

Руководство по установке и эксплуатации

«DucatPay»

Содержание

Список сокращений	4
Список терминов	5
1 Введение	6
1.1 О сервисе	6
1.2 Системные требования.....	7
1.3 Уровень подготовки пользователя.....	8
2 Подготовка к работе	9
2.1 Регистрация организации в DucatPay	9
2.2 Получение логина и пароля.....	9
2.3 Авторизация в сервисе.....	10
2.4 Создание магазина.....	11
2.5 Получение API-ключа и ID организации	12
3 Платежи	13
3.1 Платежная форма DucatPay	14
3.1.1 Порядок взаимодействия с платежной формой DucatPay.....	15
3.1.2 Атрибуты для формирования платежной формы	19
3.2 Платежная форма Мерчанта.....	20
3.2.1 Порядок взаимодействия с платежной формой Мерчанта	20
3.3 Возврат платежей	23
3.3.1 Возврат платежа в ЛК DucatPay	23
3.3.2 Возврат платежа через API.....	25
3.4 Рекуррентные платежи.....	26
3.4.1 Порядок проведения рекуррентного платежа	26
4 Выплаты	28
4.1 Подготовительный этап	28
4.1.1 Создание цифровой личности.....	28
4.1.2 Создание виртуального кошелька	29
4.1.3 Пополнение баланса виртуального кошелька	29
4.2 Процесс проведения выплаты	30
5 Личный кабинет	33
5.1 Меню «Платежи».....	40

5.1.1	Раздел «Магазины»	40
5.1.2	Раздел «Аналитика»	42
5.1.3	Раздел «Операции».....	44
5.1.4	Раздел «Отчеты».....	52
5.1.5	Раздел «Интеграция»	56
5.2	Меню «Выплаты»	59
5.2.1	Раздел «Кошельки»	59
5.2.2	Раздел «Пополнения»	61
5.2.3	Раздел «Операции».....	62
5.2.4	Раздел «Интеграция»	64
6	Спецификация API.....	67
6.1	Группа «Платежи».....	70
6.2	Группа «Выплаты»	87

Список сокращений

Сокращение	Расшифровка
БИК	Банковский идентификационный код
ВДС	Виртуальные денежные средства
ЕИО	Единоличный исполнительный орган
ЛК	Личный кабинет
ПО	Программное обеспечение
3DS	3D-Secure — протокол двухфакторной аутентификации Покупателя при проведении платежа
API	Application programming interface — программный интерфейс приложения, который позволяет Мерчанту взаимодействовать с DucatPay
HTTPS	HyperText Transfer Protocol Secure — безопасный протокол передачи данных в сети Интернет
ID	Уникальный идентификатор
PCI DSS	Payment Card Industry Data Security Standard — международный стандарт безопасности, для защиты данных платежных карт
RRN	Идентификатор банковской транзакции
URL	Uniform Resource Locator — адрес ресурса в сети Интернет

Список терминов

Термин	Определение
DucatPay	Процессинговый сервис, предоставляющий услуги интернет-эквайринга для электронной коммерции и онлайн-торговли
iFrame	Контейнер, который позволяет вставить любой HTML-документ из другого источника
Webhook	Способ отправки уведомлений при воспроизведении определенных событий
Покупатель	Физическое лицо, осуществляющее оплату товаров или услуг
Мерчант	Продавец или компания, которая продает товары или услуги и принимает онлайн-платежи от Покупателей
Банк-эквайер	Банк, осуществляющий первичную обработку транзакций по банковским картам, в пользу Мерчанта — клиента этого банка
Банк-эмитент	Банк, выпустивший и обслуживающий карту плательщика (физическое лицо), участвует в проведении операций
Виртуальный кошелек	Сущность в DucatPay, в рамках которой ведется учет баланса ВДС
Возврат платежа	Возврат денежных средств Покупателю
Двухстадийный платеж	Платеж, который включает выполнение двух операций: блокирование средств на карте и завершение авторизации — списание
Инвойс	Документ, который Мерчант выставляет Покупателю в качества требования платежа. Содержит информацию о деталях сделки
Магазин	Торговая точка Мерчанта в DucatPay, необходимая для работы с платежными операциями
Одностадийный платеж	Платеж, состоящий из единственной операции — прямого списания средств с карты Покупателя
Платежное поручение	Документ, на основании которого банк по поручению Мерчанта переводит деньги на счет Покупателя
Платежная форма	Форма для ввода банковских реквизитов и проведения онлайн-оплаты
Получатель выплаты	Сущность в DucatPay, которая хранит данные о получателе и назначении платежа при выводе средств с баланса виртуального кошелька
Привязка карты	Процесс оплаты инвойса с использованием двухстадийного платежа
Публичный ключ	Криптографический код, используемый для подтверждения при получении уведомлений
Рекуррентный платеж	Платеж, который происходит регулярно или автоматически по инициативе Мерчанта, без необходимости повторного ввода платежных данных и прохождения 3DS проверки со стороны Покупателя
Родительский платеж	Первый оплаченный инвойс Покупателем с разрешением запомнить реквизиты банковской карты и согласием на условия проведения рекуррентных платежей
Транзакция	Операция, связанная с движением денежных средств на банковском счете. Каждая транзакция имеет свой RRN
Цифровая личность	Сущность в DucatPay, соответствующая ID организации для проведения выплат

1 Введение

Руководство предназначено для пользователей программного обеспечения «DucatPay» (далее — DucatPay, сервис). Документ содержит описание ключевых бизнес-процессов, порядок технической интеграции, описание процесса эксплуатации и спецификацию API.

Техническая поддержка DucatPay осуществляется по электронной почте support@ducatpay.com с 05:00 до 22:00 по московскому времени, за исключением выходных и праздничных дней. Время ответа на обращение — сутки.

1.1 О сервисе

DucatPay — процессинговый сервис, предоставляющий услуги интернет-эквайринга для электронной коммерции и онлайн-торговли.

Сервис обеспечивает выполнение следующих функций:

- Сопровождение движения денежных средств.
- Формирование счетов и ссылок на оплату.
- Полный и частичный возврат платежей.
- Выплаты по реквизитам банковских карт.
- Создание магазинов для работы с платежами.
- Просмотр аналитики и создание отчетов по магазинам.

Сервис включает в себя:

1. Личный кабинет (ЛК) — веб-приложение для мониторинга, аналитики и создания отчетов по операциям.
2. API — техническая реализация взаимодействия с сервисом, представленная двумя группами методов «Платежи» и «Выплаты».

Доступ к веб-приложению осуществляется с помощью браузера, запускаемого на персональном устройстве пользователя. Интеграция с API реализуется средствами программного обеспечения пользователя.

Сервис имеет сертификат и соответствует требованиям стандарта безопасности PCI DSS level 1.

1.2 Системные требования

ЛК DucatPay доступен для работы на всех видах устройств, поддерживающих работу браузера, такими как:

- персональный компьютер (ПК);
- ноутбук;
- планшет;
- мобильный телефон.

Для корректной работы в ЛК устройство пользователя должно обладать минимальными техническими характеристиками, приведенными в таблице 1.

Таблица 1 — Минимальные технические характеристики

Характеристики	Требование
Оперативная память	Не менее 0,5 Гб
Свободное место на диске	Не менее 100 Мб
Тип интернет-соединения	Ethernet, Wi-Fi
Скорость интернет-соединения	Не менее 128 Кбит/с

Требования к операционным системам и браузерам, установленным на устройстве пользователя, приведены в таблице 2.

Таблица 2 — Требования к операционным системам и браузерам

Операционная система и версия	Браузеры
Windows: 8/8.1, 10, 11	<ul style="list-style-type: none">• Microsoft Edge версии 122 и выше.• Яндекс Браузер версии 20.6 и выше.• Google Chrome версии 109 и выше.• Firefox версии 102 и выше.• Opera версии 95 и выше
Linux: <ul style="list-style-type: none">• Ubuntu 18.04 и выше.• Debian 10 и выше.• Fedora 32 и выше	<ul style="list-style-type: none">• Google Chrome версии 115 и выше.• Яндекс Браузер версии 20.6 и выше.• Firefox версии 110 и выше.• Opera версии 75 и выше
Android: 8.0 и выше	
macOS: Big Sur 11.7.10 и выше	<ul style="list-style-type: none">• Safari версии 14 и выше.• Firefox версии 110 и выше.• Opera версии 75 и выше.• Яндекс Браузер версии 20.6 и выше.• Google Chrome версии 115 и выше
iOS и iPadOS: 15.8.3 и выше	

Взаимодействие с API DucatPay должно осуществляться по защищенному протоколу (HTTPS). HTTP-запросы по не зашифрованному каналу не поддерживаются.

1.3 Уровень подготовки пользователя

Для работы с **личным кабинетом DucatPay** пользователь должен иметь навык работы с любым из поддерживаемых интернет-браузеров, а также обладать базовыми навыками работы с персональным компьютером и мобильным телефоном. Знание соответствующей предметной области желательно, но не обязательно.

Взаимодействуя с **API DucatPay**, пользователь должен иметь знания и навыки работы с клиент-серверной и REST архитектурой, основными типами запросов, обработкой распространенных кодов ошибок. К настройке взаимодействия рекомендуем привлечь технического специалиста вашей компании.

2 Подготовка к работе

Чтобы начать работу с DucatPay, нужно пройти несколько обязательных шагов:

1. [Зарегистрировать организацию в DucatPay.](#)
2. [Получить логин и пароль от ЛК.](#)
3. [Авторизоваться в ЛК.](#)
4. [Создать магазин.](#)
5. [Получить API-ключ и ID организации.](#)

2.1 Регистрация организации в DucatPay

Регистрация организации в сервисе происходит через обращение в [техническую поддержку](#). В заявке укажите информацию из таблицы 3.

Таблица 3 — Информация, необходимая для регистрации организации

Вид	Состав
Информация о юридическом лице компании	<ul style="list-style-type: none">• Наименование юридического лица.• Юридический адрес.• ИНН.• ОГРН/ОГРНИП
Информация о банковском терминале	<ul style="list-style-type: none">• Идентификатор банковского терминала.• Реквизиты доступа к терминалу — логин и ключ авторизации
Контактная информация	<ul style="list-style-type: none">• Должность ЕОИ/представителя.• Ф.И.О ЕОИ/представителя.• Документ ЕОИ/представителя.• E-mail — для получения отчетов

Техническая поддержка может обратиться за уточнением информации по заявке в случаях, когда указанной информации недостаточно.

Срок регистрации занимает 1-2 рабочих дня. Уведомление об успешной регистрации придет в ответном письме.

2.2 Получение логина и пароля

Ссылка на ЛК, логин и пароль для входа в него придут в письме по электронной почте после регистрации организации технической поддержкой. Организации предоставляется учетная запись для 1 пользователя.

2.3 Авторизация в сервисе

Перейдите в ЛК DucatPay по ссылке из письма. Введите логин (E-mail) и пароль от учетной записи пользователя, полученные при регистрации организации, в соответствующие поля и нажмите кнопку «Вход» (рис. 1).

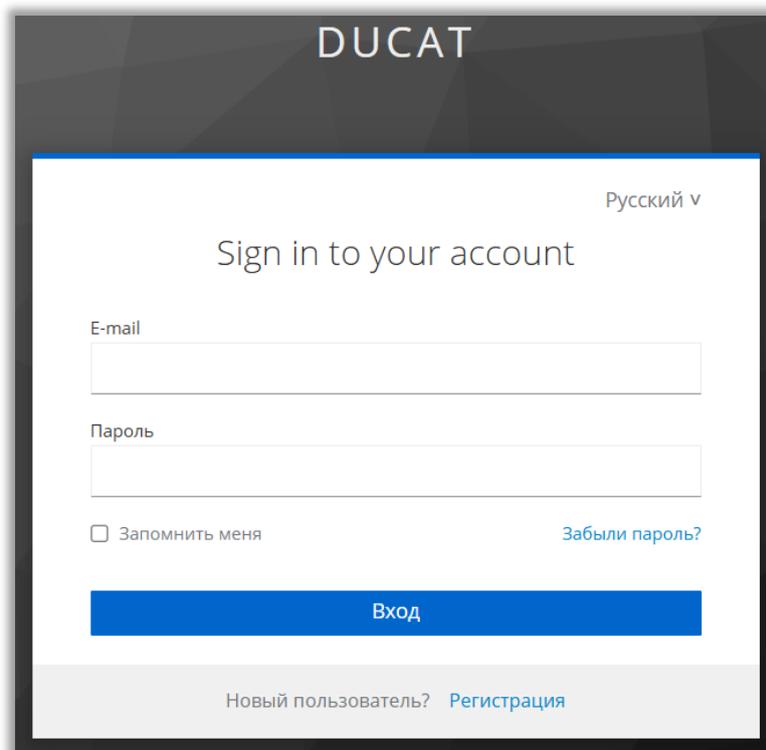
The image shows a login form for DucatPay. At the top, the word "DUCAT" is displayed in white on a dark background. Below it, the text "Русский v" is visible in the top right corner. The main heading is "Sign in to your account". There are two input fields: "E-mail" and "Пароль". Below the "E-mail" field is a checkbox labeled "Запомнить меня". To the right of the "Пароль" field is a link "Забыли пароль?". A large blue button labeled "Вход" is positioned below the input fields. At the bottom of the form, there is a link "Новый пользователь? Регистрация".

Рисунок 1 — Форма авторизации

⚠ Если данные для входа были забыты, потеряны или скомпрометированы, обратитесь в [техническую поддержку DucatPay](#).

2.4 Создание магазина

Для работы с платежными операциями в DucatPay нужна торговая точка — магазин. Для одной организации можно создать несколько магазинов в сервисе, которые могут иметь разные условия для проведения платежей.

Чтобы создать магазин, обратитесь в [техническую поддержку](#) и предоставьте следующую информацию:

- Название магазина.
- URL-адрес сайта, где размещена платежная форма.
- Наименование банка, с которым работаете.
- Расчетный счет.
- БИК.
- Корреспондентский счет.

На основании предоставленных данных техническая поддержка создаст магазин, настроит конфигурацию системы и проведение операций через банковские терминалы. Срок создания магазина занимает 1-2 рабочих дня.

Созданный магазин появится в ЛК в [разделе «Магазины»](#). Чтобы посмотреть информацию о магазине, нажмите на его название — откроется карточка магазина.

Для идентификации магазина при отправке запросов через API нужен уникальный **ID магазина**, который указывается в теле запроса в параметре `shopID`. ID магазина можно скопировать в [карточке магазина](#) в блоке «Идентификатор магазина» (рис. 2).

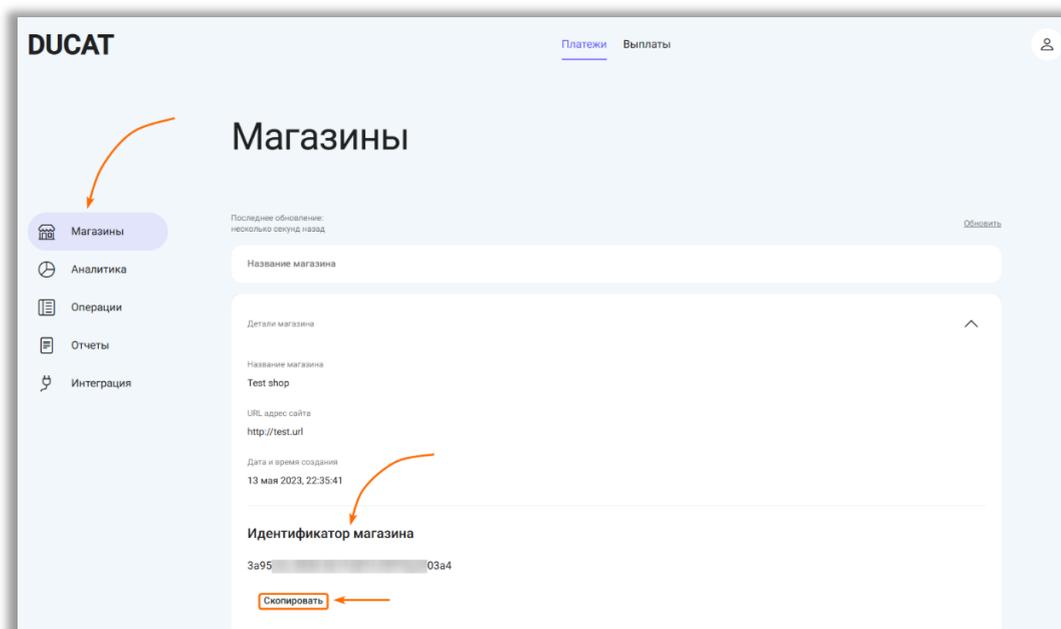


Рисунок 2 — Получение ID магазина из карточки

2.5 Получение API-ключа и ID организации

Для успешной аутентификации запроса к API необходимо в заголовке запроса передавать API-ключ организации.

API-ключ хранится в ЛК DucatPay в разделе «Интеграция» на вкладке «API-ключ» и доступен для копирования по кнопке «Скопировать» (рис. 3). API-ключ передается в заголовке Authorization.

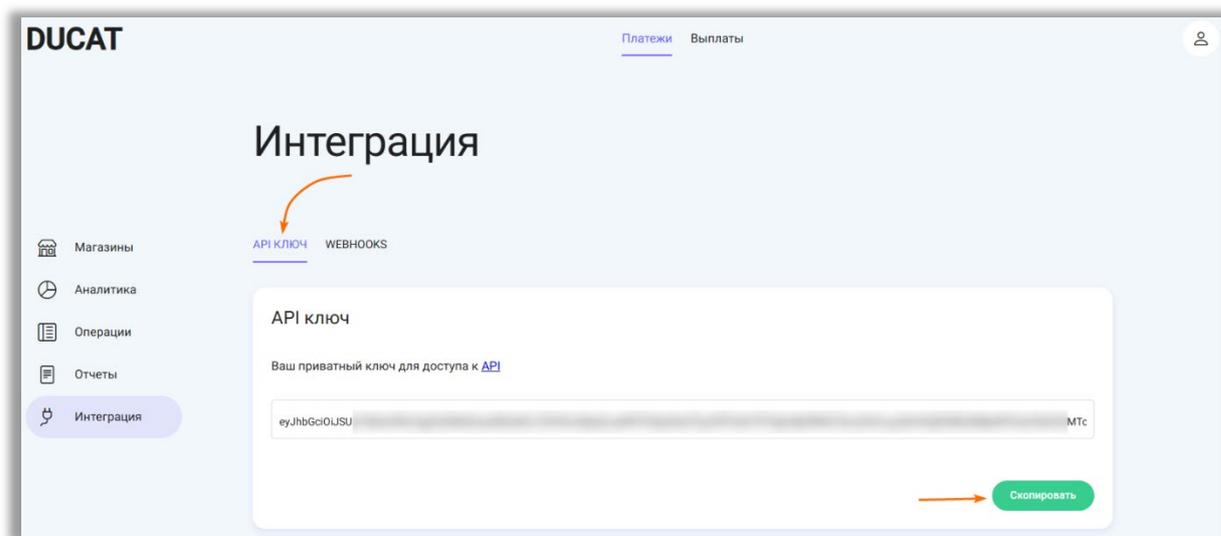


Рисунок 3 — Раздел «Интеграция» вкладка «API-ключ»

⚠ Не передавайте API-ключ третьим лицам!

Если API-ключ был скомпрометирован, незамедлительно обратитесь в [техническую поддержку DucatPay](#).

Чтобы идентифицировать организацию в сервисе, нужен уникальный **ID организации**, который указывается в теле запроса в параметре partyID. ID организации приходит после регистрации организации в сервисе вместе с учетными данными для входа.

3 Платежи

Чтобы денежные средства за предложенные товары или услуги поступили на банковский расчетный счет Мерчанта, Покупателю необходимо оплатить счет (инвойс). Оплата происходит через списание денежных средств с банковского счета Покупателя. Здесь и далее по тексту процесс списания денежных средств называется **платежом**.

Прием платежей осуществляется с банковских карт. Данные платежных карт Покупателей могут, при необходимости, храниться в сервисе в зашифрованном виде.

Проверка реквизитов и актуальности банковских карт происходит на стороне банка-эмитента.

Все платежи, проводимые через DucatPay, требуют использования 3D-Secure (3DS) аутентификации. 3DS проверка осуществляется на усмотрение банка-эмитента.

Платеж может состоять из одной или нескольких последовательных операций. В случае одностадийного проведения платежа денежные средства списываются с банковской карты Покупателя сразу. При двухстадийном платеже денежные средства удерживаются (холдируются) банком-эмитентом на определенный срок до момента подтверждения покупки Мерчантом или возврата платежа Покупателю.

Создание инвойсов и платежей происходит через API DucatPay. Осуществлять мониторинг статусов инвойсов и платежей, а также операций с ними можно через ЛК или API DucatPay.

Проведение онлайн-платежей доступно через платежную форму DucatPay или собственную платежную форму Мерчанта.

3.1 Платежная форма DucatPay

При использовании платежной формы DucatPay данные банковской карты вводятся Покупателем на стороне сервиса. Мерчант избавлен от необходимости обрабатывать платежные данные на своей стороне и соответствовать требованиям международного стандарта безопасности PCI DSS level 1.

Покупатель находится на сайте или в приложении Мерчанта только до момента ввода данных своей банковской карты. Например, до нажатия кнопки «Оплатить». Далее происходит открытие платежной формы DucatPay в HTML-странице в элементе iFrame (рис. 4).

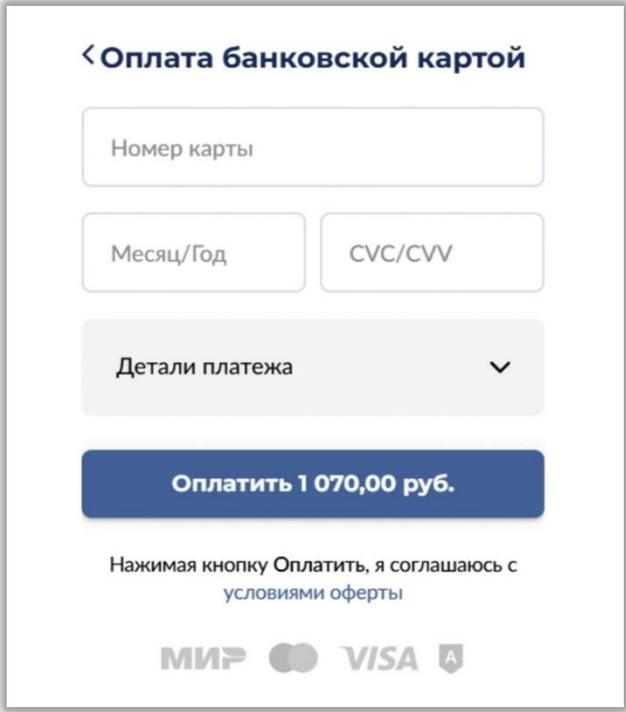


Рисунок 4 — Платежная форма DucatPay

Внешний вид платежной формы DucatPay можно изменить, добавить логотип организации или выбрать цвет кнопки оплаты. Чтобы изменить дизайн платежной формы, обратитесь в [техническую поддержку DucatPay](#).

Встроить готовую платежную форму DucatPay в ваше программное обеспечение (ПО) можно любым удобным для вас способом. Для этого ознакомьтесь с пунктами [Порядок взаимодействия с платежной формой DucatPay](#) и [Атрибуты для формирования платежной формы](#).

3.1.1 Порядок взаимодействия с платежной формой DucatPay

1. Покупатель формирует заказ на сайте или в приложении Мерчанта и переходит к шагу оплаты.
2. ПО Мерчанта запускает процесс создания инвойса вызовом API-метода [createInvoice](#). В запросе указываются обязательные параметры из таблицы 4.

Таблица 4 — Обязательные параметры API-метода createInvoice

Параметр и значение	Описание
api_key	API-ключ организации
X-Request-ID	ID запроса к сервису (слепок времени)
partyID	ID организации
shopID	ID магазина
dueDate	Дата и время окончания действия инвойса
amount	Сумма заказа
currency: RUB	Валюта
product	Наименование предлагаемых товаров и услуг
cart	Данные корзины товаров
metadata	Метаданные, которые необходимо связать с инвойсом

Подробное описание параметров запроса приведено в [спецификации](#).

