

ПРОТОКОЛ ЕАСQ БЕЗ PCI DSS

24.04.2023



Тинькофф

Оглавление

| | |
|---|----|
| ИСТОРИЯ ИЗМЕНЕНИЙ | 4 |
| ТЕРМИНЫ И СОКРАЩЕНИЯ | 7 |
| РЕКОМЕНДАЦИИ ПРИ ИНТЕГРАЦИИ | 8 |
| 1. ПАРАМЕТРЫ ПРИЕМА ПЛАТЕЖЕЙ | 9 |
| 2. МЕТОДЫ ПРИЕМА ПЛАТЕЖЕЙ..... | 11 |
| 2.1. Общая информация | 11 |
| 2.2. Схема проведения платежа | 11 |
| 2.3. Метод Init | 13 |
| 2.4. Метод Authorize | 35 |
| 2.5. Метод FinishAuthorize..... | 35 |
| 2.6. Оплата картой на платежной форме банка | 35 |
| 2.7 Метод Confirm | 36 |
| 2.8. Метод Charge | 39 |
| 2.9. Метод Cancel | 42 |
| См параметр MemberId в Методе QrMembersList | 43 |
| 2.10. Метод GetState | 45 |
| 2.11. Метод CheckOrder | 48 |
| 2.12. Метод SendClosingReceipt | 51 |
| 2.13. Оплата картой на платежной форме банка | 53 |
| 2.14. Оплата через Tinkoff Pay Web на сайте мерчанта | 53 |
| 2.14.1. Метод TinkoffPay/terminals/{terminalKey}/status..... | 53 |
| 2.14.2. Метод TinkoffPay/transactions/{paymentId}/versions/{version}/link | 55 |
| 2.14.3. Метод GET /v2/TinkoffPay/{paymentId}/QR..... | 56 |
| 3. МЕТОДЫ ОПЛАТЫ ПО СБП..... | 58 |
| 3.1. Схема проведения платежа при оплате по QR..... | 58 |
| 3.2. Метод GetQr | 59 |
| 3.3. Метод QrMembersList | 60 |
| 3.4. Метод AddAccountQr | 62 |
| 3.5. Метод ChargeQr | 66 |
| 3.6. Метод GetAddAccountQrState | 68 |
| 3.7. Метод GetAccountQrList..... | 69 |
| 3.8. Метод GetQRState | 71 |
| 3.9 Метод SbpPayTest..... | 72 |
| 3.10. Тесты по QR | 74 |
| 4. НОТИФИКАЦИЯ ПРОДАВЦА ОБ ОПЕРАЦИЯХ | 75 |
| 4.1. Нотификации по электронной почте | 75 |
| 4.2.Нотификации по http(s) | 75 |
| 4.3. Нотификация о фискализации | 79 |
| 4.4 Нотификация о статусе привязки счета | 81 |
| 5. МЕТОДЫ РАБОТЫ С ПРИВЯЗАННЫМИ КАРТАМИ И КЛИЕНТАМИ..... | 83 |

| | |
|--|----|
| 5.1. Метод AddCustomer..... | 83 |
| 5.2. Метод GetCustomer..... | 85 |
| 5.3. Метод RemoveCustomer..... | 86 |
| 5.4. Метод GetCardList | 88 |
| 6. Коды ошибок, передаваемые на FAILURL | 90 |
| 7. Список тестовых карт | 91 |
| 8. Правила расчета возмещений по операционному реестру | 94 |

