot	Принято наличными
AuxData	Перечень товаров в установленном формате
TaxMode	Режим начисления налогов, примененный при фискализации
TaxContributions	Налоговые начисления в представлении TaxContribution , без учета товаров в AuxData
Taxes	Перечень налогов в представлении Tax , примененных в транзакции, без учета товаров в AuxData
TaxSystemName	Система налогообложения, примененная при фискализации
FiscalInfo	Фискальные данные платежа в представлении FiscalInfo
Products	Привязанные к платежу пользовательские продукты

Класс Transaction.Product

Содержит данные привязанного к транзакции пользовательского продукта.

Свойства класса:

Название	Описание
Description	Описание продукта в представлении PaymentProduct
Fields	Значения полей продукта

Класс Transaction.ProductField

Содержит данные полей привязанного к транзакции пользовательского продукта.

Свойства класса:

Название	Описание
Description	Описание поля в представлении PaymentProductField
TextValue	Значение текстового поля
ImageUrl	Ссылка на изображение для поля с изображением

Класс Account

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>
LinkedCards	Привязанные карты
TaxID	ИНН агента

Класс Card

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
BankName	Название банка
BankCountryID	

Методы класса:

IsCash

Сигнатура	IsCash()
Входные параметры	Нет
Возвращаемое значение	true , если оплата была проведена наличными
Описание	Возвращает признак того, что оплата была проведена наличными

IsPrepaid

Сигнатура	IsPrepaid()
Входные параметры	Нет
Возвращаемое значение	true , если операция выполнена по предоплате
Описание	Возвращает признак того, что оплата была проведена по предоплате

IsCredit

Сигнатура	IsCredit()
Входные параметры	Нет
Возвращаемое значение	true , если операция выполнена в кредит
Описание	Возвращает признак того, что оплата была проведена в кредит

IsOuter

Сигнатура	IsOuter()
Входные параметры	Нет

Возвращаемое значение	true , если операция выполнена через внешний POS-терминал
Описание	Возвращает признак того, что оплата была проведена через внешний POS-терминал

Класс Schedule

Является объектным представлением данных регулярного платежа

Свойства класса:

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

Класс LinkedCard

Содержит данные о привязанной карте.

Свойства класса:

Название	Описание
ID	ID привязанной карты
State	Состояние
Deleted	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 “*”

Класс ExternalPayment

Является объектным представлением данных 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 FiscalInfo

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

Класс Purchase

Данные товара в стандартном формате. Внимание! При наличии в продукте тегов ФФД соответствующие им значения свойств, указанные через поля "Title", "Price", "Quantity", "TaxCode" будут проигнорированы. Т.е. запрос, к примеру, свойства "Title" будет возвращать значение тега 1030, если такой будет указан. Иначе будет возвращено значение свойства "Title". Описание формата ФФД 1.05 см. Приложение № 2 к приказу ФНС России от 21 марта 2017 г. № ММВ-7-20/229. При построении JSON экземпляра класса в формате ФФД 1.05 необходимо придерживаться следующих правил по выбору типов данных:

- int (теги типа *Целое* и *Флаги*)
- decimal (только тег 1079 и тег 1023)
- string – все остальные

Константы:

Название	Описание
VAT_NA	Код налоговой ставки «Без НДС»
VAT_0	Код налоговой ставки «НДС 0%»
VAT_10	Код налоговой ставки «НДС 10%»
VAT_18	Код налоговой ставки «НДС 18%»
VAT_20	Код налоговой ставки «НДС 20%»

Свойства класса:

Название	Описание
Title	Название товара
Price	Цена за единицу товара
Quantity	Количество
TaxCode	Перечень кодов применяемых налоговых ставок
TitleAmount	Сумма позиции с учетом надбавок(тег 1043)

Методы класса:

GetTags

Сигнатура	Dictionary<string, string> GetTags()
Входные параметры	Нет
Возвращаемое значение	Представление товара в форме перечня <тег, значение> согласно ФФД 1.05
Описание	Позволяет получить представление товара в форме перечня <тег, значение> согласно ФФД 1.05

Класс Tax

Содержит данные о налоговой ставке

Свойства класса:

Название	Описание
Code	Код ставки
Name	Название
Rate	Значение ставки

Класс TaxContribution

Содержит данные о налоговом начислении по ставке

Свойства класса:

Название	Описание
Code	Код ставки
Name	Название
Rate	Значение ставки
Total	Сумма начислений

Класс PaymentProduct

Содержит данные о пользовательском продукте. Внимание!

При создании регулярных платежей для продуктов с RecurrentMode == MANAGED дополнительно ознакомьтесь с описанием класса **RegularPaymentContext**.

Свойства класса:

Название	Описание
Code	Уникальный код продукта
State	Состояние продукта
Apply	Доступные типы платежа
Title	Название продукта
TitleReceipt	Название продукта для печати в чеке
Fields	Поля продукта
Preparable	Признак того, что заполнение полей продукта должно выполняться результатом вызова метода PaymentController.prepare().
PreparableEditable	Признак того, что разрешено редактирование загруженных значений полей
PreparableOptional	Признак того, что значения полей могут быть заполнены без вызова метода PaymentController.prepare().
RecurrentMode	Настройка регулярного платежа

Наборы параметров:

PaymentType

Доступные типы платежа для продукта

Тип	Описание
Payment	Только обычный платеж
Recurrent	Только регулярный платеж
Both	Оба типа платежа
None	Платежи недоступны

ProductRecurrentMode

Настройка регулярного платежа для продукта

Тип	Описание
Undefined	Не определено
Simple	График и настройки регулярного платежа передаются при создании платежа
Managed	График и настройки регулярного платежа устанавливаются сервером и не передаются при платеже

Класс PaymentProductField

Содержит данные о поле пользовательского продукта. При формировании платежа с продуктом рекомендуется передавать в том числе и недоступные для пользователя (!UserVisible) поля со значениями по умолчанию.