3. В ответ на запрос вернется json-объект с ключевыми параметрами:
 - id — идентификатор инвойса (далее — invoiceID).
 - invoiceAccessToken — ключ доступа к инвойсу, авторизующий ограниченное количество операций для проведения платежей по указанному инвойсу.

Созданный инвойс имеет статус **Не оплачен** (unpaid) и доступен к просмотру в ЛК (рис. 5) в разделе «Операции» на [вкладке «Инвойсы»](#).

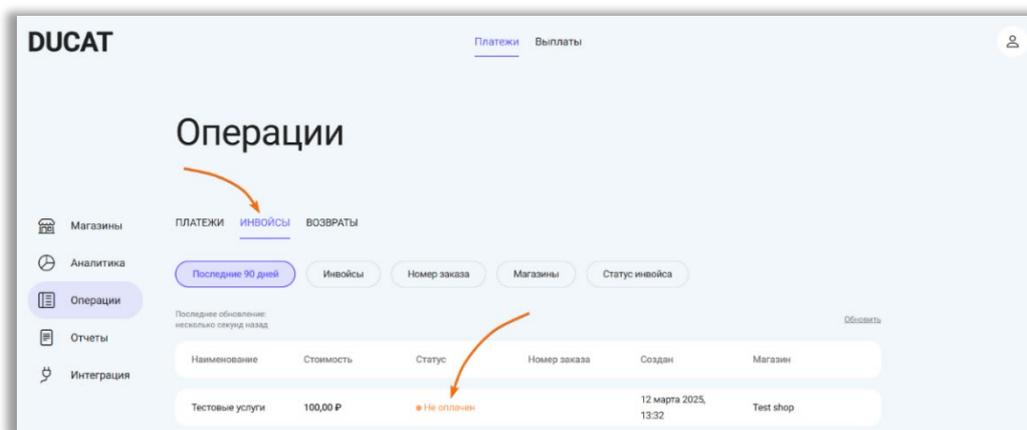


Рисунок 5 — Созданный инвойс в статусе Не оплачен

4. ПО Мерчанта генерирует ссылку на платежную форму DucatPay на домене `checkout-test.dev.ducat.pro`.

Ссылка дополняется обязательными параметрами:

- `invoiceID`;
- `invoiceAccessToken`;
- `email` — адрес электронной почты Покупателя;
- `locale` — локализация (язык) платежной формы;
- `initialPaymentMethod` — способ оплаты;
- другие параметры ссылки.

Пример

```
https://checkout-test.dev.ducat.pro/v1/checkout.html?invoiceID=9251vbRGkXEE
&invoiceAccessToken=eyJhbGciOiJIUzI1NiIsImtp3OEwzT3Vqa2ZtRWdIYUEiL
... 4pUkQezrTWE4nToVUYTITHU&email=example%40example.com&locale=ru
&initialPaymentMethod=bankCard
```

Чтобы провести **двухстадийный платеж**, дополнительно передайте параметр `paymentFlowHold=true` и укажите политику действий с удержанными средствами:

- `holdExpiration=cancel` — вернутся Покупателю;
- `holdExpiration=capture` — поступят Мерчанту.

Пример

```
https://checkout-test.dev.ducat.pro/v1/checkout.html?invoiceID=9251vbRGkXEE
&invoiceAccessToken=eyJhbGciOiJIUzI1NiIsImtp3OEwzT3Vqa2ZtRWdIYUEiL
... 4pUkQezrTWE4nToVUYTITHU&email=example%40example.com&locale=ru&
initialPaymentMethod=bankcard&paymentFlowHold=true&holdExpiration=cancel
```

Если по инвойсу в будущем требуется провести рекуррентные платежи, дополнительно в ссылке передайте параметр `recurring=true`.

5. ПО Мерчанта предоставляет сформированную ссылку на платежную форму Покупателю любым удобным способом. Например, через кнопку «Оплатить» на сайте или в приложении Мерчанта.

6. Покупатель переходит на платежную форму и выбирает способ оплаты «Банковская карта». Заполняет реквизиты банковской карты (рис. 6) и нажимает кнопку «Оплатить».

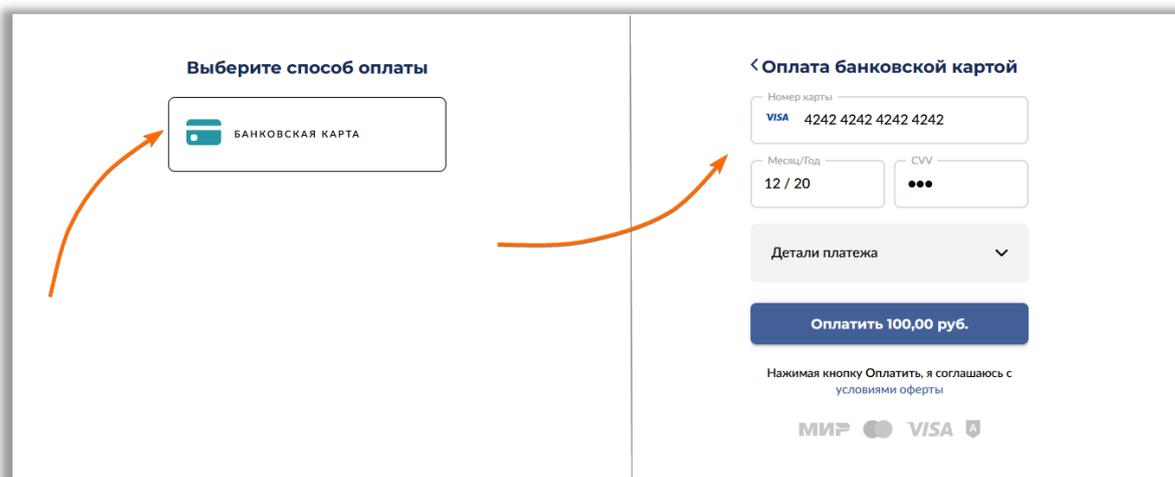


Рисунок 6 — Процесс выбора способа оплаты и ввода платежных реквизитов

7. Платежная форма DucatPay формирует платеж и направляет в банк-эквайер данные банковской карты и переданные параметры.
8. На усмотрение банка-эмитента Покупатель подтверждает покупку 3DS-кодом.
9. После подтверждения оплаты (рис. 7) Покупатель возвращается на сайт или в приложение Мерчанта.

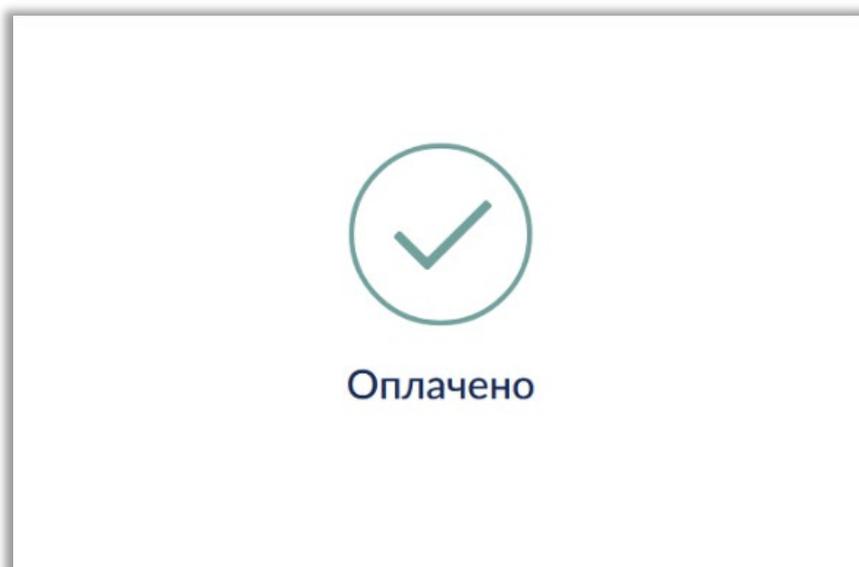


Рисунок 7 — Подтверждение оплаты от DucatPay

10. Одностадийный платеж в случае успеха получает статус **Подтвержден** (processed), а статус инвойса **Оплачен** (paid).

В зависимости от выбранного параметра для двухстадийного платежа:

- Параметр `cancel` — денежные средства через 1 минуту после подтверждения оплаты вернутся Покупателю. Статус платежа меняется с **Обработан** (`processed`) на **Отменен** (`cancelled`). Статус инвойса меняется на **Отменен** (`cancelled`).
- Параметр `capture` — денежные средства будут удержаны со счета Покупателя до момента подтверждения заказа Мерчантом. Статус платежа меняется с **Обработан** (`processed`) на **Подтвержден** (`captured`). Статус инвойса меняется с **Не оплачен** (`unpaid`) на **Оплачен** (`paid`).

Проверить статус платежа (рис. 8) и инвойса (рис. 9) можно в ЛК в [разделе «Операции»](#) или настроив [Webhook-уведомление по событиям платежей](#).

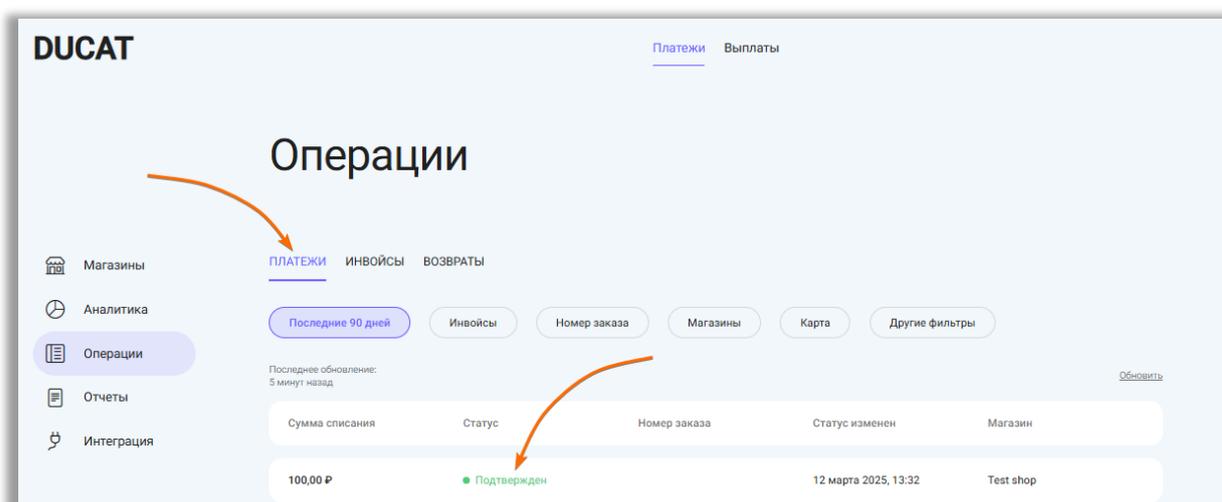


Рисунок 9 — Проверка статуса платежа в ЛК

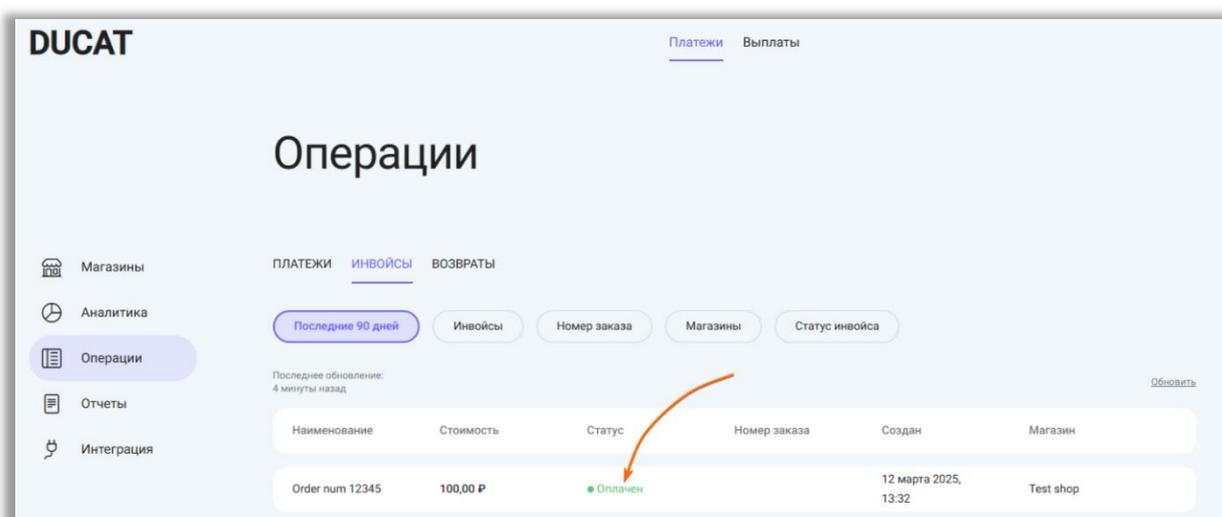


Рисунок 8 — Проверка статуса инвойса в ЛК

3.1.2 Атрибуты для формирования платежной формы

В таблице 5 приведены атрибуты для управления поведением платежной формы DucatPay.

Таблица 5 — Атрибуты управления платежной формы DucatPay

HTML (data-*)	Параметр ссылки/Свойство объекта JS	Описание	Обязательный	Возможные значения
data-invoice-id	invoiceID	Идентификатор инвойса	+	oVU2LzUCb
data-invoice-access-token	invoiceAccessToken	Ключ доступа к инвойсу	+	eJhGiJSUzI...
data-initial-payment-method	initialPaymentMethod	Способ оплаты	+	bankCard
data-email	email	Email Покупателя, предзаполнен на форме	—	mail@mail.ru
data-locale	locale	Настройка локализации платёжной формы	—	ru/en
data-recurring	recurring	Признак рекуррентного платежа	—	true/false
data-payment-flow-hold	paymentFlowHold	Признак двухстадийного платежа	—	true/false
data-hold-expiration	holdExpiration	Действие с удержанными средствами	—	cancel/capture

Пример вызова платежной формы DucatPay по кнопке «Оплатить» с указанными атрибутами:

```
<form action="https://your-web-site.address/success.action.page"
method="GET">
  <script src="https://checkout.ducat.pro/checkout.js" class="ps-checkout"
data-invoice-id="9250vbRGkXEE"
data-invoice-access-token="eyJhbGciOiJI... 4pUkQezrTWE4niTHU"
data-initial-payment-method="bankCard"
data-payment-flow-hold="true"
data-hold-expiration="capture"
data-email='example@mail.com'
data-locale="ru"
data-pay-button-label="Оплатить">
  </script>
</form>
```

 success.action.page — адрес страницы переадресации Покупателя на сайт или в приложение Мерчанта в случае успешного платежа.

3.2 Платежная форма Мерчанта

Мерчант может использовать собственную платежную форму для проведения онлайн-платежей. В этом случае Покупатель не покидает сайт или приложение Мерчанта во время ввода реквизитов банковской карты или совершения платежных операций. ПО Мерчанта передает полученные платежные данные через API-запросы в DucatPay для проведения платежей.

 Организации, которые оперируют данными платежных карт или могут повлиять на безопасность платежных данных, должны иметь сертификат и соответствовать требованиям международного стандарта безопасности PCI DSS level 1.

Мерчант самостоятельно подтверждает и обеспечивает соответствие собственной платежной формы и ПО стандарту PCI DSS.

3.2.1 Порядок взаимодействия с платежной формой Мерчанта

1. Покупатель формирует заказ на сайте или в приложении Мерчанта и переходит к шагу оплаты.
2. ПО Мерчанта создает инвойс вызовом API-метода [createInvoice](#). Процесс аналогичен шагам 2 и 3 пункта [Порядок взаимодействия с платежной формой DucatPay](#). В ответ на запрос вернется json-объект с ключевыми параметрами: `invoiceID` и `invoiceAccessToken`.
3. Покупатель на сайте или в приложении Мерчанта переходит на форму оплаты, вводит реквизиты банковской карты и нажимает кнопку «Оплатить».
4. ПО Мерчанта вызывает API-метод [createPaymentResource](#) для создания одноразового токена платежного средства и уникальной платежной сессии. В запросе укажите обязательные параметры из таблицы 6.

Таблица 6 — Обязательные параметры API-метода `createPaymentResource`

Параметр и значение	Описание
<code>api_key</code>	API-ключ организации
<code>X-Request-ID</code>	ID запроса к сервису (слепок времени)
<code>paymentTool</code>	Данные платежного средства
<code>paymentToolType: CardData</code>	Тип платежного средства — банковская карта
<code>cardNumber</code>	Номер банковской карты
<code>expDate</code>	Срок действия банковской карты
<code>cvv</code>	Код верификации
<code>clientInfo</code>	Данные устройства Покупателя
<code>fingerprint</code>	ID платежного средства

Подробное описание параметров запроса приведено в [спецификации](#).

5. В ответ на запрос вернется json-объект с ключевыми параметрами:

- `paymentToolToken` — токен платежного средства, в котором зашифрованы реквизиты банковской карты.
- `paymentSession` — идентификатор платежной сессии.

 Токен платежного средства и идентификатор сессии имеют ограниченный срок жизни — 1 час.

6. ПО Мерчанта вызывает API-метод [createPayment](#) для создания платежа по указанному инвойсу. В запросе укажите обязательные параметры из таблицы 7.

Таблица 7 — Обязательные параметры API-метода `createPayment`

Параметр и значение	Описание
<code>api_key</code>	API-ключ организации
<code>X-Request-ID</code>	ID запроса к сервису (слепок времени)
<code>externalID</code>	Внешний ID запроса для внутреннего операционного учета Мерчанта
<code>flow</code>	Тип платежа и порядок удержания денежных средств
<code>type</code>	Тип платежа: <ul style="list-style-type: none"> • <code>PaymentFlowInstant</code> — одностадийный; • <code>PaymentFlowHold</code> — двухстадийный
<code>onHoldExpiration</code>	Действие с удержанными средствами для двухстадийного платежа: <ul style="list-style-type: none"> • <code>cancel</code> — вернуться Покупателю; • <code>capture</code> — поступят Мерчанту
<code>payer</code>	Информация о платежных данных
<code>payerType: PaymentResourcePayer</code>	Тип платежного средства: <ul style="list-style-type: none"> • <code>PaymentResourcePayer</code> — платеж по банковской карте; • <code>RecurrentPayer</code> — рекуррентный платеж
<code>paymentToolToken</code>	Токен платежного средства, в котором зашифрованы реквизиты банковской карты
<code>paymentSession</code>	ID платежной сессии
<code>contactInfo</code>	Контактные данные Покупателя (<code>email</code> и <code>phoneNumber</code>)
<code>makeRecurrent: false</code>	Признак создания родительского рекуррентного платежа: <ul style="list-style-type: none"> • <code>false</code> — признак выключен; • <code>true</code> — признак включен

Подробное описание параметров запроса приведено в [спецификации](#).

В ответ в поле `id` json-объекта возвращается идентификатор платежа (далее — `paymentID`), необходимый для проверки статуса платежа.

7. API DucatPay получает запрос на обработку платежа и отправляет полученные данные в банк-эквайер.
8. DucatPay получает ссылку на прохождение авторизации 3DS от банка-эмитента и передает ее в ПО Мерчанта, которое организует перенаправление Покупателя на страницу авторизации 3DS.
9. После получения успешного статуса от банка-эмитента DucatPay меняет статус одностадийного платежа на **Обработан** (`processed`), а статус инвойса — на **Оплачен** (`paid`).

Для двухстадийного платежа в зависимости от выбранного параметра:

- Параметр `cancel` — денежные средства через 1 минуту после подтверждения оплаты вернутся Покупателю. Статус платежа меняется с **Обработан** (`processed`) на **Отменен** (`cancelled`). Статус инвойса меняется на **Отменен** (`cancelled`).
- Параметр `capture` — денежные средства будут удержаны со счета Покупателя до момента подтверждения заказа Мерчантом. Статус платежа меняется с **Обработан** (`processed`) на **Подтвержден** (`captured`). Статус инвойса меняется с **Не оплачен** (`unpaid`) на **Оплачен** (`paid`).

10. Чтобы проверить статус платежа, ПО Мерчанта может настроить [Webhook-уведомление](#) или вызывать API-метод [getPaymentByID](#). В запросе укажите обязательные параметры из таблицы 8.

Таблица 8 — Обязательные параметры API-метода `getPaymentByID`

Параметр и значение	Описание
<code>api_key</code>	API-ключ организации
<code>X-Request-ID</code>	ID запроса к сервису (слепок времени)
<code>invoiceID</code>	ID инвойса
<code>paymentID</code>	ID платежа

В ответ возвращается json-объект с информацией по платежу и его статусом.

3.3 Возврат платежей

В сервисе доступен полный или частичный возврат платежа. Инициировать возврат можно двумя способами:

- Через ЛК DucatPay.
- Через запрос к API.

⚠ Сумма возврата не должна превышать сумму платежа.

3.3.1 Возврат платежа в ЛК DucatPay

1. Откройте [раздел «Операции»](#) и перейдите на [вкладку «Платежи»](#).
2. Найдите нужный платеж в таблице и раскройте информацию по нему.
3. В блоке «Возвраты» [карточки платежа](#) (рис. 10) нажмите кнопку «Совершить возврат».

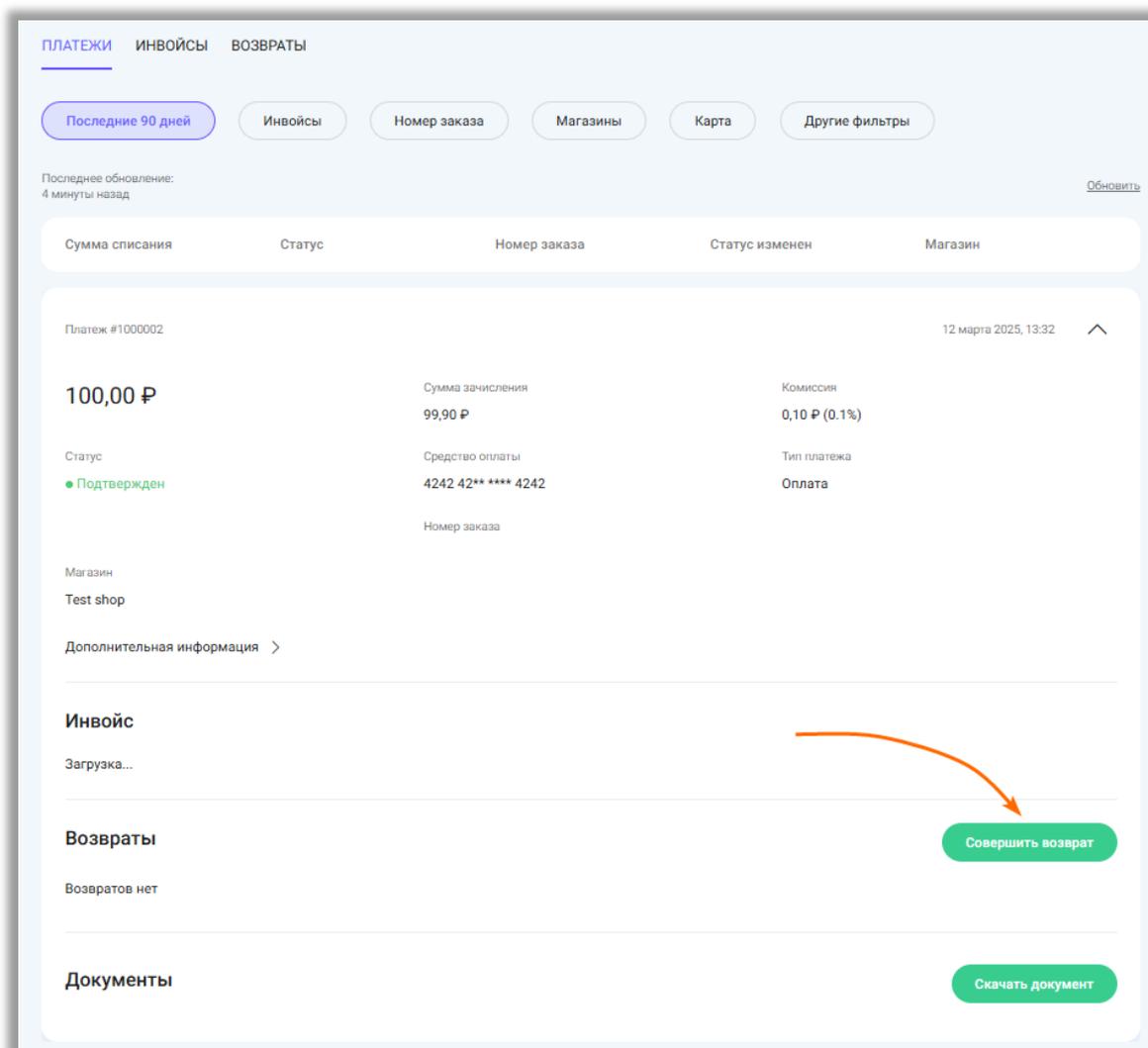


Рисунок 10 — Кнопка «Совершить возврат» в карточке платежа

4. В открывшейся форме укажите причину возврата.
5. Чтобы вернуть **всю сумму платежа**, нажмите кнопку «Подтвердить». Для проведения частичного возврата, поставьте галочку в пункте «Частичный возврат» и укажите сумму к возврату (рис. 11). Нажмите кнопку «Подтвердить».