История изменений

| Версия | Описание | Дата |
|--------|---|------------|
| 1.0 | Документ создан | 10.04.2018 |
| 1.1 | Добавлено описание метода Resend. Добавлены параметры Route и Source в метод FinishAuthorize. Незначительные изменения оформления. | 12.04.2018 |
| 1.2 | Исправлено описание метода Submit3DSAuthorization | 28.04.2018 |
| 1.3 | Добавлен метод EditCustomer. Дополнено описание нотификации о фискализации. Добавлено описание статуса PARTIAL_REFUNDED. Незначительные изменения оформления. | 04.05.2018 |
| 1.4 | Удалено описание метода GetAddCardState | 08.05.2018 |
| 1.5 | Удален метод EditCustomer. Добавлен статус карты D в ответ GetCardList. Добавлен код ошибки 7 в список кодов | 21.05.2018 |
| 1.6 | Изменено описание параметра Quantity в объекте Items | 22.05.2018 |
| 1.7 | Удален PaymentUrl в ответе Init | 24.05.2018 |
| 1.8 | Добавлено описание метода GetAddCardState | 19.06.2018 |
| 1.9 | Добавлен параметр EncryptedPaymentData в метод FinishAuthorize (для проведения ApplePay и GooglePay) | 26.06.2018 |
| 1.10 | Добавлено описание нового параметра descriptor в методе Init | 04.07.2018 |
| 1.11 | Добавлены параметры Shops и Receipts в методы Init, Confirm, Cancel. Изменена схема проведения платежа | 14.02.2019 |
| 1.12 | Добавлено описание параметра ApplePayWeb на Init | 03.06.2019 |
| 1.13 | Добавлена информация об оплате по QR | 27.05.2020 |
| 1.14 | Добавлено описание опционального параметра DefaultCard в объекте Data метода Init | 22.06.2020 |
| 1.15 | Исправлен пример в разделе 3.2. Нотификации по http(s) | 29.09.2020 |
| 1.16 | Добавлено описание объекта Payments (в метод Init). Добавлено описание метода 2.12. Метод SendClosingReceipt Добавлено описание требований к формату передачи параметра Ean13 | 26.01.2021 |
| 1.17 | Добавлен параметр ResidentSate в таблицу и пример запроса | 17.03.2021 |

| Версия | Описание | Дата |
|--------|--|------------|
| 1.18 | Описаны двухстадийные рекурренты | 13.06.2021 |
| 1.19 | Добавлено описание метода AddAccountQr (п 3.5) Добавлено описание метода ChargeQr (п 3.6) Добавлено описание метода GetAddAccountQrState (п 3.7) Добавлено описание метода GetAccountQrList (п 3.8) | 13.09.2021 |
| 1.20 | Добавлено описание метода Нотификация о статусе привязки счета (п 4.5) | 15.10.2021 |
| 1.21 | Дополнено описание метода Init в части особенности использования параметра RedirectDueDate (п. 2.3) | 19.10.2021 |
| 1.22 | Для платежей «в Рассрочку» расширен протокол методов /confirm /cancel, расширены ответы методов /confirm /GetState, добавлены параметры в Нотификации по https | 20.10.2021 |
| 1.23 | Дополнено описание метода Init в части особенности использования для объекта Data (2.3. Метод Init) Добавлено описание метода CheckOrder | 03.12.2021 |
| 1.24 | Добавлено описание статуса ASYNC_REFUNDING. | 07.12.2021 |
| 1.25 | Добавлено описание метода SbpPayTest. Добавлены тесты по QR-коду. | 25.01.2022 |
| 1.26 | Добавлен параметр account в метод init в параметр DATA | 31.01.2022 |
| 1.27 | Удалено описание методов: AddCard, AttachCard, GetAddCardState, RemoveCard, SubmitRandomAmount. | 01.02.2022 |
| 1.28 | Доработки протокола для поддержки ФФД 1.2 в онлайн-кассах. | 18.02.2022 |
| 1.29 | Дополнено описание в пункты 2.8. Метод Confirm и 2.10. Метод Cancel, для платежей «Долями» | 18.03.2022 |
| 1.30 | Добавлено примечание по предоставлению доступа к тестовому URL | 12.05.2022 |
| 1.31 | Добавлены новые параметры в метод CheckOrder, объект Payments | 26.05.2022 |
| 1.32 | Добавлен новый метод getQRstate, удален метод GetStaticQR | 06.06.2022 |
| 1.33 | Удалено описание метода Resend | 11.07.2022 |
| 1.34 | Добавлен параметр account в метод init в параметр DATA для TinkoffPay Добавлено описание методов проведения TinkoffpayWeb 2.15 | 21.07.2022 |
| 1.35 | Добавлено примечание по закрытию платежных сессий по статусу | 28.07.2022 |