Свойства класса:

Название	Описание
Type	Тип поля
Code	Уникальный код поля
Required	Обязательное поле
Preparable	Значение поля используется при вызове метода PaymentController.prepare().
Title	Название поля
TitleReceipt	Название поля для печати в чеке
DefaultValue	Значение по умолчанию
TextMask	Маска ввода(регулярное выражение)
TextRegExp	Регулярное выражение для проверки правильности ввода
Multiline	Признак многострочности поля
Numeric	Признак того, что значение поля является десятичным числом
ReceiptEmail	Признак того, что значение поля должно использоваться как email для отправки чека
ReceiptPhone	Признак того, что значение поля должно использоваться как телефон для отправки чека
UserVisible	Признак того, что поле доступно для пользователя
PrintInReceipt	Признак того, что поле должно быть напечатано в чеке

Наборы параметров:

Type

Тип поля

Тип	Описание
Text	Текстовое поле
Image	Поле с изображением

Класс PreparedField

Обертка для пар код/значение, получаемых в результате PaymentController.Prepare()

Свойства класса:

Название	Описание
Code	Код поля
Value	Значение поля
IsPaymentAmount	Признак того, что значение поля должно быть использовано как сумма платежа

Пространство имен Ibox.Pro.SDK.External.Result

Класс APIResult

Является примитивной сущностью, содержащую ответ от сервера

Свойства класса:

Название	Описание
ErrorCode	Код ошибки. 0 – если ответ не содержит сообщений об ошибке.
ErrorMessage	Сообщение об ошибке

Класс **APIGetHistoryResult**

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

Свойства класса:

Название	Описание
Transactions	Набор транзакций, содержащихся в ответе
InProcessTransactions	Набор транзакций со статусом «В обработке»
Products	Набор пользовательских продуктов

Класс **APIAuthResult**

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

Свойства класса:

Название	Описание
Account	Информация об учетной записи

Класс **APIReadLinkedCardsResult**

Дочерний класс **APIResult**. Содержит набор привязанных карт.

Свойства класса:

Название	Описание
LinkedCards	Набор привязанных карт, содержащихся в ответе

Класс APIPrepareResult

Дочерний класс **APIResult**.

Свойства класса:

Название	Описание
Fields	Перечень полученных значений для полей пользовательского продукта

Класс `APITryGetPaymentStatusResult`

Дочерний класс `APIResult`.

Свойства класса:

Название	Описание
Transaction	Транакция с обновленными данными

Приложение 1: Печать слипа

Данные для слипа приходят в событии onPaymentFinished

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

Поля слипа:

Название	Описание
Банк	Account.BankName
Название компании	Account. ClientName
Название юридического лица	Account. ClientLegalName
Телефон компании	Account. ClientPhone
WEB-сайт компании	Account. ClientWeb
Дата и время операции	result.TransactionItem.Date
Номер терминала	result.TransactionItem.TerminalName
Номер чека	result.TransactionItem.Invoice
Код подтверждения	result.TransactionItem.AcquirerApprovalCode
Номер и тип карты	result.TransactionItem.Card.IIN, result.TransactionItem.Card.PANMasked
EMV тэги транзакции	result.TransactionItem.EMVData, печатаются в виде ключ-значение
Тип операции	result.TransactionItem.Operation
Сумма операции	result.TransactionItem.Amount
Комиссия	0.00 р.
Статус	Успешно
Подпись клиента	Место для подписи в случае, если result.TransactionItem.RequiresSignature==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.

Приложение 2: Пример чека

Позиция 1 – Без НДС

Позиция 2 – НДС 10%

Позиция 3 – НДС 18%

```
-----ЧЕК-----
ПРИХОД
КАССИР          Тестовый агент с принтером
ДАТА И ВРЕМЯ      01.02.18 12:00:11
НОМЕР ЧЕКА        ABCDEFGHIJKL
КОД ПОДТВЕРЖДЕНИЯ 123456ABCD78
-----
Позиция 1          2,000x111,26=222,52
Позиция 2          1,000x14,00=14,00
Позиция 3          2,990x43,22=129,23
-----
ИТОГ                365,75
-----
НАЛИЧНЫМИ          365,75
ЭЛЕКТРОННЫМИ        0,00
ПРЕДВАРИТЕЛЬНАЯ ОПЛАТА (АВАНС)  0,00
ПОСЛЕДУЮЩАЯ ОПЛАТА (КРЕДИТ)     0,00
СДАЧА                0,00
Сумма без НДС        222,52
Сумма по НДС 0%      0,00
Сумма НДС 10%        1,27
Сумма НДС 18%        19,71
-----ФИСК. БЛОК-----
14.03.18 14:58
ИНН                  123567890
СНО                  ОЧН
САЙТ ФНС             http://nalog.ru
СМЕНА №              188
ЧЕК №                1
ЗН ККТ               000000000011111
РН ККТ               000000002222222
ФН                   999999999999999
ФД                   0000003333
ФПД                  4444444444
-----КОНЕЦ ФИСК. БЛОКА-----
-----КОНЕЦ ЧЕКА-----
```