Совершить возврат

Причина возврата
Возврат тестового платежа
25 / 100

Частичный возврат

Сумма возврата *
60
Доступно к возврату: 100,00 Р

Подтвердить

Рисунок 11 — Форма для частичного возврата платежа

Информация о возвращенных платежах (рис. 12) хранится в разделе «Операции» на [вкладке «Возвраты»](#).

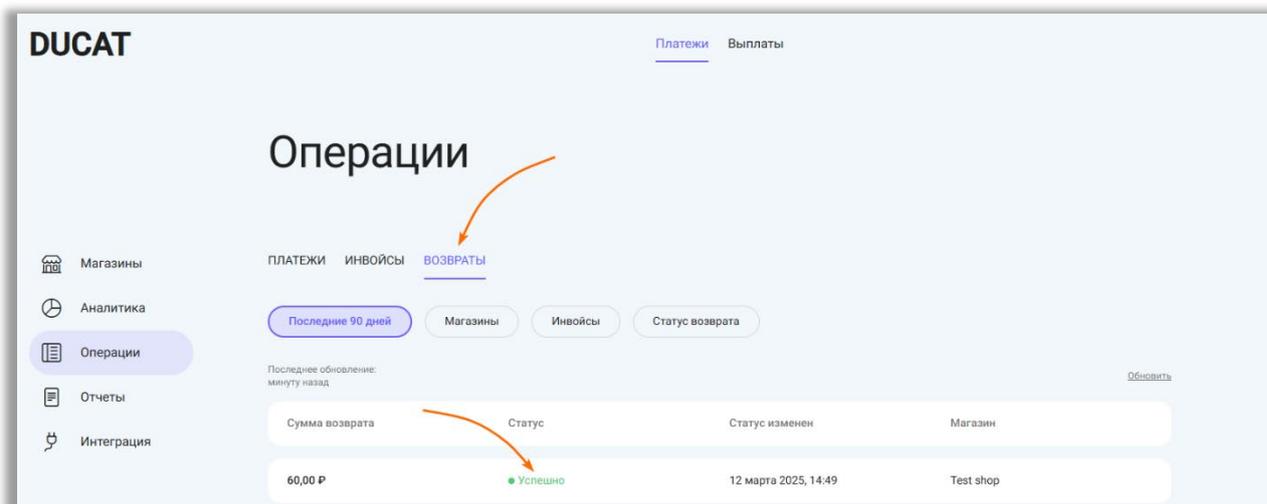


Рисунок 12 — Информация о возвратах в ЛК

3.3.2 Возврат платежа через API

Чтобы вернуть платеж, используйте API-метод [createRefund](#). В запросе укажите обязательные параметры из таблицы 9.

Таблица 9 — Обязательные параметры API-метода createRefund

Параметр и значение	Описание
api_key	API-ключ организации
X-Request-ID	ID запроса к сервису (слепок времени)
invoiceID	ID инвойса
paymentID	ID платежа по инвойсу
amount	Сумма возврата (полная или частичная)
currency: RUB	Валюта платежа
reason	Причина возврата

Подробное описание параметров запроса приведено в [спецификации](#).

✦ Параметры invoiceID и paymentID доступны для просмотра в ЛК DucatPay.

В ответ в поле id json-объекта возвращается идентификатор возврата (refundID). Статус платежа меняется с **Подтвержден** (captured) на **Возвращен** (refunded).

3.4 Рекуррентные платежи

Рекуррентный платеж — это платеж, который происходит регулярно или автоматически по инициативе Мерчанта, без необходимости повторного ввода платежных данных и прохождения 3DS проверки со стороны Покупателя. Такая механика подойдет организациям, которые предлагают подписку на свои товары или услуги.

Для создания и проведения рекуррентного платежа необходим **родительский платеж**.

Родительский платеж — первая оплата инвойса Покупателем с разрешением запомнить реквизиты банковской карты и согласием на условия проведения рекуррентных платежей.

Для создания родительского платежа, укажите параметры из таблицы 10.

Таблица 10 — Создание родительского платежа для разных платежных форм

Платежная форма DucatPay	Собственная платежная форма Мерчанта
При формировании ссылки на платежную форму DucatPay обязательно передайте параметр <code>recurring=true</code> .	В момент создания платежа API-методом <code>createPayment</code> , передайте параметр <code>makeRecurrent: true</code> .

3.4.1 Порядок проведения рекуррентного платежа

1. Покупатель формирует заказ на сайте или в приложении Мерчанта и переходит к шагу оплаты.
2. Покупатель разрешает запомнить реквизиты банковской карты и соглашается с условиями рекуррентных платежей. Нажимает кнопку «Оплатить».
3. Создается и проводится платеж, который будет считаться родительским.
4. При наступлении даты списания рекуррентного платежа, ПО Мерчанта создает новый инвойс через API-метод [createInvoice](#).
5. Для создания рекуррентного платежа ПО Мерчанта вызывает API-метод [createPayment](#). В запросе указываются обязательные параметры из таблицы 11.

Таблица 11 — Обязательные параметры API-метода createPayment при рекуррентном платеже

Параметр и значение		Описание
api_key		API-ключ организации
X-Request-ID		ID запроса к сервису (слепок времени)
invoiceID		ID инвойса
flow		Тип платежа и порядок удержания денежных средств
	type	Тип платежа: <ul style="list-style-type: none"> • PaymentFlowInstant — одностадийный; • PaymentFlowHold — двухстадийный
	onHoldExpiration	Действие с удержанными средствами для двухстадийного платежа: <ul style="list-style-type: none"> • cancel — вернуться покупателю; • capture — поступят Мерчанту
payer		Информация о платежных данных
	payerType: RecurrentPayer	Тип платежного средства
	contactInfo	Контактные данные Покупателя (email и phoneNumber)
	recurrentParentPayment	Параметры родительского платежа
	invoiceID	ID родительского инвойса
	paymentID	ID родительского платежа
	makeRecurrent: true	Признак создания родительского рекуррентного платежа: <ul style="list-style-type: none"> • false — признак выключен; • true — признак включен

Подробное описание параметров запроса приведено в [спецификации](#).

- С банковской карты Покупателя автоматически списываются денежные средства.

4 Выплаты

Через сервис DucatPay можно отправлять денежные средства физическим лицам по реквизитам банковских карт. Функциональность подойдет организациям, которые оплачивают услуги исполнителей или контрагентов, выплачивают маркетинговые вознаграждения и т.д. Здесь и далее по тексту процесс отправки денежных средств называется **выплатой**.

Управление выплатами происходит через создание виртуального кошелька в DucatPay. Кошелек необходим для ведения учета выплат в сервисе и не отражает реальный счет организации. Баланс кошелька пополняется только виртуальными денежными средствами (ВДС) через обращение в [техническую поддержку](#).

 DucatPay не работает с реальными денежными средствами организации. Сервис передает запрос о выплате в банк-эквайер, который производит выплату с расчетного счета организации по указанным реквизитам.

Получить информацию о виртуальных кошельках и операциях с их использованием можно через API или ЛК DucatPay. Управление ВДС происходит через API DucatPay.

 Возврат выплат недоступен через DucatPay.

4.1 Подготовительный этап

Пройдите обязательные шаги для **первичной настройки** сервиса:

1. [Создайте цифровую личность](#).
2. [Создайте виртуальный кошелек](#).
3. [Пополните баланс кошелька](#).

4.1.1 Создание цифровой личности

Цифровая личность (identity) — это ID организации для проведения выплат.

Одна организация может иметь несколько цифровых личностей для отдельного учета и разных сценариев использования. Например, юридическое лицо может владеть несколькими брендами, в таком случае каждый бренд будет являться отдельной цифровой личностью.

Чтобы создать цифровую личность, используйте API-метод [createIdentities](#). В запросе укажите обязательные параметры из таблицы 12.

Таблица 12 — Обязательные параметры API-метода createIdentities

Параметр и значение	Описание
api_key	API-ключ организации
X-Request-ID	ID запроса к сервису (слепок времени)
partyID	ID организации
name	Наименование юридического лица
provider	Название банка-эквайера

Подробное описание параметров запроса приведено в [спецификации](#).

В ответ в поле id json-объекта возвращается идентификатор цифровой личности (далее — identity).

4.1.2 Создание виртуального кошелька

Виртуальный кошелек (wallet) — сущность, в рамках которой ведётся учёт баланса ВДС. Одна цифровая личность может иметь несколько виртуальных кошельков.

Чтобы выпустить кошелек для цифровой личности, используйте API-метод [createWallet](#). В запросе укажите обязательные параметры из таблицы 13.

Таблица 13 — Обязательные параметры API-метода createWallet

Параметр и значение	Описание
api_key	API-ключ организации
X-Request-ID	ID запроса к сервису (слепок времени)
identity	ID организации для выплат
name	Название кошелька
currency: RUB	Валюта кошелька

Подробное описание параметров запроса приведено в [спецификации](#).

В ответ в поле id json-объекта вернется идентификатор кошелька (далее — wallet).

4.1.3 Пополнение баланса виртуального кошелька

Для проведения выплаты на балансе виртуального кошелька должны быть ВДС в сумме большей или равной сумме операции.

Чтобы пополнить баланс виртуального кошелька, обратитесь в [техническую поддержку DucatPay](#). В обращении укажите идентификатор кошелька и сумму пополнения в рублях.

Техническая поддержка на основании предоставленных данных создаст виртуальный депозит и начислит необходимую сумму на баланс виртуального кошелька.

 Баланс виртуального кошелька может не соответствовать реальной сумме денежных средств на счете организации.

4.2 Процесс проведения выплаты

Чтобы операция выплаты прошла успешно, необходимо выполнить условия:

- Баланс кошелька должен быть больше или равен сумме выплаты.
- Карта физического лица должна быть привязана в сервисе.

Привязка карты — процесс оплаты инвойса с использованием двухстадийного платежа.

Процесс оплаты инвойса описан в пунктах [Порядок взаимодействия с платежной формой DucatPay](#) и [Порядок взаимодействия с платежной формой Мерчанта](#).

 В DucatPay могут храниться реквизиты банковских карт в зашифрованном виде.

Порядок проведения выплаты

1. Получите токен платежного средства (значение параметра `paymentToolToken`) из инвойса. Подробнее в таблице 14.

Таблица 14 — Получение токена для разных платежных форм

Платежная форма DucatPay	Собственная платежная форма Мерчанта
<p>Вызовите API-метод getPaymentByID. В запросе передайте параметры:</p> <ul style="list-style-type: none">• <code>invoiceID</code> — ID инвойса;• <code>paymentID</code> — ID платежа. <p> Параметры доступны для просмотра в ЛК DucatPay.</p> <p>В ответ вернется json-объект, из которой нужно будет взять значение поля <code>paymentToolToken</code></p>	<p>Если в ПО Мерчанта настроено хранение токенов, возьмите его значение из параметра <code>paymentToolToken</code>.</p> <p>Иначе вызовите API-метод getPaymentByID, как в случае с платежной формой DucatPay</p>

2. Создайте получателя выплаты (destination) используя API-метод [createDestination](#). В запросе укажите обязательные параметры из таблицы 15.

Таблица 15 — Обязательные параметры API-метода createDestination

Параметр и значение	Описание
api_key	API-ключ организации
X-Request-ID	ID запроса к сервису (слепок времени)
name	Наименование получателя
identity	ID организации для выплат
currency: RUB	Валюта кошелька
resource	Платежные данные получателя
type: BankCardDestinationResource	Способ выплаты на банковскую карту
token	Токен платежного средства, с зашифрованными реквизитами банковской карты (значение параметра paymentToolToken)

Подробное описание параметров запроса приведено в [спецификации](#).

В ответ в поле id json-объекта возвращается идентификатор получателя (далее — destination).

3. Создайте выплату используя API-метод [createWithdrawal](#). В запросе укажите обязательные параметры из таблицы 16.

Таблица 16 — Обязательные параметры API-метода createWithdrawal

Параметр и значение	Описание
api_key	API-ключ организации
X-Request-ID	ID запроса к сервису (слепок времени)
wallet	ID кошелька
destination	ID приемника денежных средств
body	Параметры выплаты
amount	Сумма выплаты
currency: RUB	Валюта выплаты

Подробное описание параметров запроса приведено в [спецификации](#).

В ответ в поле id json-объекта возвращается идентификатор выплаты (далее — withdrawalID) и статус **В ожидании** (pending). Выплата направляется на обработку в банк-эквайер.

Когда банк-эквайер обработает выплату и переведет денежные средства с банковского счета организации на банковскую карту физического лица, статус выплаты в сервисе поменяется на **Успешно** (succeeded).

Узнать статус выплаты можно через API-метод [getWithdrawalStatus](#).

Информация по всем выплатам отображается в ЛК DucatPay (рис. 13) в меню «Выплаты» в [разделе «Операции»](#).

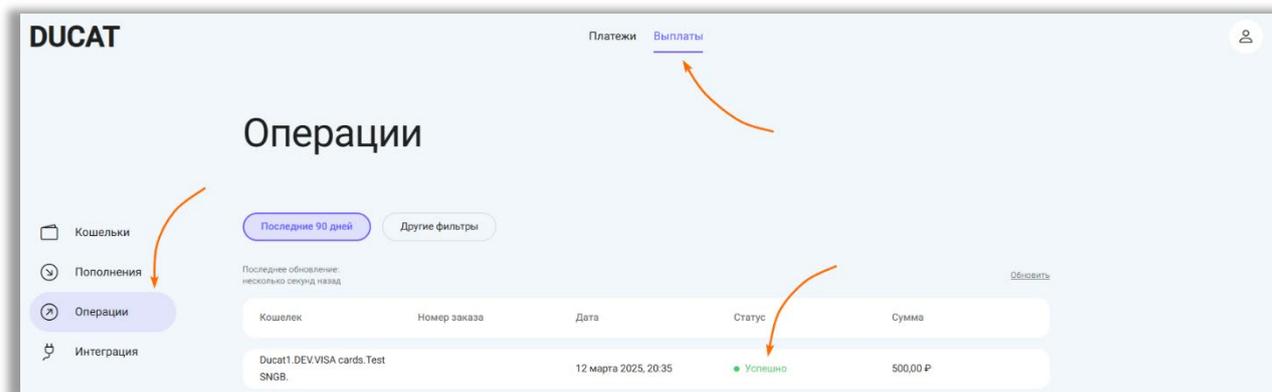


Рисунок 13 — Информация о выплатах в ЛК

5 Личный кабинет

Личный кабинет — это веб-приложение DucatPay, в котором можно:

- Просмотреть информацию по всем операциям, проведенным в сервисе.
- Получить API-ключ и ID магазина.
- Просмотреть сводную статистику.
- Создать отчет по операциям за указанный период.
- Настроить и создать Webhook-уведомления.
- Просмотреть информацию по виртуальным кошелькам и операциям с ними.

Вход в ЛК DucatPay осуществляется по логину и паролю, подробнее в пунктах [Получение логина и пароля](#) и [Авторизация в сервисе](#).

ЛК разделен по процессам [Платежей](#) и [Выплат](#). Используйте меню для переключения между разделами.

После входа в ЛК (рис. 14) по умолчанию открывается меню «Платежи» раздел «Магазины».

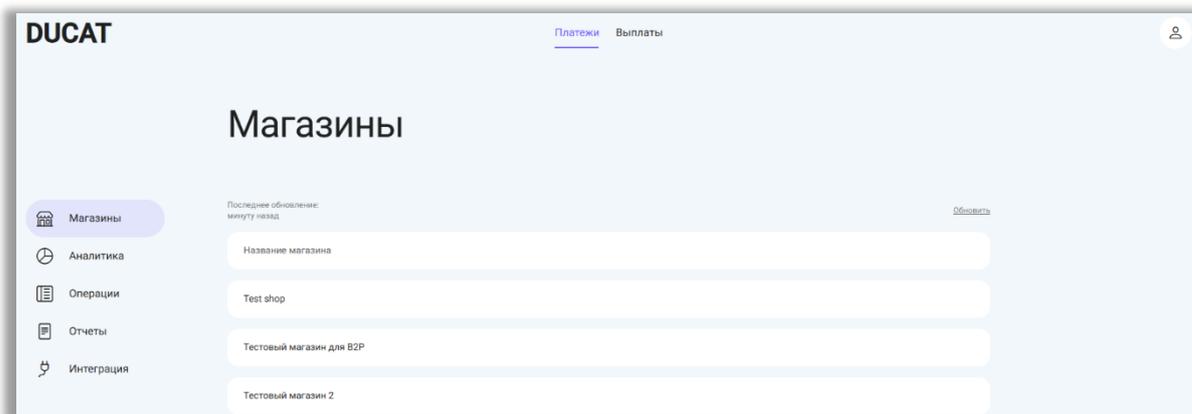


Рисунок 14 — Главный экран ЛК

В профиле пользователя ЛК, расположенном в правом верхнем углу, доступно (рис. 15):

- Изменение пароля.
- Управление сессиями.
- Настройка двухэтапной аутентификации.
- Выход из ЛК.

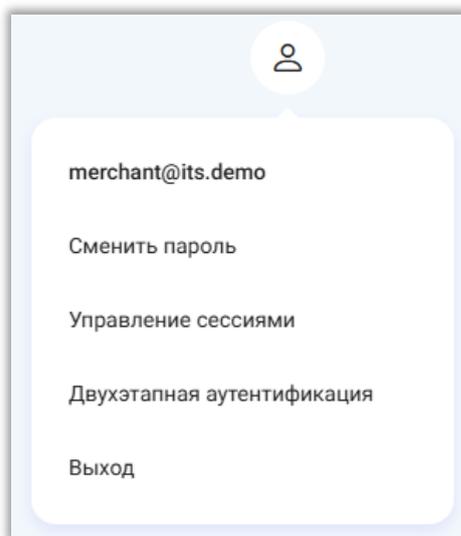


Рисунок 15 — Настройки в профиле пользователя

Фильтры

Для эффективного поиска данных в разделах ЛК используйте кнопки фильтров.

⚠ После выбора параметров фильтра нажмите кнопку «Сохранить». Чтобы убрать фильтр, нажмите «Очистить»/ «Сбросить параметры».

Последние 90 дней (рис. 16) — поиск за определенный период. Для удобства в фильтр добавлены наиболее часто используемые периоды просмотра: последние 365 дней, последние 90 дней, последние 30 дней, последние 24 часа. По умолчанию фильтр всегда показывает данные за последние 90 дней.

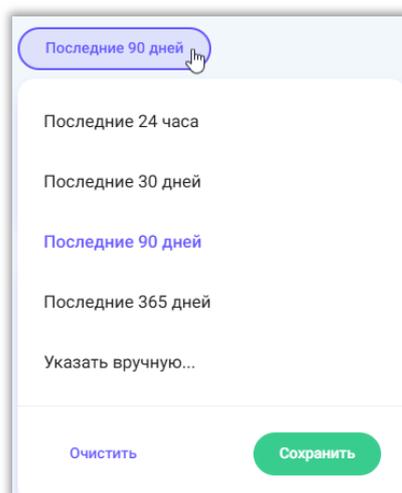


Рисунок 16 — Фильтр «Последние 90 дней»

Чтобы выбрать собственный период (рис. 17), нажмите «Указать вручную», отметьте нужные даты в календаре и нажмите кнопку «Сохранить».

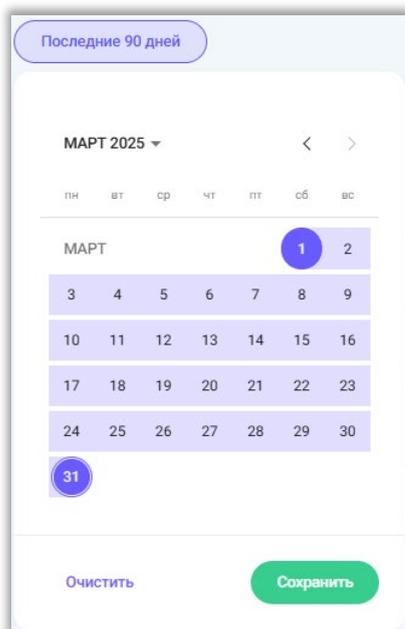


Рисунок 17 — Настройка собственного периода в фильтре «Последние 90 дней»

Инвойсы (рис. 18) — поиск по идентификатору инвойса, используется в [разделе «Операции»](#). Чтобы осуществить поиск по нескольким инвойсам, нажмите на значок «+» и укажите дополнительный идентификатор инвойса.

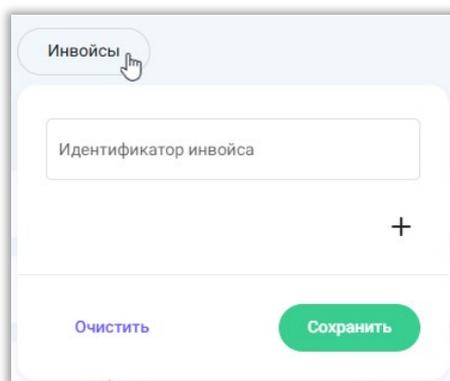


Рисунок 18 — Фильтр «Инвойсы»

Номер заказа (рис. 19) — поиск по идентификатору заказа, используется в разделе «Операции» на вкладках [«Платежи»](#) и [«Инвойсы»](#).

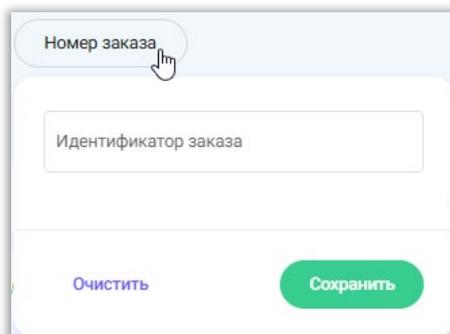


Рисунок 19 — Фильтр «Номер заказа»

Магазины (рис. 20) — поиск по выбранному магазину, используется в разделах [«Аналитика»](#) и [«Операции»](#). Найдите нужный магазин через строку поиска или выберите из списка.

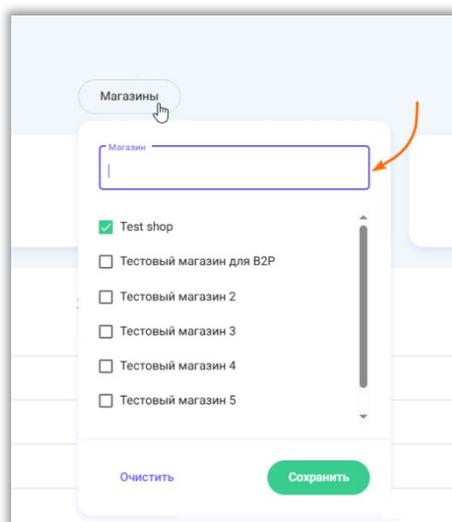


Рисунок 20 — Фильтр «Магазины»

Карта (рис. 21) — поиск по BIN банка-эмитента карты или последним 4 цифрам номера карты. Фильтр используется в разделе «Операции» на [вкладке «Платежи»](#).

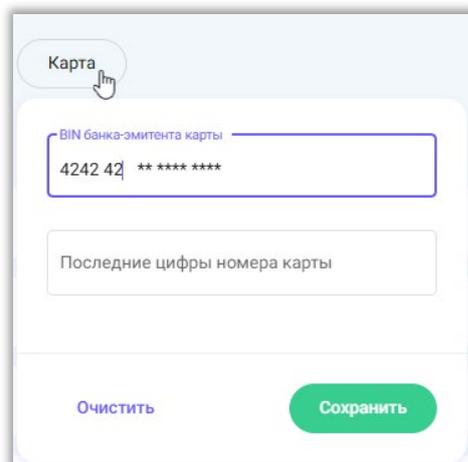


Рисунок 21 — Фильтр «Карта»

Статус инвойса (рис. 22) — поиск по статусам инвойса: Оплачен, Отменен, Погашен, Не оплачен. Фильтр используется в разделе «Операции» на [вкладке «Инвойсы»](#).

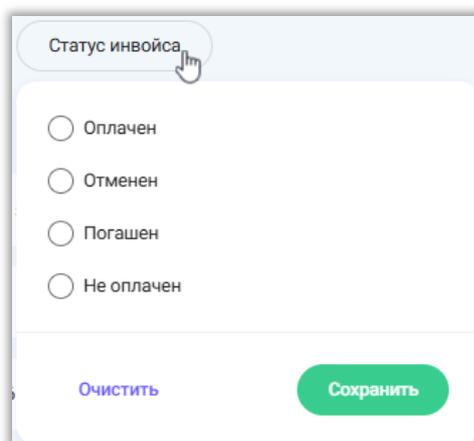


Рисунок 22 — Фильтр «Статус инвойса»

Статус возврата (рис. 23) — поиск по статусам возврата: Успешен, Неуспешен, Запущен. Фильтр используется в разделе «Операции» на [вкладке «Возвраты»](#).

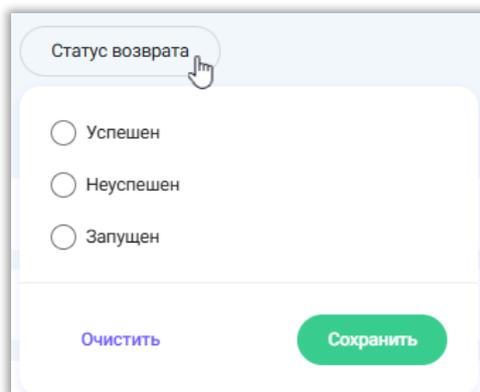


Рисунок 23 — Фильтр «Статус возврата»

Другие фильтры на [вкладке «Платежи»](#) (рис. 24) — поиск по одному или нескольким полям: email или ID Покупателя, идентификатор банковской транзакции (RRN), статус платежа, сумма платежа.

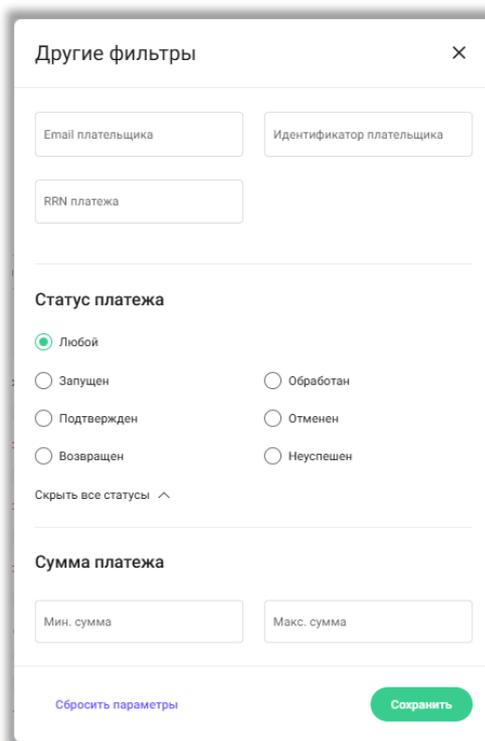


Рисунок 24 — Фильтр «Другие фильтры» на вкладке «Платежи»

Другие фильтры по выплатам в [разделе «Операции»](#) (рис. 25) — поиск по одному или нескольким полям: ID выплаты, ID организации, ID приемника денежных средств, название кошелька, номер заказа, статус и сумма выплаты.

Другие фильтры

Идентификатор вывода Кошелек

Идентификатор владельца Идентификатор приёмника ср...

Номер заказа

Статус вывода

Любой В процессе Успешно Неуспешно

Сумма вывода

Мин. сумма Макс. сумма

Сбросить параметры Сохранить

Рисунок 25 — Фильтр «Другие фильтры» для выплат в разделе «Операции»

Все типы формирования в [разделе «Отчеты»](#) (рис. 26) — поиск по отчету «Реестр операций» и «Разовый платеж».

Все типы формирования

Реестр операций Разовый платеж

Очистить Сохранить

Рисунок 26 — Фильтр «Все типы формирования»

5.1 Меню «Платежи»

Раздел меню предназначен для получения:

- Информации о магазинах.
- Данных о всех платежных операциях.
- Сводной статистики.
- API-ключа организации.

В меню «Платежи» доступно [формирование отчета по реестру операций](#) и создание [Webhook-уведомлений](#).

5.1.1 Раздел «Магазины»

Раздел содержит список всех созданных в сервисе магазинов организации. Процесс создания магазина описан в пункте [Создание магазина](#).

Чтобы посмотреть информацию о магазине, нажмите на его название — откроется карточка магазина (рис. 27). Карточка магазина состоит из нескольких информационных блоков, которые описаны в таблице 17.

Таблица 17 — Содержание карточки магазина

Название блока	Содержание
Информация о магазине	<ul style="list-style-type: none">• Название магазина.• URL адрес сайта магазина.• Дата и время создания
Идентификатор магазина	ID магазина , необходим для идентификации магазина при отправке запросов через API
Информация о юридическом лице компании	<ul style="list-style-type: none">• Наименование юридического лица.• Юридический адрес.• ИНН.• ОГРН/ОГРНИП
Контактная информация	<ul style="list-style-type: none">• Должность ЕОИ/представителя.• Ф.И.О ЕОИ/представителя.• Документ ЕОИ/представителя
Информация о банковском терминале	<ul style="list-style-type: none">• Наименование банка.• БИК.• Корреспондентский счет.• Расчетный счет

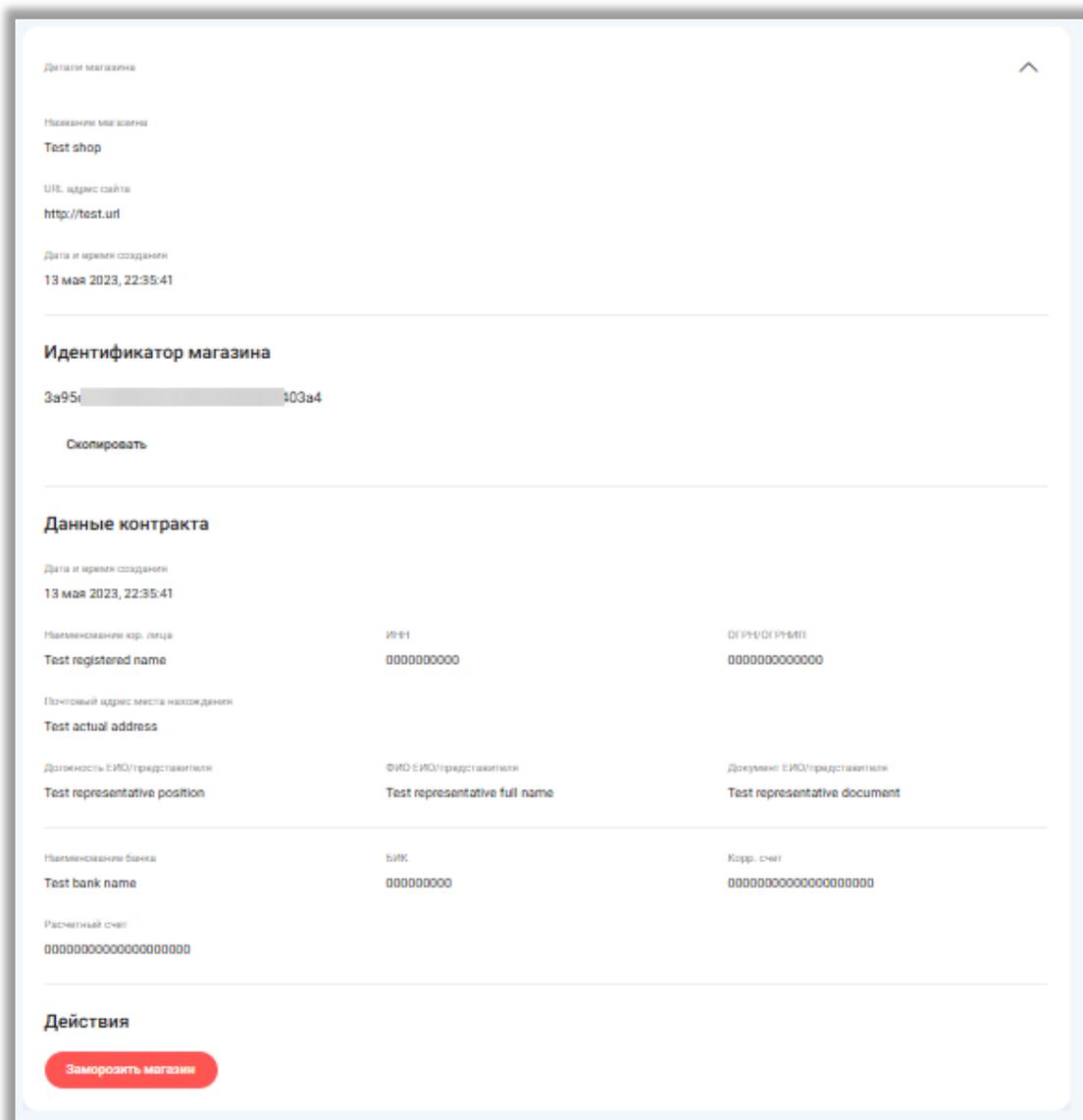


Рисунок 27 — Карточка магазина

Чтобы ограничить проведение платежных операций по конкретному магазину:

1. Перейдите в карточку магазина.
2. В блоке «Действия» нажмите кнопку «Заморозить магазин».
3. Подтвердите действие в появившемся информационном окне.

Магазин не будет удален из сервиса, но проведение платежных операций по нему будет приостановлено. Для разморозки магазина нажмите кнопку «Активировать магазин» в блоке «Действия» карточки магазина.

Чтобы удалить магазин из сервиса, обратитесь в [техническую поддержку DucatPay](#).

5.1.2 Раздел «Аналитика»

В разделе отображается сводная статистика по всем операциям магазинов организации. Статистика формируется на основе данных из [раздела «Операции»](#) и представлена в виде графиков и метрик показателей платежей (рис. 28).

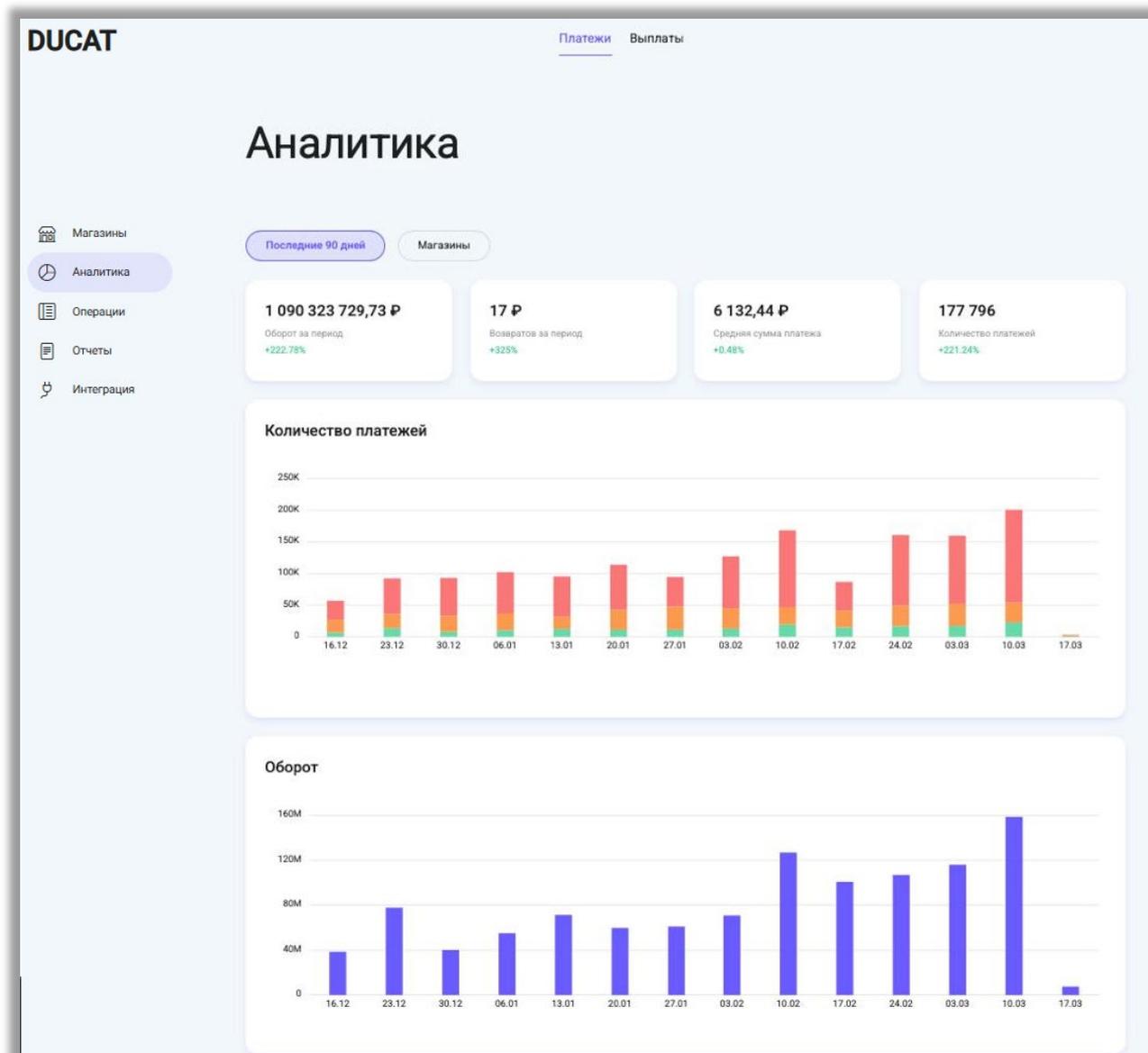


Рисунок 28 — Интерфейс раздела «Аналитика»

Статистика доступна за любой выбранный период и по любому магазину организации. Для настройки отображения используйте кнопки фильтров [Последние 90 дней](#) и [Магазины](#). По умолчанию отображаются данные за последние 90 дней по всем магазинам.

В блоках над графиками (рис. 29) отображается сводная статистика по метрикам:

- Оборот денежных средств за период.
- Сумма возвратов за период.
- Средняя сумма платежа.
- Количество платежей.

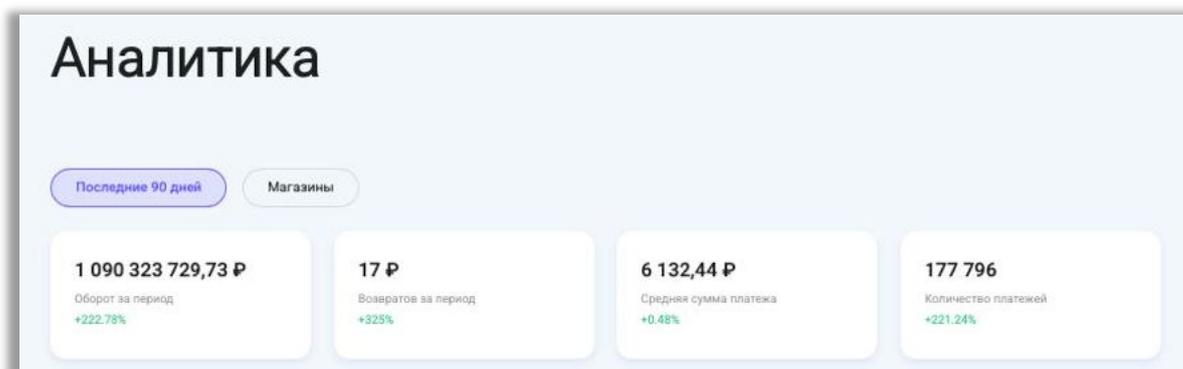


Рисунок 29 — Метрики представленные в блоках

Графиками и диаграммами представлены метрики (рис. 30):

- Количество платежей — зеленым цветом выделены платежи в статусе **Подтвержден**, оранжевым — **Отменен**, красным — **Неуспешен**.
- Оборот — сумма платежей в статусе **Подтвержден** без учета возвратов и отмен.
- Платежные инструменты — на данный момент представлен только один тип оплаты — банковской картой.
- Причины неуспешных платежей — процентное соотношение платежей с ошибкой.

⚠ Если через магазин еще не были проведены платежи, графики и метрики будут отображать сообщение «Нет данных».



Рисунок 30 — Метрики, представленные графиками и диаграммами

5.1.3 Раздел «Операции»

Раздел содержит информацию о проведенных и созданных в сервисе платежных операциях, сгруппированную по вкладкам (рис. 31):

- [Платежи](#)
- [Инвойсы](#)
- [Возвраты](#)

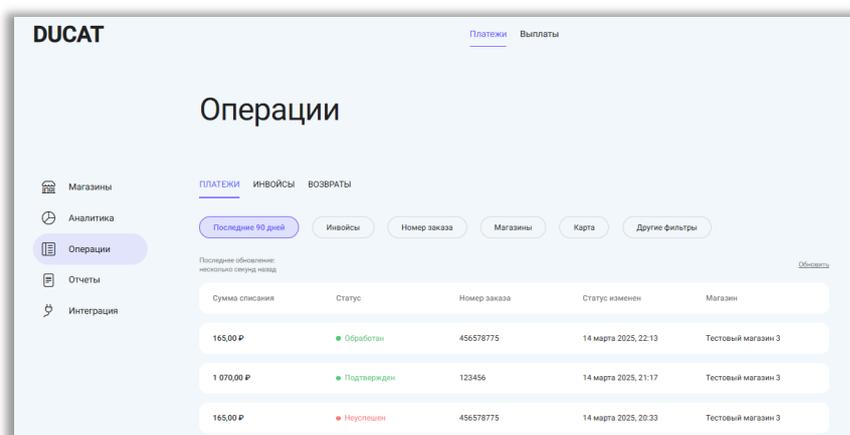


Рисунок 31 — Интерфейс раздела «Операции»

Вкладка «Платежи»

Данные о платежах представлены в виде таблицы и содержат (рис. 32):

- Сумму списания;
- Статус платежа;
- Номер заказа;
- Дату изменения статуса;
- Название магазина.

Операции

ПЛАТЕЖИ ИНВОЙСЫ ВОЗВРАТЫ

Последние 90 дней Инвойсы Номер заказа Магазины Карта Другие фильтры

Последнее обновление: несколько секунд назад [Обновить](#)

Сумма списания	Статус	Номер заказа	Статус изменен	Магазин
165,00 Р	● Обработан	456578775	14 марта 2025, 22:13	Тестовый магазин 3
1 070,00 Р	● Подтвержден	123456	14 марта 2025, 21:17	Тестовый магазин 3
165,00 Р	● Неуспешен	456578775	14 марта 2025, 20:33	Тестовый магазин 3

Рисунок 32 — Интерфейс вкладки «Платежи»

Чтобы найти интересующий платеж, используйте расположенные над таблицей кнопки фильтров: [Последние 90 дней](#), [Инвойсы](#), [Номер заказа](#), [Магазины](#), [Карта](#) и [Другие фильтры](#). По умолчанию в таблице отображаются данные за последние 90 дней по всем магазинам.

Чтобы посмотреть детали платежа, нажмите на строку с интересующим платежом в таблице — откроется карточка платежа.

Карточка платежа содержит (рис. 33):

- Сумму платежа;
- Статус платежа;
- Сумму зачисления на банковский счет Мерчанта;
- Сумму комиссии сервиса;
- Способ оплаты;
- Тип платежа;
- Email Покупателя;
- Номер заказа;
- Название магазина.

В поле «Дополнительная информация» указан код авторизации (Approval Code) и идентификатор банковской транзакции (RRN).

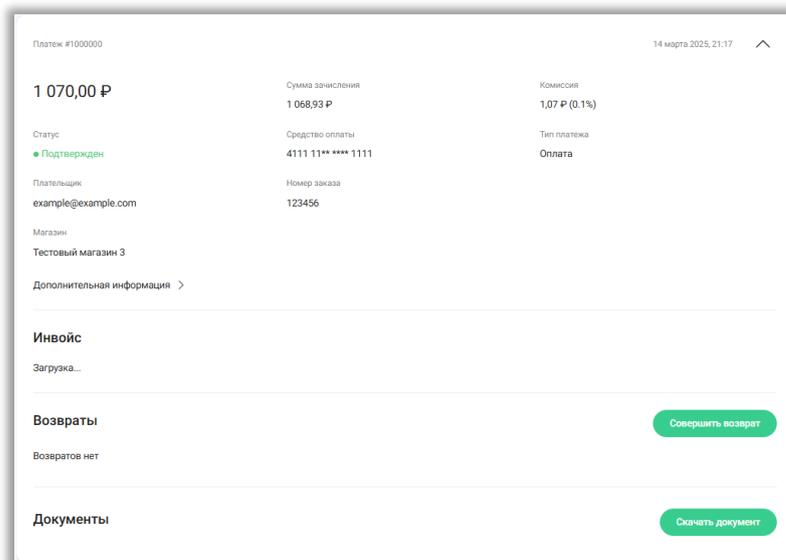


Рисунок 33 — Карточка платежа

Через карточку платежа можно полностью или частично вернуть платеж. В блоке «Возвраты» нажмите кнопку «Совершить возврат» и проделайте действия из пункта [Возврат платежа в ЛК DucatPay](#).

⚠ Если кнопка «Совершить возврат» недоступна, проверьте статус платежа. Вернуть платеж можно только из статуса **Подтвержден**.

В карточке платежа можно сформировать и скачать платежное поручение в формате *.pdf (рис. 34). Для этого в блоке «Документы» нажмите кнопку «Скачать документ».

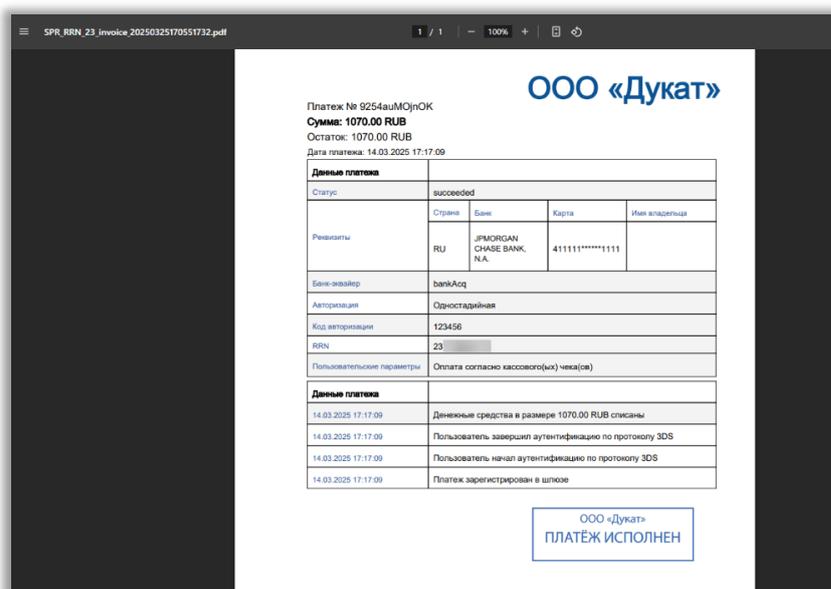


Рисунок 34 — Пример платежного поручения в формате *.pdf

Сформированное платежное поручение также сохранится в [разделе «Отчеты»](#).

Вкладка «Инвойсы»

Информация об инвойсах представлена в виде таблицы и содержит данные (рис. 35):

- Наименование;
- Стоимость;
- Статус;
- Номер заказа;
- Дата создания инвойса;
- Название магазина.

Операции

ПЛАТЕЖИ **ИНВОЙСЫ** ВОЗВРАТЫ

Последние 90 дней Инвойсы Номер заказа Магазины Статус инвойса

Последнее обновление: несколько секунд назад [Обновить](#)

Наименование	Стоимость	Статус	Номер заказа	Создан	Магазин
Тестовый платеж Ducat	165,00 ₽	● Не оплачен	456578775	14 марта 2025, 22:13	Тестовый магазин 3
Тестовый платеж Ducat	1 070,00 ₽	● Оплачен	123456	14 марта 2025, 21:16	Тестовый магазин 3
Тестовый платеж Ducat	165,00 ₽	● Отменен	456578775	14 марта 2025, 20:33	Тестовый магазин 3

Рисунок 35 — Интерфейс вкладки «Инвойсы»

Чтобы найти интересующий инвойс, используйте расположенные над таблицей кнопки фильтров: [Последние 90 дней](#), [Инвойсы](#), [Номер заказа](#), [Магазины](#) и [Статус инвойса](#). По умолчанию в таблице отображаются данные за последние 90 дней по всем магазинам.

Чтобы раскрыть детали по инвойсу, нажмите на строку с ним в таблице — откроется карточка инвойса (рис. 36).

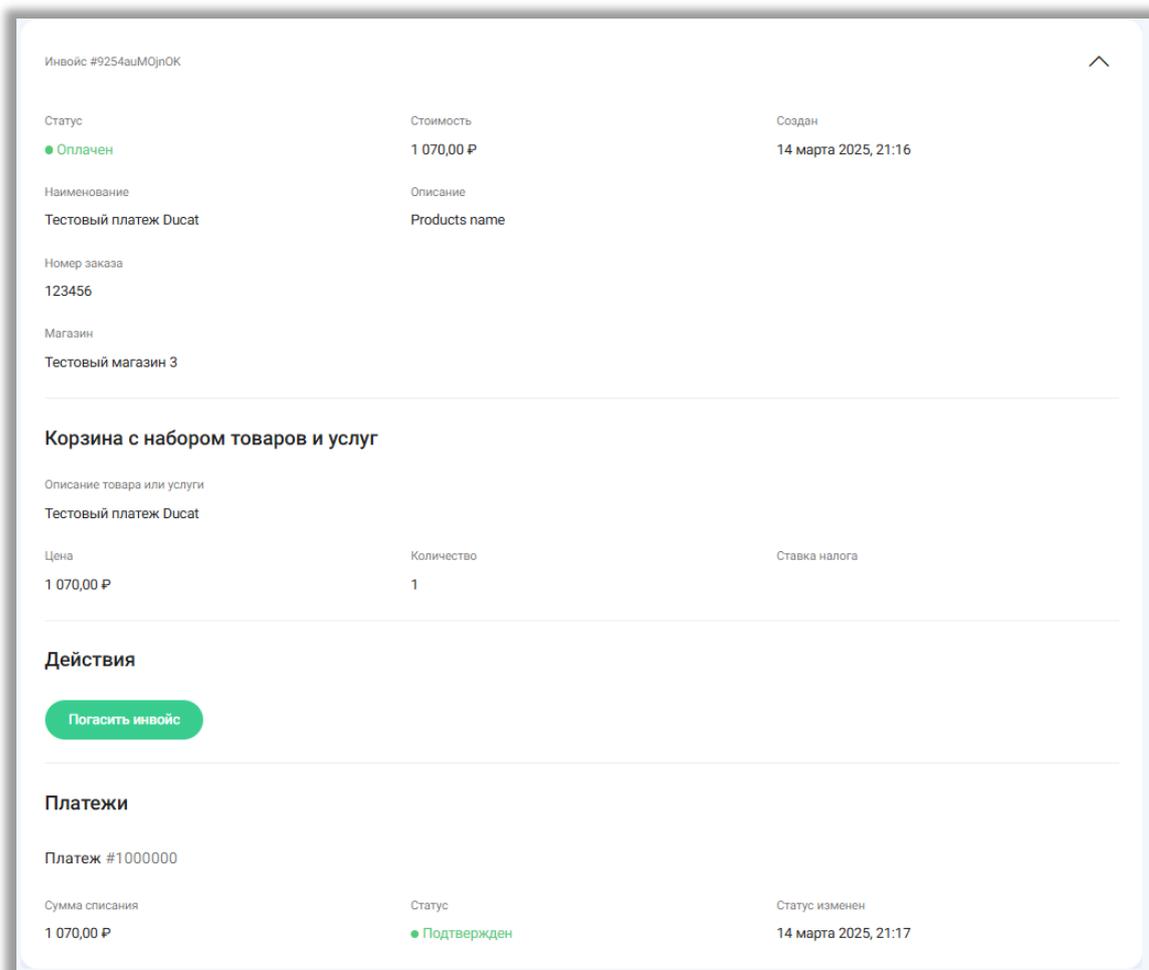


Рисунок 36 — Карточка инвойса

Карточка инвойса содержит:

- Статус инвойса;
- Стоимость;
- Дату и время создания;
- Наименование инвойса;
- Описание назначения инвойса;
- Номер заказа;
- Название магазина;
- Корзину с набором товаров и услуг.

При проведении двухстадийного платежа, после получения подтверждения оплаты от Покупателя инвойс можно погасить. Для этого в блоке «Действия» карточки инвойса нажмите кнопку «Погасить инвойс». Статус инвойса изменится с **Оплачен** на **Погашен**.

В блоке «Платежи» отображается информация только об успешном платеже по инвойсу со статусом **Подтвержден**.

Отмена инвойса

⚠ Отменить инвойс можно только из статуса **Не оплачен**.

1. Перейдите в карточку инвойса.
2. В блоке «Действия» нажмите кнопку «Отменить инвойс» (рис. 37).

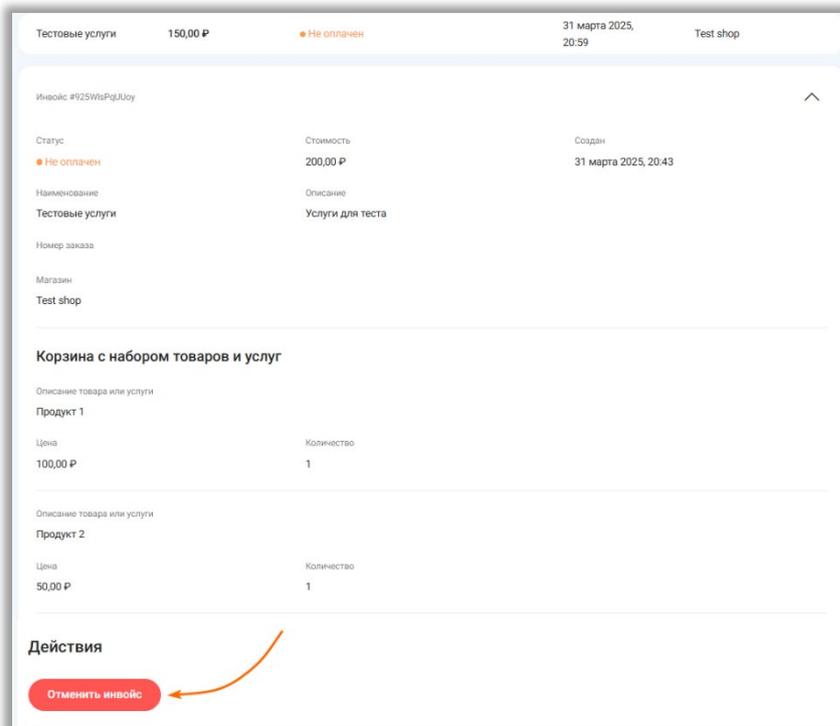


Рисунок 37 — Кнопка «Отменить инвойс» в карточке инвойса

3. Укажите причину отмены (рис. 38) и нажмите кнопку «Подтвердить».

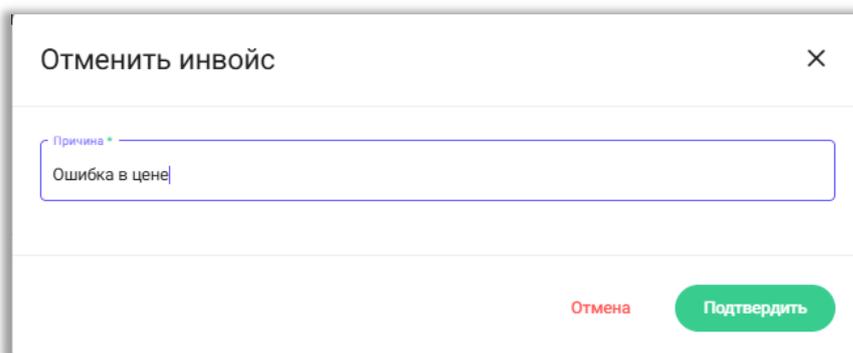


Рисунок 38 — Окно для ввода причины отмены инвойса

4. Статус инвойса изменится с **Не оплачен** на **Отменен**.

Вкладка «Возвраты»

Информация о возвратах представлена в виде таблицы и содержит (рис. 39):

- Сумму возврата;
- Статус возврата;
- Дату и время изменения статуса;
- Название магазина.

Операции

ПЛАТЕЖИ ИНВОЙСЫ **ВОЗВРАТЫ**

Последние 90 дней Магазины Инвойсы Статус возврата

Последнее обновление: несколько секунд назад Обновить

Сумма возврата	Статус	Статус изменен	Магазин
60,00 Р	● Успешно	14 марта 2025, 01:49	Test shop
10,00 Р	● Успешно	23 декабря 2024, 21:08	Test shop

Рисунок 39 — Интерфейс вкладки «Возвраты»

Чтобы найти интересующий возврат, используйте расположенные над таблицей кнопки фильтров: [Последние 90 дней](#), [Магазины](#), [Инвойсы](#) и [Статус возврата](#). По умолчанию в таблице отображаются данные за последние 90 дней по всем магазинам.

Для просмотра деталей возврата нажмите на строку с интересующим возвратом в таблице — откроется карточка возврата (рис. 40).

Карточка возврата состоит из нескольких блоков и содержит информацию из таблицы 18.

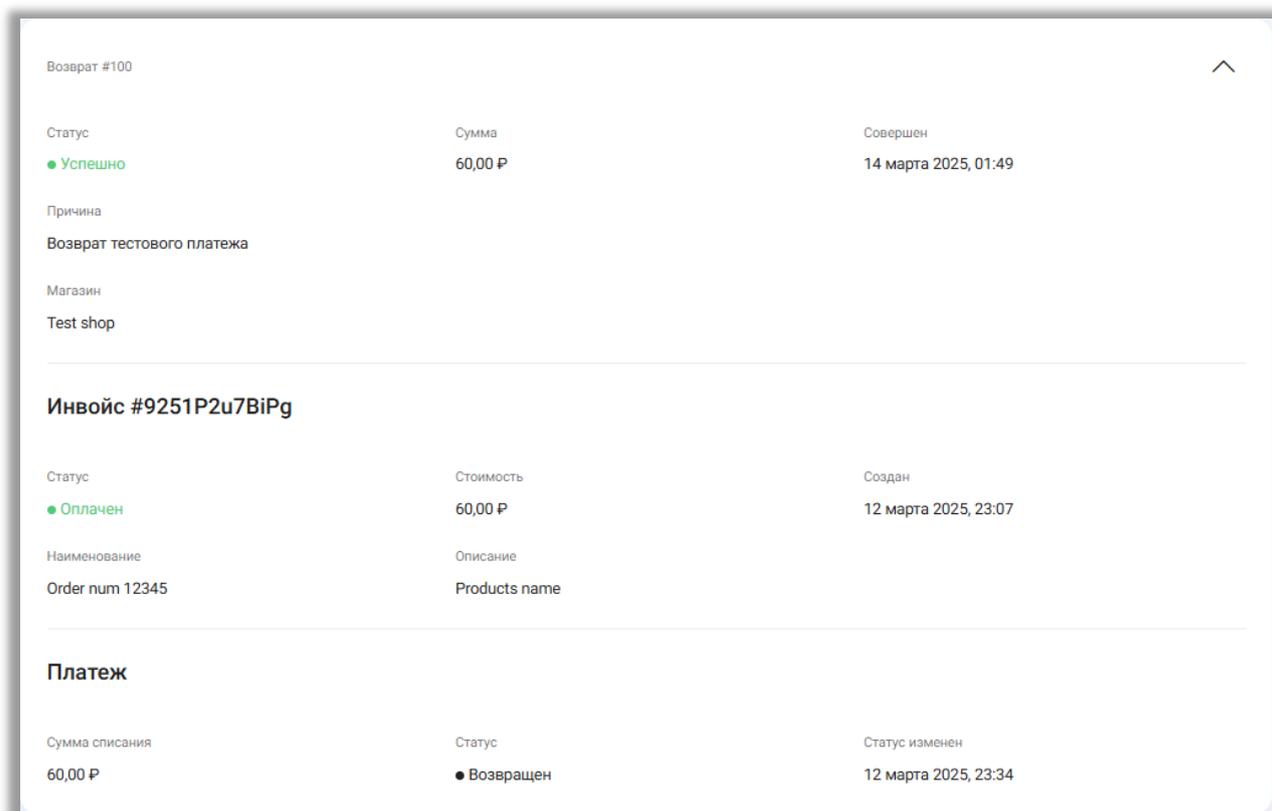


Рисунок 40 — Карточка возврата

Таблица 18 — Данные из карточки возврата

Название блока	Содержание
Детали возврата	<ul style="list-style-type: none"> • Статус возврата; • Сумма возврата; • Дата возврата; • Причина возврата; • Название магазина
Инвойс	<ul style="list-style-type: none"> • ID инвойса; • Статус инвойса; • Стоимость; • Дата создания инвойса; • Наименование; • Описание
Платеж	<ul style="list-style-type: none"> • Сумма списания; • Статус возврата; • Дата изменения статуса

5.1.4 Раздел «Отчеты»

Раздел предназначен для создания и просмотра отчетов по **реестру операций** магазина за выбранный период.

Информация о созданных отчетах отображается в виде таблицы и содержит (рис. 41):

- Дату создания отчета;
- Статус отчета;
- Тип формирования;
- Период отчетности.

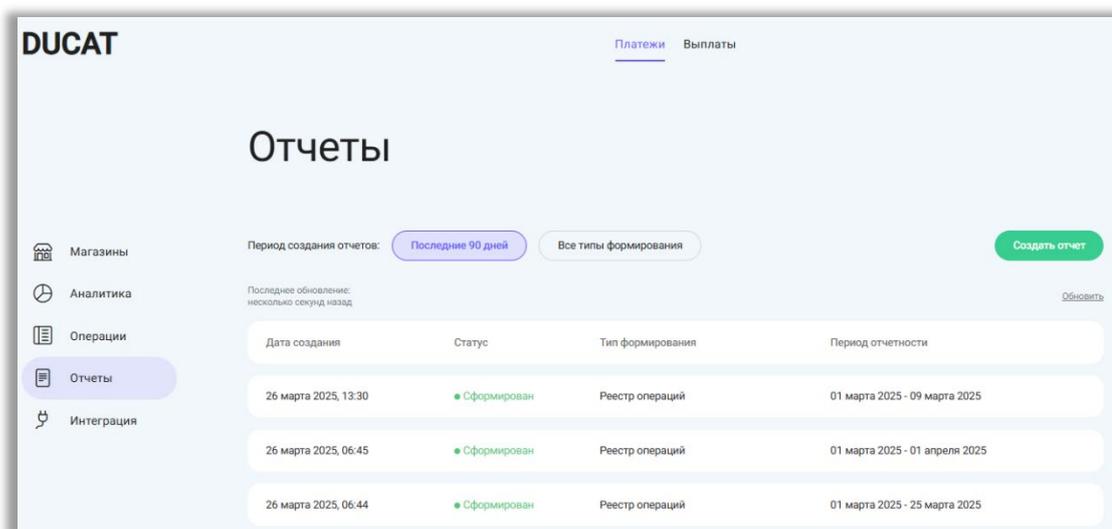


Рисунок 41 — Интерфейс раздела «Отчеты»

Сформированные платежные поручения из карточек платежей отображаются в таблице отчетов с типом формирования «Разовый платеж».

Для поиска созданного отчета используйте кнопки фильтров Последние 90 дней и Все типы формирования, расположенные над таблицей. По умолчанию в таблице отображаются данные за последние 90 дней по всем типам отчетов.

Чтобы посмотреть отчет, нажмите на строку с созданным отчетом в таблице — откроется карточка отчета (рис. 42).

Карточка отчета содержит:

- Статус отчета;
- Тип формирования;
- Дату создания;
- Период отчетности;
- Название магазина;
- Сформированный отчет в формате *.xlsx.

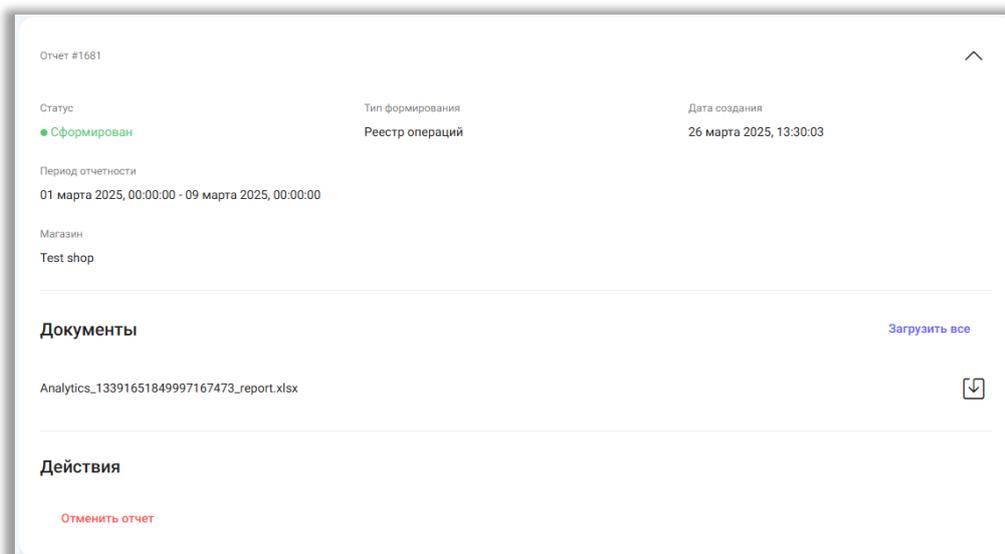


Рисунок 42 — Карточка отчета

Создание отчета

1. Нажмите кнопку «Создать отчет» (рис. 43).

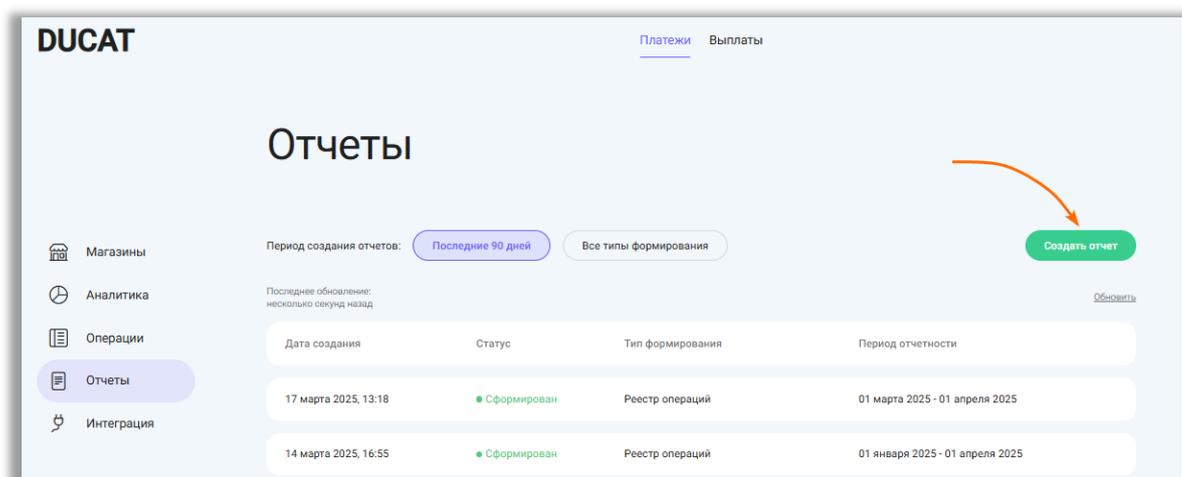


Рисунок 43 — Кнопка «Создать отчет» в разделе «Отчеты»

2. Укажите дату и время начала и окончания периода. Выберите магазин, по которому нужно сформировать отчет (рис. 44).

Параметры формирования отчета

Отчет с реестром операций за выбранный период

Начало периода * 1.3.2025

Время * 00:00:00

Конец периода * 1.4.2025

Время * 00:00:00

Магазин Test shop

Сформировать

Рисунок 44 — Параметры формирования отчета

3. Нажмите кнопку «Сформировать».
4. Отчет появится в таблице со статусом **В процессе**. Чтобы актуализировать информацию о статусе отчета, нажмите «Обновить» (рис. 45).
5. Когда сервис закончит формирование отчета, статус в таблице изменится на **Сформирован**.

Отчеты

Период создания отчетов: Последние 90 дней Все типы формирования Создать отчет

Последнее обновление: несколько секунд назад Обновить

Дата создания	Статус	Тип формирования	Период отчетности
17 марта 2025, 13:18	В процессе	Реестр операций	01 марта 2025 - 01 апреля 2025
14 марта 2025, 16:55	Сформирован	Реестр операций	01 января 2025 - 01 апреля 2025

Рисунок 45 — Карточка отчета со статусом В процессе и кнопка «Обновить»

Отчет доступен для скачивания в формате *.xlsx (рис. 46) при нажатии «Загрузить все» или значка  в блоке «Документы» карточки отчета.

MerchantContractName	OrderId	RRN	Date	Amount	CurrencyCode	TransactionStatus	CardHolder	PANMask	CardTypeName
Test shop	123456	5,06155E+11	02.03.2025 18:35:33	1070	RUB	failed	Ivanov Ivan Ivanovich	424242*****4242	VISA
Test shop	123457	5,06155E+11	03.03.2025 13:52:56	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4243	VISA
Test shop	123458	5,06155E+11	03.03.2025 15:38:17	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4244	VISA
Test shop	123459	5,06155E+11	04.03.2025 20:40:09	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4245	VISA
Test shop	123460	5,06155E+11	04.03.2025 20:44:16	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4246	VISA
Test shop	123461	5,06155E+11	04.03.2025 20:49:24	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4247	VISA
Test shop	123462	5,06155E+11	04.03.2025 21:23:14	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4248	VISA
Test shop	123463	5,06155E+11	04.03.2025 21:25:29	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4249	VISA
Test shop	123464	5,06155E+11	04.03.2025 21:26:42	1070	RUB	failed	Ivanov Ivan Ivanovich	424242*****4250	VISA
Test shop	123465	5,06155E+11	04.03.2025 21:27:12	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4251	VISA
Test shop	123466	5,06155E+11	04.03.2025 21:45:03	1070	RUB	failed	Ivanov Ivan Ivanovich	424242*****4252	VISA
Test shop	123467	5,06155E+11	04.03.2025 21:47:21	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4253	VISA
Test shop	123468	5,06155E+11	04.03.2025 23:35:00	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4254	VISA
Test shop	123469	5,06155E+11	04.03.2025 23:35:36	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4255	VISA
Test shop	123470	5,06155E+11	04.03.2025 23:37:18	1070	RUB	failed	Ivanov Ivan Ivanovich	424242*****4256	VISA
Test shop	123471	5,06155E+11	04.03.2025 23:37:36	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4257	VISA
Test shop	123472	5,06155E+11	05.03.2025 10:12:59	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4258	VISA
Test shop	123473	5,06155E+11	05.03.2025 10:16:30	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4259	VISA
Test shop	123474	5,06155E+11	05.03.2025 13:31:23	200	RUB	failed	Ivanov Ivan Ivanovich	424242*****4260	VISA
Test shop	123475	5,06155E+11	06.03.2025 13:55:21	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4261	VISA
Test shop	123476	5,06155E+11	06.03.2025 14:00:11	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4262	VISA
Test shop	123477	5,06155E+11	06.03.2025 15:08:23	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4263	VISA
Test shop	123478	5,06155E+11	06.03.2025 15:22:54	1070	RUB	captured	Ivanov Ivan Ivanovich	424242*****4264	VISA
Test shop	123479	5,06155E+11	06.03.2025 19:39:22	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4265	VISA
Test shop	123480	5,06155E+11	06.03.2025 20:02:21	200	RUB	captured	Ivanov Ivan Ivanovich	424242*****4266	VISA

Рисунок 46 — Отчет в формате *.xlsx

Описание полей отчета по реестру операций приведено в таблице 19.

Таблица 19 — Описание полей отчета реестр операций

Название поля	Описание
MerchantContractName	Название магазина
OrderId	Номер заказа
RRN	Идентификатор банковской транзакции
Date	Дата и время операции
Amount	Сумма операции
CurrencyCode	Валюта
TransactionStatus	Статус платежа
CardHolder	Владелец карты
PANMask	Номер банковской карты
CardTypeName	Тип платежной системы
PaymentMethod	Тип платежного средства
BankName	Наименование банка
ProcessingName	Платежная система
UserData	Описание продукта
PayPageParams	Сервисное поле
Product	Наименование продукта
ClientFIO	Ф.И.О. Покупателя
Contract	Номер договора
Terminal	Терминал
Fee	Комиссия
ProviderOrderId	Номер заказа провайдера

Чтобы удалить отчет из таблицы:

1. Откройте карточку отчета.
2. В блоке «Действия» нажмите «Отменить отчет».
3. Подтвердите действие в появившемся окне.

После подтверждения отчет будет удален из таблицы.

5.1.5 Раздел «Интеграция»

Раздел предназначен для настройки интеграции с DucatPay: получения API-ключа и настройки Webhook-уведомлений.

Вкладка «API-ключ»

Во вкладке хранится API-ключ организации, необходимый для аутентификации запросов к API. Подробнее в пункте [Получение API-ключа и ID организации](#).

Вкладка «Webhooks»

Чтобы автоматизировано отслеживать состояния и события по инвойсам и платежам, настройте Webhook-уведомления. Управлять Webhook-уведомлениями можно с помощью [запросов к API](#) или через ЛК.

Создание Webhook-уведомления по событиям платежей

1. Нажмите кнопку «Установить webhook» (рис. 47).

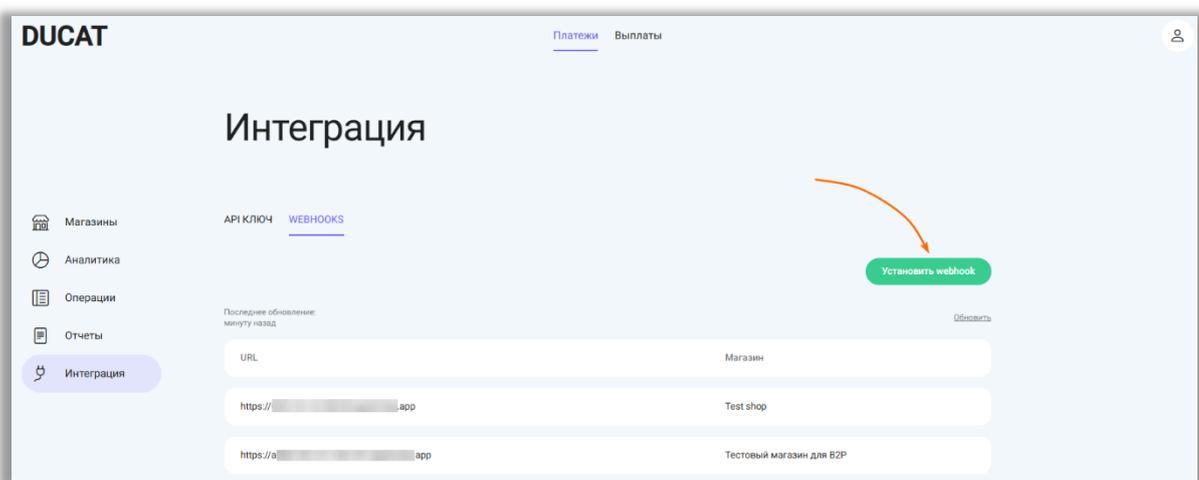


Рисунок 47 — Кнопка «Установить webhook» для событий платежей

2. Выберите магазин, по которому хотите получать уведомления (рис. 48).
3. Укажите URL-адрес, на который необходимо отправлять уведомления.
4. Выберите одно или несколько событий из списка, о которых необходимо оповещать.
5. Нажмите кнопку «Установить».

Установить новый webhook

Магазин *
Test shop

URL, на который будут поступать оповещения о произошедших событиях *
https://test_url.ru

Предмет оповещений

События инвойса

События
Набор типов событий, о которых следует оповещать

InvoiceCreated (создан новый инвойс)
 InvoicePaid (инвойс перешел в состояние "Оплачен")
 InvoiceCancelled (инвойс отменен по истечению срока давности)
 InvoiceFulfilled (инвойс успешно погашен)
 PaymentStarted (создан платеж)

Установить

Рисунок 48 — Настройка параметров Webhook-уведомления по платежам

При нажатии на Webhook-уведомление откроются данные по нему (рис. 49):

- URL-адрес, на который поступают уведомления.
- Название магазина.
- Набор событий, по которым приходят уведомления.
- Публичный ключ — нужен для проверки подлинности приходящих уведомлений на указанный URL-адрес. Чтобы скопировать, нажмите «Скопировать ключ».

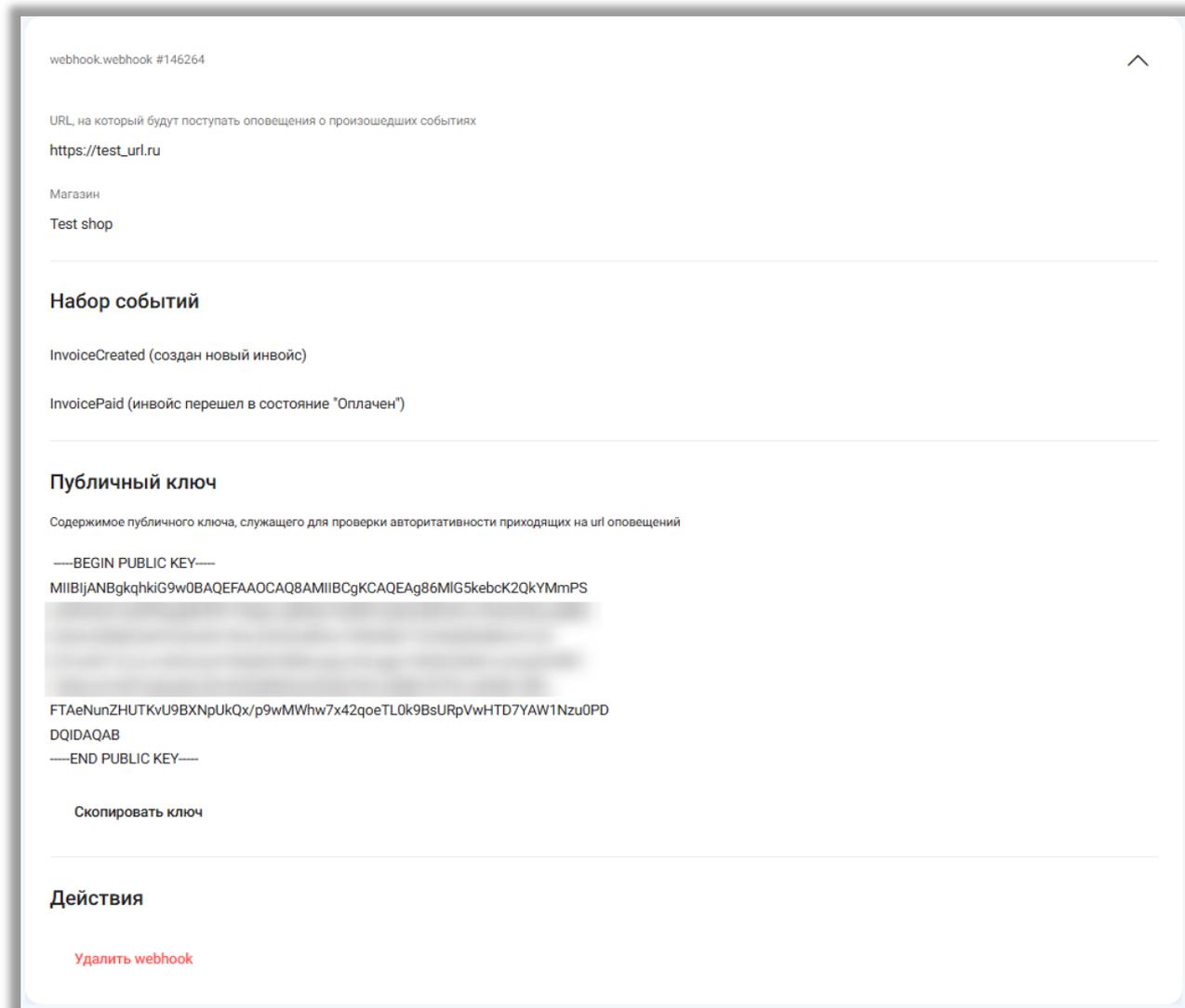


Рисунок 49 — Информация о Webhook-уведомлении по событиям платежей

Чтобы удалить Webhook-уведомление, нажмите «Удалить webhook» в блоке «Действия».

⚠ Изменить параметры созданного Webhook-уведомления нельзя. Рекомендуем удалить Webhook, который нужно изменить, и создать новый.

5.2 Меню «Выплаты»

Раздел меню предназначен для получения данных о [виртуальных кошельках](#) и операциях [выплат](#). Настройка Webhook-уведомлений по событиям выплат доступна в [разделе «Интеграция»](#).

5.2.1 Раздел «Кошельки»

Раздел содержит список всех созданных в сервисе виртуальных кошельков организации (рис. 50). Процесс создания виртуального кошелька описан в пункте [Создание виртуального кошелька](#).

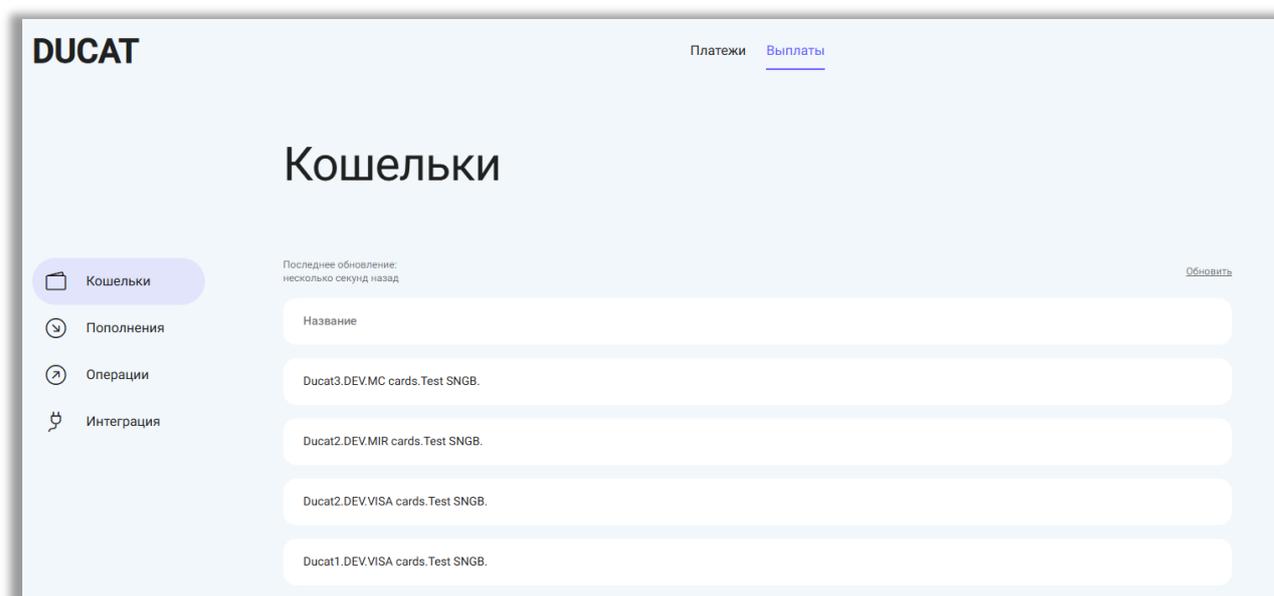


Рисунок 50 — Интерфейс раздела «Кошельки»

Чтобы посмотреть информацию о виртуальном кошельке, нажмите на его название — откроется карточка виртуального кошелька (рис. 51). Карточка разделена на несколько информационных блоков, описанных в таблице 20.

Нажмите «Показать еще» в карточке кошелька, чтобы отобразить все операции выплат по виртуальному кошельку. Удобный поиск и просмотр информации о выплатах доступен в [разделе «Операции»](#).

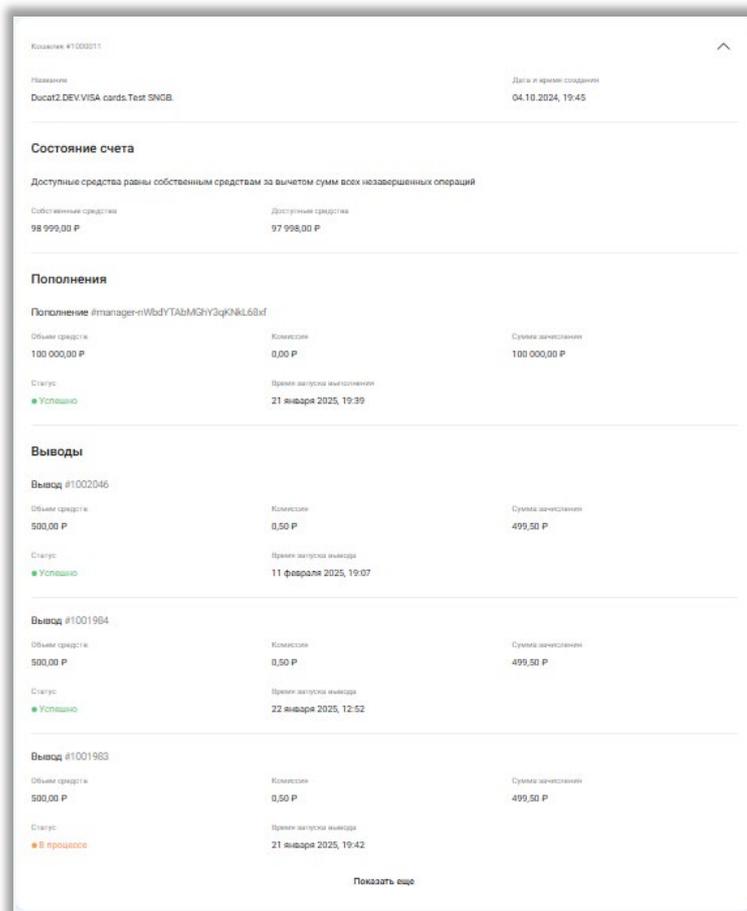


Рисунок 51 — Карточка кошелька

Таблица 20 — Содержание карточки виртуального кошелька

Блок	Назначение	Содержание
Информация о виртуальном кошельке	Общая информация	<ul style="list-style-type: none"> • Название кошелька; • Дата и время создания
Состояние счета	Просмотр баланса виртуального кошелька	<ul style="list-style-type: none"> • Собственные средства; • Доступные средства = Собственные средства — Сумма незавершенных операций
Пополнения	Список всех пополнений виртуального баланса кошелька	<ul style="list-style-type: none"> • Объем средств; • Комиссия; • Сумма зачисления; • Статус; • Дата и время запуска выполнения
Выводы	Список всех выплат с виртуального кошелька	Каждая выплата содержит поля: <ul style="list-style-type: none"> • ID выплаты. • Объем средств. • Комиссия сервиса; • Сумма зачисления; • Статус выплаты; • Дата и время запуска выполнения

5.2.2 Раздел «Пополнения»

Раздел содержит данные о пополнениях НДС баланса виртуального кошелька. Процесс пополнения описан в пункте [Пополнение баланса виртуального кошелька](#).

⚠ Баланс виртуального кошелька нужен для учета НДС в сервисе и может не соответствовать реальной сумме денежных средств на счете организации.

Информация о пополнениях представлена в виде таблицы и содержит данные (рис. 52):

- Сумму пополнения;
- Статус пополнения;
- Дату пополнения;
- Название кошелька.

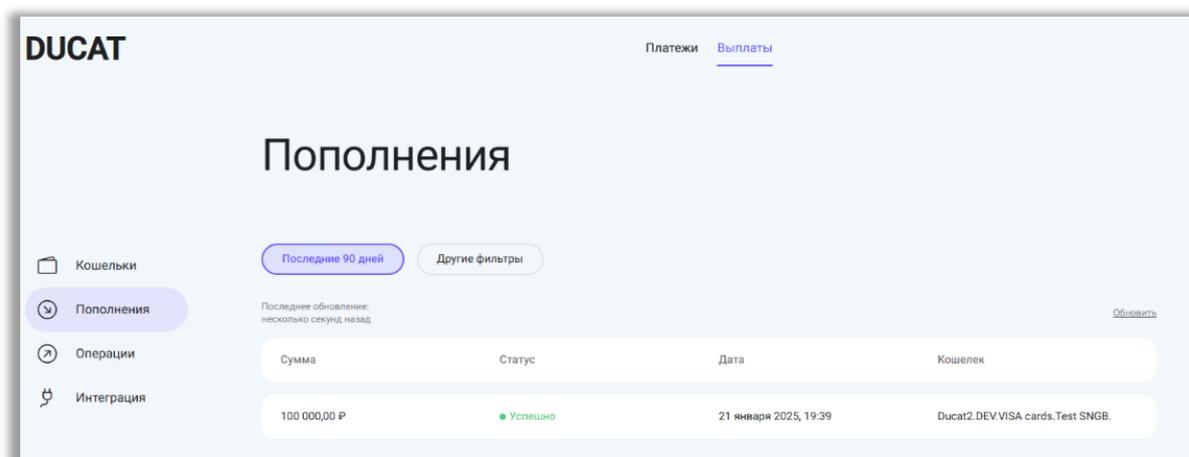


Рисунок 52 — Интерфейс раздела «Пополнения»

Чтобы раскрыть информацию о пополнении, нажмите на выбранную строку в таблице — откроется карточка пополнения баланса (рис. 53).

Карточка пополнения баланса содержит:

- Сумму зачисления НДС;
- Статус пополнения;
- Название кошелька;
- Комиссию сервиса (при наличии).

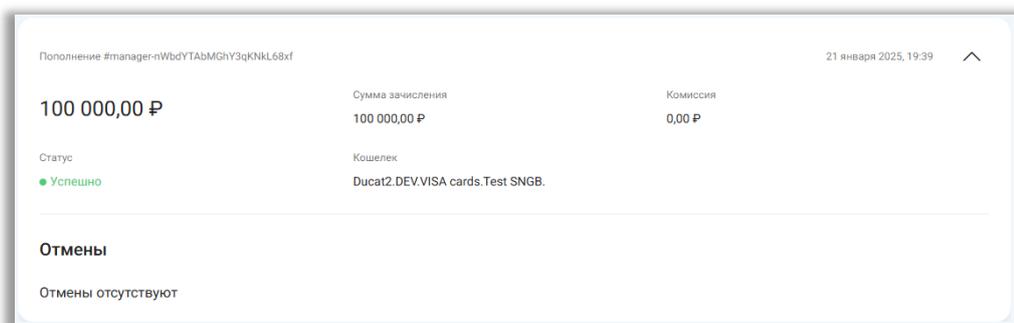


Рисунок 53 — Карточка пополнения баланса

Чтобы отменить пополнение баланса, обратитесь в [техническую поддержку DucatPay](#). Информация об отмене пополнения баланса отображается в блоке «Отмены» карточки пополнения баланса.

5.2.3 Раздел «Операции»

Раздел предназначен для мониторинга операций по процессу выплат, описанному в пункте [Процесс проведения выплаты](#). Информация о выплатах представлена в виде таблицы, содержащей (рис. 54):

- Название кошелька;
- Номер заказа — заданный Мерчантом при создании выплаты;
- Дату создания выплаты;
- Статус выплаты;
- Сумму выплаты.

⚠ Если информация о выплате не отображается в таблице, нажмите «Обновить» над правым верхним углом таблицы.

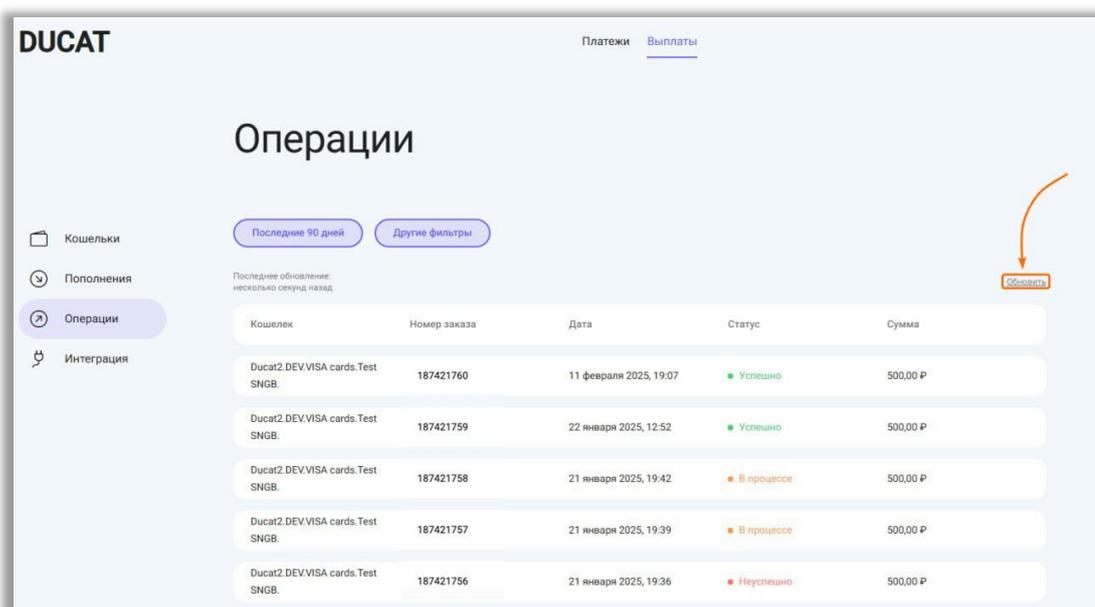


Рисунок 54 — Интерфейс раздела «Операции» в меню «Выплаты»

Для поиска выплаты используйте кнопки фильтров [Последние 90 дней](#) и [Другие фильтры](#), расположенные над таблицей. По умолчанию в таблице отображаются данные за последние 90 дней по всем кошелькам.

Чтобы посмотреть детали выплаты, нажмите на выбранную строку в таблице — откроется карточка выплаты (рис. 55).

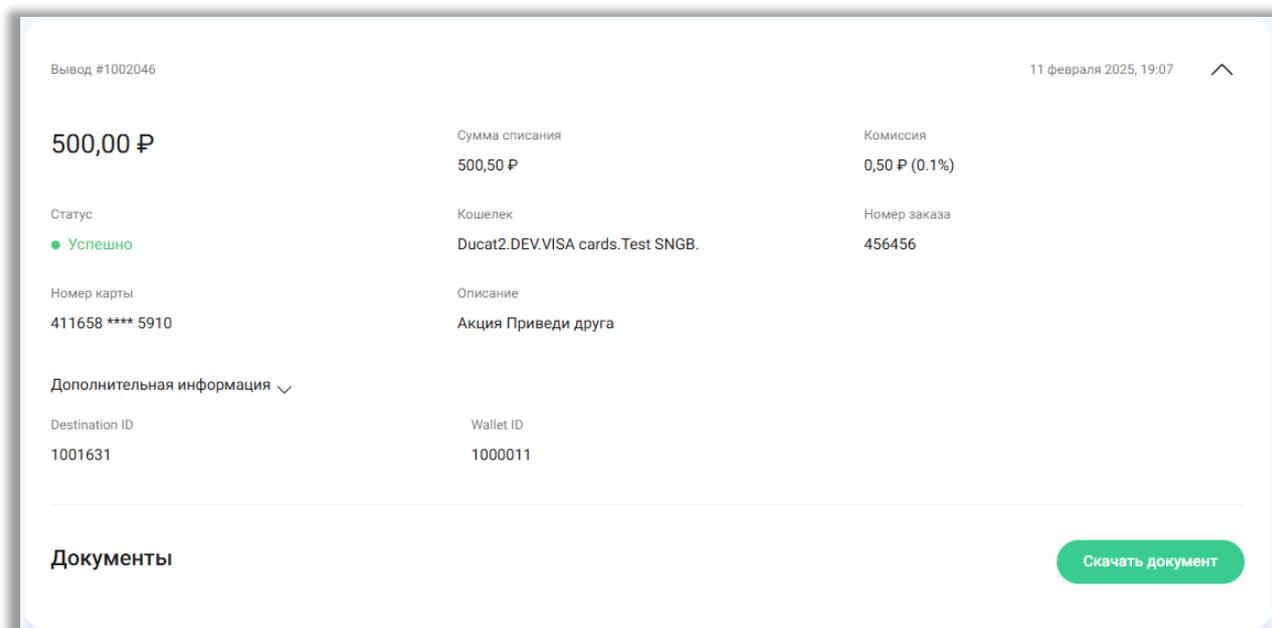


Рисунок 55 — Карточка выплаты

Карточка содержит:

- Сумму выплаты;
- Сумму списания с баланса виртуального кошелька;
- Комиссию сервиса;
- Статус выплаты;
- Название кошелька;
- Номер заказа;
- Номер карты физического лица;
- Описание выплаты.

В блоке «Дополнительная информация» указан идентификатор получателя выплаты (Destination ID) и идентификатор кошелька (Wallet ID).

В карточке выплаты можно сформировать и скачать платежное поручение в формате *.pdf (рис. 56). Для этого в блоке «Документы» нажмите кнопку «Скачать документ».

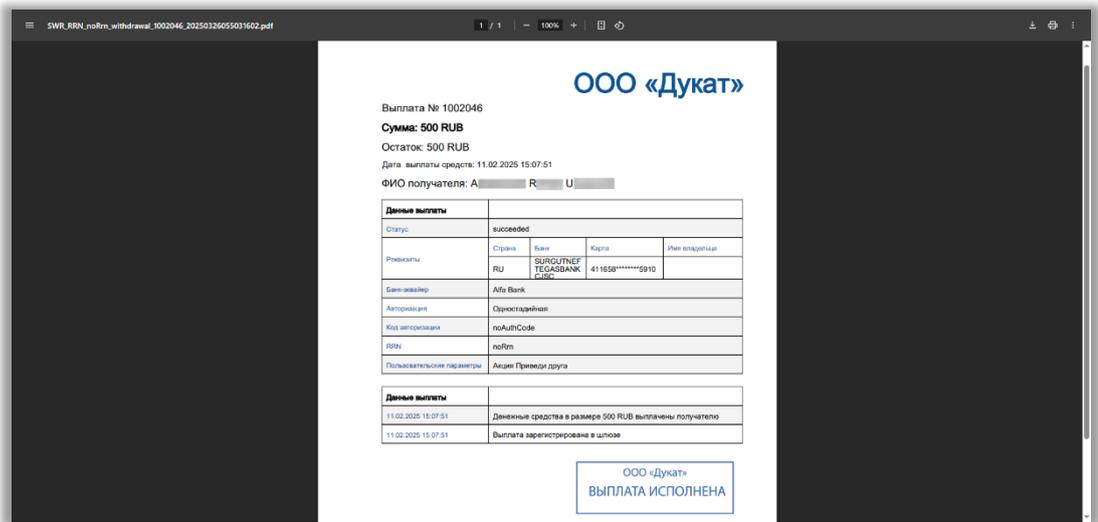


Рисунок 56 — Пример платежного поручения для выплаты в формате *.pdf

5.2.4 Раздел «Интеграция»

Раздел предназначен для создания и настройки Webhook-уведомлений по событиям выплат через ЛК. Уведомлениями также можно управлять с помощью [запросов к API](#).

Создание Webhook-уведомления по событиям выплат

1. Нажмите кнопку «Установить webhook» (рис. 57).

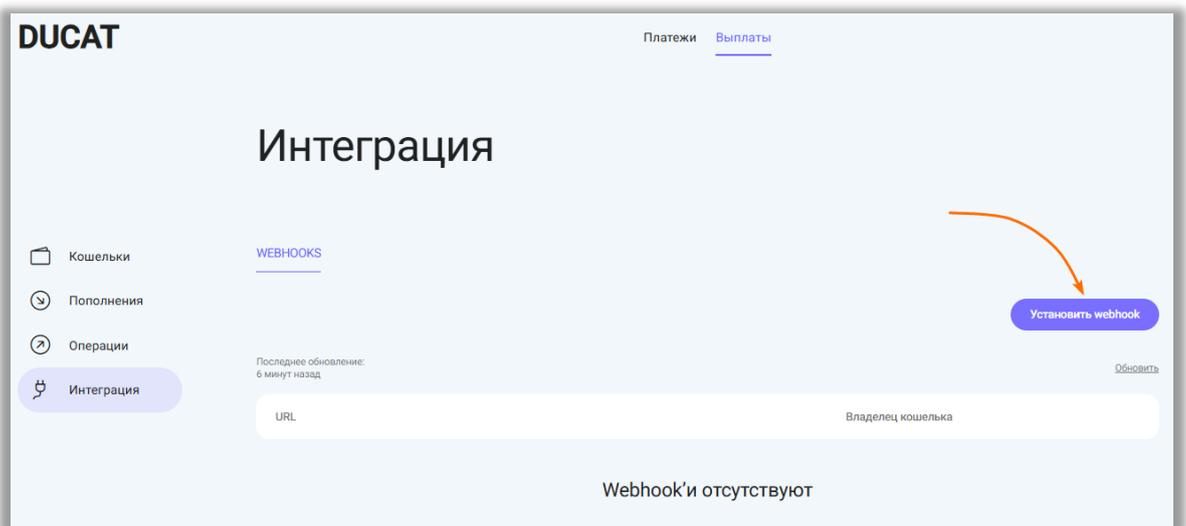


Рисунок 57 — Кнопка «Установить webhook» для событий выплат

2. Укажите Владельца кошелька (ID организации) и URL-адрес, на который необходимо отправлять уведомления (рис. 58).
3. Выберите предмет оповещений: «События выплаты» или «События приемника средств».
4. Для «События выплаты» можно указать по какому виртуальному кошельку настроить уведомления.
5. Выберите одно или несколько событий из списка, о которых необходимо оповещать.
6. Нажмите кнопку «Установить».

Установить новый webhook

Владелец кошелька *
1000009

URL, на который будут поступать оповещения о произошедших событиях *
https://test_url.ru

Предмет оповещений

События выплаты События приемника средств

Кошелек (не обязательно)
1000010 - Ducat1.DEV.VISA cards.Test SNGB

События
Набор типов событий, о которых следует оповещать

WithdrawalStarted (Выплата создана)
 WithdrawalSucceeded (Выплата совершена успешно)
 WithdrawalFailed (Выплата завершилась неудачей)

Отмена

Рисунок 58 — Настройка параметров Webhook-уведомления по событиям выплат

При нажатии на созданное Webhook-уведомление откроются данные по нему (рис. 59):

- URL-адрес, на который поступают уведомления.
- Владелец кошелька (ID организации).
- Название кошелька.
- Набор событий, по которым приходят уведомления.
- Публичный ключ — нужен для проверки подлинности приходящих уведомлений на указанный URL-адрес. Чтобы скопировать, нажмите «Скопировать ключ».

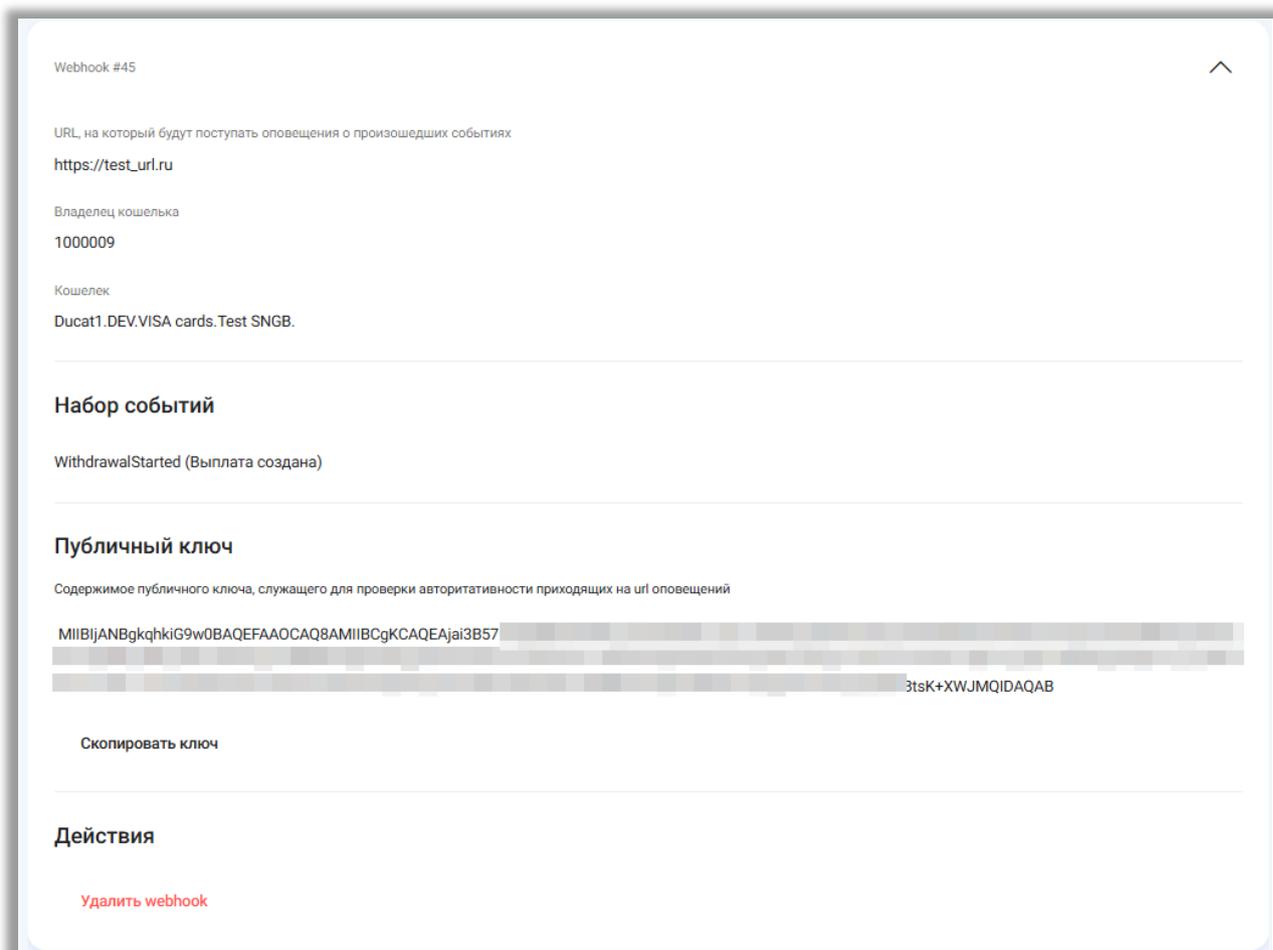


Рисунок 59 — Информация о Webhook-уведомлении по событиям выплат

Чтобы удалить Webhook-уведомление, нажмите «Удалить webhook» в блоке «Действия».

⚠ Изменить параметры созданного Webhook-уведомления нельзя. Рекомендуем удалить Webhook, который нужно изменить, и создать новый.

6 Спецификация API

API DucatPay включает в себя группы методов «Платежи» и «Выплаты». В руководстве приведено описание ключевых методов, необходимых для начала работы с DucatPay.

Полная API-документация для групп методов:

- Платежи — <https://doc.api.ducat.pro/>.
- Выплаты — <https://doc.wallet-api.ducat.pro/>.

Взаимодействие с DucatPay строится по схеме запрос-ответ. Запрос с необходимыми параметрами формируется на стороне Мерчанта и передается методом GET или POST по протоколу HTTPS.

Результаты обработки запросов возвращаются DucatPay в формате json-объекта.

 Перед началом работы с API получите [API-ключ организации](#).

Заголовки запросов (Headers)

В заголовках запросов указываются общие для всех групп параметры из таблицы 21.

Таблица 21 — Параметры заголовка запроса

Параметр	Описание
<code>api_key</code>	API-ключ организации
<code>X-Request-ID</code>	Уникальный идентификатор запроса к API, например, слепок времени. <code>X-Request-ID: 37d735d4-0f42-4f05-89fa-eea478fb5aa9</code>
<code>Content-Type</code>	Запрос к API должен выполняться в кодировке UTF-8 и с указанием содержимого в формате JSON. <code>Content-Type: application/json; charset=utf-8</code>
<code>X-Request-Deadline</code>	Максимальное время ожидания завершения операции по запросу. По истечении указанного времени прекращается обработка запроса. Может задаваться в: <ul style="list-style-type: none">• формате <code>date-time</code>;• относительных величинах: в миллисекундах (150000ms), секундах (540s) или минутах (3.5m). Рекомендуем указывать значение не менее 3 секунд и не более 1 минуты

Важно

Значения отметок времени принимаются и возвращаются в формате date-time, описанном в [RFC 3339](#):

```
2017-01-01T00:00:00Z
2017-01-01T00:00:01+00:00
```

Статусы ответов

В таблице 22 приведены HTTP-статусы, которые могут возвращаться в ответ на запрос.

Таблица 22 — Описание статусов ответов на запрос

HTTP Status Code	Статус	Описание
200 — 202	Success	Запрос успешно обработан
400	Bad Request	Неверные данные в запросе
401	Unauthorized	Ошибка авторизации
404	Not Found	Заданный ресурс/сущность не найдены
409	Conflict	Переданное значение externalID уже использовалось ранее с другими параметрами запроса
422	Unprocessable Entity Error	Неверные данные для осуществления выплаты

Формат представления HTTP Status Code **400** для группы методов «Платежи»:

```
{
  "code": "string", //Тип ошибки
  "message": "string" //Описание ошибки
}
```

Формат представления HTTP Status Code **400** для группы методов «Выплаты»:

```
{
  "errorType": "string", //Тип ошибки
  "name": "string", //Имя или ID элемента, содержащего недопустимые данные
  "description": "string" //Описание ошибки
}
```

Формат представления HTTP Status Code **404**:

```
}
  "message": "string" //Описание ошибки
}
```

Формат представления HTTP Status Code 409:

```
{
  "externalID": "string", //Значение, для которого обнаружен конфликт
  "id": "string", //ID сущности, с таким же externalID
  "message": "string" //Описание ошибки
}
```

Формат представления HTTP Status Code 422:

```
}
  "message": "string" //Описание ошибки
}
```

Ошибки при обработке запросов

В процессе обработки запросов могут происходить различные непредвиденные ситуации. Об их появлении сервис сообщает HTTP-статусами из таблицы 23.

Таблица 23 — Статусы ошибок и рекомендации по решению

HTTP Status Code	Описание	Решение
500	В процессе обработки запроса возникла непредвиденная ситуация	Обратитесь в техническую поддержку
503	Сервис временно недоступен. Запрос не был выполнен	Отправьте запрос повторно, когда доступность системы будет восстановлена
504	Превышено допустимое время обработки запроса. Результат запроса не определен	Отправьте запрос повторно. Если повторная отправка запроса нежелательна, уточните результат выполнения исходного запроса

6.1 Группа «Платежи»

URL запроса: `https://{domain}/v2/processing/{method-path}`

- `{domain}` — адрес обращения к ресурсу.
- `{method-path}` — путь доступа по API для выполнения запроса.

Адреса для обращения к ресурсам:

- Тестовый контур — <https://api.dev.ducat.pro/>.
- Промышленный контур — <https://api.ducatpay.com/>.

Методы

- [createInvoice](#) — создание инвойса.
- [createPaymentResource](#) — создание одноразового токена платежного средства и платежной сессии.
- [createPayment](#) — создание платежа по указанному инвойсу.
- [createRefund](#) — возврат платежа.
- [getPaymentByID](#) — получение данных платежа по указанному инвойсу.

Статусы инвойса приведены в таблице 24, статусы платежа отражены в таблице 25, возврата — в таблице 26.

Таблица 24 — Статусы инвойса

Состояние	Значение	Описание
Не оплачен	<code>unpaid</code>	Инвойс создан, но не оплачен
Отменен	<code>cancelled</code>	Инвойс отменен с указанием причины
Оплачен	<code>paid</code>	Платеж по инвойсу поступил, но товары или услуги Покупателю еще не предоставлены
Погашен	<code>fulfilled</code>	Инвойс погашен, все обязательства, как Покупателя, так и Мерчанта завершены

Таблица 25 — Статусы платежа

Состояние	Значение	Описание
Запущен	pending	Платеж создан, но еще не завершен
Обработан	processed	Получен ответ об успешном холдировании средств банком-эмитентом на счету Покупателя
Подтвержден	captured	Денежные средства были удержаны с карты Покупателя при двухстадийном платеже
Отменен	cancelled	Денежные средства вернулись на счет Покупателя при двухстадийном платеже
Возвращен	refunded	Платеж был возвращен Покупателю
Неуспешен	failed	Платеж завершился с ошибкой

Таблица 26 — Статусы возврата

Состояние	Значение	Описание
Запущен	pending	Возврат создан
Обработан	succeeded	Платеж возвращен Покупателю
Неуспешен	failed	Возврат завершился с ошибкой

createInvoice

Создание нового инвойса.

Метод

POST /v2/processing/invoices

Параметры тела запроса

* — обязательные параметры.

Параметр	Тип	Описание
shopID*	string [1..40] characters	Идентификатор магазина
partyID	string [1..40] characters	Уникальный идентификатор организации
externalID	string [1..40] characters	Внешний идентификатор запроса для внутреннего операционного учета Мерчанта
dueDate*	string <date-time>	Дата и время окончания действия инвойса. Оплата после наступления указанной даты невозможна
amount	integer <int64> ≥ 1	Стоимость предлагаемых товаров или услуг. Если стоимость не указана, то стоимость рассчитывается по сумме позиций в корзине
currency*	string ^{[A-Z]{3}}	Валюта, символьный код согласно ISO 4217. ENUM: RUB
product*	string ≤100 characters	Наименование предлагаемых товаров или услуг
description	string ≤ 1000 characters	Описание предлагаемых товаров или услуг
cart	InvoiceCart	Корзина с набором позиций товаров или услуг
bankAccount	InvoiceBankAccount	Информация о банковском счете плательщика, к операциям с которым относится данный инвойс
metadata*	object	Метаданные, которые необходимо связать с инвойсом
clientInfo	object	Дополнительная информация о клиенте
trustLevel*	string	Надежность плательщика ENUM: wellKnown, unknown

InvoiceCart

Параметр	Тип	Описание
product*	string ≤1000 characters	Описание предлагаемого товара или услуги
quantity*	integer <int64> ≥ 1	Количество товаров или услуг по позиции
price*	integer <int64> ≥ 1	Цена за товар или услугу
taxMode	object	Схема налогообложения товара или услуги. Указывается, только если предлагаемый товар или услуга облагается налогом
type*	string	Тип схемы налогообложения. ENUM: InvoiceLineTaxVAT
rate*	string	Ставка налога

InvoiceBankAccount

Параметр	Тип	Описание
accountType*	string	Вид банковского счета. ENUM: InvoiceRussianBankAccount
account*	string ^\d{20}\$	Номер счета
bankBik*	string ^\d{9}\$	БИК банковской организации

Пример запроса

```
{
  "shopID": "string",
  "partyID": "string",
  "externalID": "string",
  "dueDate": "2019-08-24T14:15:22Z",
  "amount": 1,
  "currency": "RUB",
  "product": "string",
  "description": "string",
  "cart": [{
    "product": "string",
    "quantity": 1,
    "price": 1,
    "taxMode": {
      "type": "InvoiceLineTaxVAT",
      "rate": "0%"
    }
  }],
  "bankAccount": {
    "accountType": "InvoiceRussianBankAccount",
    "account": "string",
    "bankBik": "000000000",
  },
  "metadata": {},
  "clientInfo": {
    "trustLevel": "wellKnown"
  }
}
```

Возвращаемое значение HTTP Status Code

- [201](#) — инвойс создан.
- 400, 401, 409 — [ошибка при выполнении запроса](#).

HTTP Status Code 201

Параметр	Тип	Описание
invoice*	object	Параметры инвойса
id*	string	Идентификатор инвойса
shopID*	string [1..40] characters	Идентификатор магазина
partyID*	string [1..40] characters	Уникальный идентификатор организации
externalID	string [1..40] characters	Внешний идентификатор запроса для внутреннего операционного учета Мерчанта
createdAt*	string <date-time>	Дата и время создания
dueDate*	string <date-time>	Дата и время окончания действия инвойса. Оплата после наступления указанной даты невозможна
amount*	integer <int64> ≥ 1	Стоимость предлагаемых товаров или услуг. Если стоимость не указана, то стоимость рассчитывается по сумме позиций в корзине
currency*	string ^{[A-Z]{3}}	Валюта, символьный код согласно ISO 4217. ENUM: RUB
product*	string ≤ 100 characters	Наименование предлагаемых товаров или услуг
description	string ≤ 1000 characters	Описание предлагаемых товаров или услуг
cart	InvoiceCart	Корзина с набором позиций товаров или услуг
bankAccount	InvoiceBankAccount	Информация о банковском счете плательщика, к операциям с которым относится данный инвойс
metadata*	object	Метаданные, которые необходимо связать с инвойсом
clientInfo	object	Дополнительная информация о клиенте
trustLevel*	string	Надежность Покупателя. ENUM: wellKnown, unknown
status*	string	Статус инвойса
invoiceAccessToken*	object	Токен доступа к инвойсу
payload*	string	Содержимое токена доступа

Пример ответа

```
{
  "invoice": {
    "id": "1000000000",
    "shopID": "string",
    "externalID": "string",
    "createdAt": "2025-01-24T14:15:22Z",
    "dueDate": "2025-01-24T14:15:22Z",
    "amount": 1,
    "currency": "RUB",
    "product": "string",
    "description": "string",
    "cart": [{
      "product": "string",
      "quantity": 1,
      "price": 1,
      "cost": 1,
      "taxMode": {
        "type": "InvoiceLineTaxVAT"
      }
    }],
    "bankAccount": {
      "accountType": "InvoiceRussianBankAccount",
      "account": "string",
      "bankBik": "000000000",
    },
    "metadata": {},
    "clientInfo": {
      "trustLevel": "wellKnown"
    },
    "status": "unpaid",
  },
  "invoiceAccessToken": {
    "payload": "string"
  }
}
```

createPaymentResource

Создание одноразового токена платежного средства и уникальной платежной сессии.

⚠ Токен платежного средства и идентификатор сессии имеют ограниченный срок жизни — 1 час.

Метод

POST /v2/processing/payment-resources

Параметры тела запроса

* — обязательные параметры.

Параметр	Тип	Описание
paymentTool*	object	Данные платежного средства
paymentToolType*	string	Тип платежного средства. ENUM: CardData
cardNumber*	string [^] \d{12,19}\$	Номер банковской карты
expDate*	string [^] \d{2}/(\d{2}\d{4})\$	Срок действия банковской карты
cvv	string [^] \d{3,4}\$	Код верификации
cardHolder	string [1..100] characters [^] [[:alpha:][:space:][:punct:]]+\$	Имя держателя карты
clientInfo*	clientInfo	Данные устройства Покупателя

clientInfo

Параметр	Тип	Описание
fingerprint*	string ≤ 1000 characters	ID платежного средства
ip	string <ip-address> ≤ 45 characters	IP-адрес Покупателя
url	string <uri> ≤ 1000 characters	URL платежной формы

Пример запроса

```
{
  "paymentTool": {
    "paymentToolType": "CardData",
    "cardNumber": "2201382000000021",
    "expDate": "12/34",
    "cvv": "123",
    "cardHolder": "Ivanov Ivan"
  },
  "clientInfo": {
    "fingerprint": "000111222",
    "ip": "164.0.0.1",
    "url": "http://example.ru"
  }
}
```

Возвращаемое значение HTTP Status Code

- [201](#) — токен и сессия созданы.
- 400, 401 — [ошибка при выполнении запроса](#).

HTTP Status Code 201

Параметр	Тип	Описание
paymentToolToken*	object	Токен платежного средства
paymentSession*	string ≤ 2000 characters	Идентификатор платежной сессии
paymentToolDetails	paymentToolDetails	Детали платежного средства
clientInfo*	clientInfo	Данные устройства Покупателя
validUntil	string <date-time>	Срок действия токена платежного средства

paymentToolDetails

Параметр	Тип	Описание
detailsType*	string	Тип информации о платежном средстве. ENUM: PaymentToolDetailsBankCard
cardHolder	string [1..100] characters ^[[:alpha:][:space:][:punct:]]+\$	Имя держателя карты
cardNumberMask*	string ^\d{0,6}*\d{0,4}\$	Маскированный номер банковской карты
expDate	string ^\d{2}/(\d{2}\d{4})\$	Срок действия банковской карты
first6	string ^\d{6}\$	Первые цифры номера карты
last4	string ^\d{0,4}\$	Последние цифры номера карты
panHash	string	Зашифрованный в формате SHA-256 полный номер карты Покупателя
paymentSystem*	string	Платежная система

Пример ответа

```
{
  "paymentToolToken": "v2...7KDW71PFAziaAGkpLMutQ",
  "paymentSession": "eyJjbGllbnRJbmZvI...iNGclcUh6NmhTaUZ0aWJBZzZFY",
  "paymentToolDetails": {
    "detailsType": "PaymentToolDetailsBankCard",
    "cardNumberMask": "220138*****0021",
    "first6": "220138",
    "last4": "0021",
    "panHash": "01224AC5B682802F3B07133...0DF4BBA33B3DD212583",
    "paymentSystem": "NSPK MIR"
  },
  "clientInfo": {
    "fingerprint": "000111222",
    "ip": "164.0.0.1",
    "url": "http://example.ru"
  }
  "validUntil": "2025-03-14T08:41:46.074Z"
}
```

createPayment

Создание платежа по указанному инвойсу.

Метод

POST /v2/processing/invoices/{invoiceID}/payments

Path-параметры

- invoiceID* — идентификатор инвойса.

* — обязательные параметры.

Параметры тела запроса

Параметр	Тип	Описание
externalID*	string [1..40] characters	Внешний идентификатор запроса для внутреннего операционного учета Мерчанта
flow*	object	Тип платежа и порядок удержания денежных средств
type*	string	Тип платежа: <ul style="list-style-type: none">• PaymentFlowInstant — одностадийный;• PaymentFlowHold — двухстадийный. Default: PaymentFlowInstant
onHoldExpiration*	string	Действие с удержанными средствами для двухстадийного платежа: <ul style="list-style-type: none">• cancel — вернуться Покупателю;• capture — поступят Мерчанту. Default: cancel
payer*	Payer request	Информация о платежных данных
processingDeadline	string [1..40] characters	Максимальное время обработки платежа
makeRecurrent	boolean	Признак создания родительского рекуррентного платежа: <ul style="list-style-type: none">• false — признак выключен;• true — признак включен. Default: false
metadata	object	Метаданные, которые необходимо связать с платежом

Payer request

Параметр	Тип	Описание
payerType*	string	Тип платежного средства: <ul style="list-style-type: none"> PaymentResourcePayer — платеж по банковской карте; RecurrentPayer — рекуррентный платеж
sessionInfo	string ^{^d{20}\$}	Данные текущей сессии
redirectUrl	string <uri-template> ≤ 2000 characters	URL-адрес куда будет перенаправлен Покупатель
paymentToolToken	string ≤ 2000 characters	Токен платежного средства. Обязательное поле при выборе типа платежного средства PaymentResourcePayer
paymentSession	string ≤ 1000 characters	Идентификатор платежной сессии. Обязательное поле при выборе типа платежного средства PaymentResourcePayer
contactInfo*	object	Контактные данные
email	string <email> ≤ 100 characters	Адрес электронной почты
phoneNumber	string <^\+d{4,15}\$>	Номер мобильного телефона с международным префиксом согласно E.164
recurrentParentPayment	object	Данные родительского платежа. Обязательное поле при выборе типа платежного средства RecurrentPayer
invoiceID*	string	Идентификатор инвойса
paymentID*	string	Идентификатор платежа

Пример запроса

```
{
  "externalID": "string",
  "flow": {
    "type": "PaymentFlowInstant"
  },
  "payer": {
    "payerType": "PaymentResourcePayer",
    "paymentToolToken": "eyJjbGllbnRJbmRT...1sTmtJT2t4dDBsWCJ9",
    "paymentSession": "v2.I6jGKoWUF3VP60iM_MVNmM...6mt_d9rjbJw",
    "sessionInfo": {
      "redirectUrl": "https://google.com"
    },
    "contactInfo": {
      "email": "example@mail.ru",
      "phoneNumber": "9991234567"
    }
  },
  "processingDeadline": "30m",
  "makeRecurrent": false,
  "metadata": {}
}
```

Возвращаемое значение HTTP Status Code

- [201](#) — платеж создан.
- 400, 401, 404, 409 — [ошибка при выполнении запроса](#).

HTTP Status Code 201

Параметр	Тип	Описание
id	string	Идентификатор платежа
externalID	string [1..40] characters	Внешний идентификатор запроса для внутреннего операционного учета Мерчанта
invoiceID	string	Идентификатор инвойса
createdAt	string <date-time>	Дата и время создания
amount	integer <int64> ≥ 1	Стоимость предлагаемых товаров или услуг. Если стоимость не указана, то стоимость рассчитывается по сумме позиций в корзине
currency	string ^{[A-Z]{3}} \$	Валюта, символьный код согласно ISO 4217. ENUM: RUB
flow	object	Тип платежа и порядок удержания денежных средств
type	string	Тип платежа: <ul style="list-style-type: none"> PaymentFlowInstant — одностадийный; PaymentFlowHold — двухстадийный. Default: PaymentFlowInstant
onHoldExpiration	string	Действие с удержанными средствами для двухстадийного платежа: <ul style="list-style-type: none"> cancel — вернуться Покупателю; capture — поступят Мерчанту. Default: cancel
heldUntil	string <date-time>	Дата и время, до которого происходит удержание денежных средств
payer	Payer response	Информация о платежных данных
transactionInfo	object	Информация о транзакции
rrn	string ^{[a-zA-Z0-9]{4,20}} \$	Идентификатор банковской транзакции
approvalCode	string	Код авторизации
makeRecurrent	boolean	Признак создания родительского рекуррентного платежа: <ul style="list-style-type: none"> false — признак выключен; true — признак включен. Default: false

metadata	object	Метаданные, которые необходимо связать с платежом
status	string	Статус платежа
error	object	Ошибка, возникшая в процессе
code	string	Код ошибки
subError	object	Описание ошибки

Payer response

Параметр	Тип	Описание
payerType*	string	Тип платежного средства: <ul style="list-style-type: none"> PaymentResourcePayer — платеж по банковской карте; RecurrentPayer — рекуррентный платеж
sessionInfo	string [^] \d{20}\$	Данные текущей сессии
redirectUrl	string <uri-template> ≤ 2000 characters	URL-адрес куда будет перенаправлен Покупатель
paymentToolToken	string ≤ 2000 characters	Токен платежного средства. Обязательное поле при выборе типа платежного средства PaymentResourcePayer
paymentSession	string ≤ 1000 characters	Идентификатор платежной сессии. Обязательное поле при выборе типа платежного средства PaymentResourcePayer
contactInfo*	object	Контактные данные
email	string <email> ≤ 100 characters	Адрес электронной почты
phoneNumber	string <^+\d{4,15}\$>	Номер мобильного телефона с международным префиксом согласно E.164
recurrentParentPayment	object	Данные родительского платежа. Обязательное поле при выборе типа платежного средства RecurrentPayer
invoiceID*	string	Идентификатор инвойса
paymentID*	string	Идентификатор платежа
paymentToolDetails	paymentToolDetails	Детали платежного средства
clientInfo	clientInfo	Данные устройства Покупателя

Пример ответа

```
{
  "id": "1000000",
  "externalID": "123U4ghrm",
  "invoiceID": "9254CUbmJzuq",
  "createdAt": "2025-03-14T08:36:33.369127Z",
  "amount": 15300,
  "currency": "RUB",
  "flow": {
    "type": "PaymentFlowInstant"
  },
  "payer": {
    "clientInfo": {
      "browserInfo": "undefined",
      "fingerprint": "000111222",
      "ip": "10.1.0.17"
    },
    "contactInfo": {
      "email": "test@test.ru",
      "phoneNumber": "9991234567"
    },
    "payerType": "PaymentResourcePayer",
    "paymentToolDetails": {
      "cardHolder": "Ivanov Ivan",
      "cardNumberMask": "220138*****0021",
      "detailsType": "PaymentToolDetailsBankCard",
      "expDate": "12/2034",
      "first6": "220138",
      "last4": "0021",
      "panHash": "01224AC5B682802F3B07133...0DF4BBA33B3DD212583",
      "paymentSystem": "NSPK MIR"
    },
    "paymentToolToken": "v2...XcdF9F1xYCaeLRcchcjFjg",
    "sessionInfo": {
      "redirectUrl": "https://google.com"
    }
  },
  "makeRecurrent": false,
  "metadata": {},
  "status": "pending"
}
```

createRefund

Возврат платежа.

Метод

POST /v2/processing/invoices/{invoiceID}/payments/{paymentID}/refunds

Path-параметры

- invoiceID* — идентификатор инвойса.
- paymentID* — идентификатор платежа по инвойсу.

* — обязательные параметры.

Параметры тела запроса

Параметр	Тип	Описание
externalID	string [1..40] characters	Внешний идентификатор запроса для внутреннего операционного учета Мерчанта
amount*	integer <int64> ≥ 1	Сумма возврата (полная или частичная)
currency*	string^[A-Z]{3}\$	Валюта, символьный код согласно ISO 4217. ENUM: RUB
reason*	string ≤ 1000 characters	Причина совершения возврата

Пример запроса

```
{
  "externalID": "string",
  "amount": 1,
  "currency": "RUB",
  "reason": "string",
}
```

Возвращаемое значение HTTP Status Code

- 201 — возврат создан.
- 400, 401, 404, 409 — [ошибка при выполнении запроса](#).

HTTP Status Code 201

Параметр	Тип	Описание
id*	string	Идентификатор возврата
externalID	string [1..40] characters	Внешний идентификатор запроса для внутреннего операционного учета Мерчанта
createdAt*	string <date-time>	Дата и время осуществления
amount*	integer <int64> ≥ 1	Сумма возврата (полная или частичная)
currency*	string^[A-Z]{3}\$	Валюта, символьный код согласно ISO 4217. ENUM: RUB
reason	string	Причина совершения возврата
status*	string	Статус возврата
error	object	Данные ошибки, в случае если возврат был неуспешен
code*	string	Код ошибки
message*	string	Описание ошибки

Пример ответа

```
{
  "id": "string",
  "externalID": "string",
  "createdAt": "2019-08-24T14:15:22Z",
  "amount": 1,
  "currency": "RUB",
  "reason": "string",
  "status": "pending",
  "error": {
    "code": "string",
    "message": "string"
  }
}
```

getPaymentByID

Получение данных платежа по указанному инвойсу.

Метод

GET	/v2/processing/invoices/{invoiceID}/payments/{paymentID}
------------	--

Path-параметры

- invoiceID* — идентификатор инвойса.
- paymentID* — идентификатор платежа по инвойсу.

* — обязательные параметры.

Возвращаемое значение HTTP Status Code

- 200 — платеж найден, данные переданы в [json-объекте](#).
- 400, 401, 404 — [ошибка при выполнении запроса](#).

6.2 Группа «Выплаты»

URL запроса: `https://{domain}/wallet/v0/{method-path}`

- {domain} — адрес обращения к ресурсу.
- {method-path} — путь доступа по API для выполнения запроса.

Адреса для обращения к ресурсам:

- Тестовый контур — <https://api.dev.ducat.pro/> .
- Промышленный контур — <https://api.ducatspay.com/> .

Методы

- [createIdentitie](#) — создание цифровой личности.
- [createWallet](#) — создание виртуального кошелька.
- [createDestination](#) — создание получателя выплаты.
- [createWithdrawal](#) — создание выплаты.
- [getWithdrawalStatus](#) — получение статуса выплаты.

createIdentities

Создание цифровой личности для проведения выплат.

Метод

POST /wallet/v0/identities

Параметры тела запроса

* — обязательные параметры.

Параметр	Тип	Описание
name*	string	Наименование юридического лица
provider*	string	Название банка-эквайера
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации
partyID	string [1..40] characters	Идентификатор организации

Пример запроса

```
{
  "name": "Keyn Fawkes",
  "provider": "serviceprovider",
  "metadata": {
    "lkDisplayName": "Иван Иванович"
  },
  "externalID": "10036274",
  "partyID": "string"
}
```

Возвращаемое значение HTTP Status Code

- 201 — цифровая личность создана.
- 400, 401, 409, 422 — [ошибка при выполнении запроса](#).

HTTP Status Code 201

Заголовок ответа

Location — URI созданной личности (string <uri>).

Тело ответа

Параметр	Тип	Описание
id	string	Идентификатор цифровой личности
name*	string	Наименование юридического лица
createdAt	string <date-time>	Дата и время создания личности
provider*	string	Название банка-эквайера
isBlocked	boolean	Признак блокировки личности владельца
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации
partyID	string [1..40] characters	Идентификатор организации

Пример ответа

```
{
  "id": "10036274",
  "name": "Keyn Fawkes",
  "createdAt": "2019-08-24T14:15:22Z",
  "provider": "serviceprovider",
  "isBlocked": false,
  "metadata": {
    "lkDisplayName": "Иван Иванович"
  },
  "externalID": "10036274",
  "partyID": "string"
}
```

createWallet

Создание виртуального кошелька для цифровой личности.

Метод

POST /wallet/v0/wallets

Параметры тела запроса

* — обязательные параметры.

Параметр	Тип	Описание
name*	string	Название виртуального кошелька
identity*	string	Идентификатор личности владельца кошелька
currency*	string ^{[A-Z]{3}}	Валюта, символьный код согласно ISO 4217. ENUM: RUB
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации

Пример запроса

```
{
  "name": "Worldwide PHP Awareness Initiative",
  "identity": "10036274",
  "currency": "RUB",
  "metadata": {
    "client_locale": "RU_ru"
  },
  "externalID": "10036274"
}
```

Возвращаемое значение HTTP Status Code

- 201 — виртуальный кошелек создан.
- 400, 401, 409, 422 — [ошибка при выполнении запроса](#).

HTTP Status Code 201

Заголовок ответа

Location — URI созданного кошелька (string <uri>).

Тело ответа

Параметр	Тип	Описание
id	string	Идентификатор виртуального кошелька
name*	string	Название кошелька
createdAt	string <date-time>	Дата и время создания кошелька
isBlocked	boolean	Признак блокировки кошелька
identity*	string	Идентификатор личности владельца кошелька
currency*	string ^{[A-Z]{3}} \$	Валюта, символьный код согласно ISO 4217. ENUM: RUB
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации

Пример ответа

```
{
  "id": "10068321",
  "name": "Worldwide PHP Awareness Initiative",
  "createdAt": "2019-08-24T14:15:22Z",
  "isBlocked": false,
  "identity": "10036274",
  "currency": "RUB",
  "metadata": {
    "client_locale": "RU_ru"
  },
  "externalID": "10036274"
}
```

createDestination

Создание получателя выплаты.

Метод

POST /wallet/v0/destinations

Параметры тела запроса

* — обязательные параметры.

Параметр	Тип	Описание
name*	string	Наименование получателя выплаты
identity*	string	Идентификатор личности владельца кошелька
currency*	string ^{[A-Z]{3}}	Валюта, символьный код согласно ISO 4217. ENUM: RUB
resource*	object	Платежные данные получателя
type*	string	Способ выплаты ENUM: BankCardDestinationResource
token*	string [1..1000] characters	Токен, идентифицирующий исходные данные карты
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации

Пример запроса

```
{
  "name": "Squarey plastic thingy",
  "identity": "10036274",
  "currency": "RUB",
  "resource": {
    "type": "BankCardDestinationResource",
    "token": "v2...XcdF9F1xYCaeLRcchcjFjg"
  },
  "metadata": {
    "color_hint": "olive-green"
  },
  "externalID": "10036274"
}
```

Возвращаемое значение HTTP Status Code

- 201 — получатель выплаты создан.
- 400, 401, 409, 422 — [ошибка при выполнении запроса](#).

HTTP Status Code 201

Заголовок ответа

Location — URI созданного получателя выплаты (string <uri>).

Тело ответа

Параметр	Тип	Описание
id	string	Идентификатор получателя выплаты
name*	string	Наименование получателя выплаты
createdAt	string <date-time>	Дата и время создания получателя
isBlocked	boolean	Признак блокировки получателя
identity*	string	Идентификатор личности владельца кошелька
currency*	string ^{[A-Z]{3}}	Валюта, символьный код согласно ISO 4217. ENUM: RUB
resource*	object	Платежные данные получателя
type*	string	Способ выплаты. ENUM: BankCardDestinationResource
token*	string [1..1000] characters	Токен, идентифицирующий исходные данные карты
bin	string ^{\d{6,8}}	Идентификационный номер карты банка-эмитента
lastDigits	string ^{\d{2,4}}	Последние цифры номера карты
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации
status	string	Статус получателя: <ul style="list-style-type: none">• Unauthorized — не авторизован владельцем на вывод средств;• Authorized — авторизован владельцем на вывод средств
validUntil	string <date-time>	Дата и время, до наступления которых авторизация действительна. Если status == Authorized

Пример ответа

```
{
  "id": "107498",
  "name": "Squarey plastic thingy",
  "createdAt": "2019-08-24T14:15:22Z",
  "isBlocked": false,
  "identity": "10036274",
  "currency": "RUB",
  "resource": {
    "type": "BankCardDestinationResource",
    "token": "v2...XcdF9F1xYCaeLRcchcjFjg",
    "bin": "200000",
    "lastDigits": "0021"
  },
  "metadata": {
    "color_hint": "olive-green"
  },
  "externalID": "10036274",
  "status": "Authorized",
  "validUntil": "2019-08-24T14:15:22Z"
}
```

createWithdrawal

Создание выплаты.

Метод

POST /wallet/v0/withdrawals

Параметры тела запроса

* — обязательные параметры.

Параметр	Тип	Описание
wallet*	string	Идентификатор кошелька
destination*	string	Идентификатор получателя выплаты
body*	object	Параметры выплаты
amount*	integer <int64>	Сумма выплаты
currency*	string ^{[A-Z]{3}} \$	Валюта выплаты, символьный код согласно ISO 4217. ENUM: RUB
fee	object	Параметры комиссии
amount*	integer <int64>	Сумма комиссии
currency*	string ^{[A-Z]{3}} \$	Валюта комиссии, символьный код согласно ISO 4217. ENUM: RUB
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации
resource	object	Платежные данные получателя
type*	string	Способ выплаты. ENUM: BankCardDestinationResource
token*	string [1..1000] characters	Токен, идентифицирующий исходные данные карты

Пример запроса

```
{
  "wallet": "10068321",
  "destination": "107498",
  "body": {
    "amount": 1430000,
    "currency": "RUB"
  },
  "fee": {
    "amount": 1430000,
    "currency": "RUB"
  },
  "metadata": {
    "notify_email": "iliketrains@example.com"
  },
  "externalID": "10036274",
  "resource": {
    "type": "BankCardDestinationResource",
    "token": "v2...XcdF9F1xYCaELRcchcjFjg"
  }
}
```

Возвращаемое значение HTTP Status Code

- 202 — выплата создана.
- 400, 401, 409, 422 — [ошибка при выполнении запроса](#).

HTTP Status Code 202

Заголовок ответа

Location — URI созданной выплаты (string <uri>).

Тело ответа

Параметр	Тип	Описание
id	string	Идентификатор выплаты
createdAt	string <date-time>	Дата и время создания выплаты
wallet*	string	Идентификатор кошелька
destination*	string	Идентификатор получателя выплаты
body*	object	Параметры выплаты
amount*	integer <int64>	Сумма выплаты
currency*	string ^{[A-Z]{3}}	Валюта выплаты, символьный код согласно ISO 4217. ENUM: RUB
fee	object	Параметры комиссии
amount*	integer <int64>	Сумма комиссии
currency*	string ^{[A-Z]{3}}	Валюта комиссии, символьный код согласно ISO 4217. ENUM: RUB
rrn	string	Идентификатор банковской транзакции
metadata	object	Произвольный набор дополнительных параметров, которые можно задать
externalID	string	Внешний идентификатор запроса для внутреннего операционного учета организации
resource	object	Платежные данные получателя
type*	string	Способ выплаты. ENUM: BankCardDestinationResource
token*	string [1..1000] characters	Токен, идентифицирующий исходные данные карты
bin	string ^{\d{6,8}}	Идентификационный номер карты банка-эмитента
lastDigits	string ^{\d{2,4}}	Последние цифры номера карты
status	string	Статус выплаты: <ul style="list-style-type: none">• Pending — выплата в процессе выполнения;• Succeeded — выплата прошла успешно;• Failed — выплата завершилась неудачей
failure	object	Причина неудачной выплаты. Если status == Failed
code	string	Код ошибки
subError	object	Описание ошибки

Пример ответа

```
{
  "id": "тZ0jUmlsv0",
  "createdAt": "2019-08-24T14:15:22Z",
  "wallet": "10068321",
  "destination": "107498",
  "body": {
    "amount": 1430000,
    "currency": "RUB"
  },
  "fee": {
    "amount": 1430000,
    "currency": "RUB"
  },
  "rrn": "string",
  "metadata": {
    "notify_email": "iliketrains@example.com"
  },
  "externalID": "10036274",
  "resource": {
    "type": "BankCardDestinationResource",
    "token": "v2...XcdF9F1xYCaeLRcchcjFjg",
    "bin": "200000",
    "lastDigits": "0021"
  },
  "status": "Pending",
  "failure": {
    "code": "string",
    "subError": {
      "code": "string",
      "subError": {}
    }
  }
}
```

getWithdrawalStatus

Получение статуса выплаты.

Метод

GET	/wallet/v0/withdrawals/{withdrawalID}
------------	---------------------------------------

Path-параметры

- `withdrawalID*` — идентификатор выплаты.

* — обязательный параметр.

Возвращаемое значение HTTP Status Code

- 200 — выплата найдена, данные переданы в [json-объекте](#).
- 400, 401, 404 — [ошибка при выполнении запроса](#).