| Версия | Описание | Дата |
|--------|---|------------|
| | 3DS_CHECKING | |
| 1.36 | Удалены статусы LOOP_CHECKING и LOOP_CHECKED и обновлено описание статусов AUTHORIZING и AUTHORIZED. | 18.08.2022 |
| 1.37 | Изменили обязательность параметра rebill_id в методе GetCardList | 30.08.2022 |
| 1.38 | Добавлен список тестовых карт | 14.09.2022 |
| 1.39 | Изменена сеть адресов, с которой приходят уведомления | 27.10.2022 |
| 1.40 | Удалены ставки НДС со значениями vat 18 и vat 1 18 из параметра Tax | 15.11.2022 |
| 1.41 | Добавлен параметр QrMemberId в параметры запроса метода Cancel | 18.11.2022 |
| 1.42 | Добавлены новые параметры для поддержки ФФД 1.2 в онлайн-кассах. | 23.11.2022 |
| 1.43 | Удалено описание параметров объекта Receipts | 06.12.2022 |
| 1.44 | Обновлен боевой URL по закрывающему чеку в методе SendClosingReceipt | 09.12.2022 |
| 1.45 | Обновлено описание метода GetQr | 26.12.2022 |
| 1.46 | Обновлено описание метода QrMembersList | 09.01.2023 |
| 1.47 | Добавлен список внешних сетей, используемых при отправке нотификаций. Обновлен список портов, которые можно использовать в Notification URL | 13.02.2023 |
| 1.48 | Добавлен новый раздел: 8. Правила расчета возмещений по операционному реестру | 15.02.2023 |
| 1.49 | Добавлена информация о том, что запросы на тестовый URL необходимо осуществлять с боевого терминала. | 23.03.2023 |
| 1.50 | Обновлено описание метода ChargeQR и параметра Success | 06.04.2023 |
| 1.51 | Добавлен параметр ExternalRequestId в метод Cancel | 13.04.2023 |
| 1.52 | Добавлен раздел «Рекомендации при интеграции» | 24.04.2023 |

Термины и сокращения

| Термин | Определение |
|------------|---|
| Продавец | Участник, принимающий и осуществляющий переводы по банковским картам на своем сайте |
| Покупатель | Участник, производящий перевод с использованием банковской карты на сайте Продавца |
| PCI DSS | Стандарт безопасности данных индустрии платёжных карт. Стандарт представляет собой совокупность 12 детализированных требований по обеспечению безопасности данных о держателях платёжных карт. Данные передаются, хранятся и обрабатываются в информационных инфраструктурах организаций. Принятие соответствующих мер по обеспечению соответствия требованиям стандарта подразумевает комплексный подход к обеспечению информационной безопасности данных платёжных карт |
| 3-D Secure | Протокол, который используется как дополнительный уровень безопасности для онлайн-кредитных и дебетовых карт. 3-D Secure добавляет ещё один шаг аутентификации для онлайн-платежей |
| Терминал | Точка приема платежей продавца (в общем случае привязывается к сайту, на котором осуществляется прием платежей) Далее в этой документации описан протокол для терминала мерчанта |
| ККМ | Контрольно-кассовая машина |

Рекомендации при интеграции

Ниже мы расписали несколько рекомендаций, которые необходимо соблюдать при интеграции с MAPI через фронтенд сайта мерчанта, а именно:

1. Наиболее безопасный способ передачи данных от мерчанта в MAPI — прямая интеграция бэкенда мерчанта с бэкендом Тинькофф Кассы. В этом случае злоумышленник сможет перехватить запрос только, если окажется в локальной сети мерчанта;
2. При интеграции с MAPI через фронтенд (в том числе и с помощью нашего платежного виджета), необходимо сверять параметры созданных через платежный виджет заказов. Для того есть два способа:
 - 2.1. Получение нотификаций:
 - **По e-mail:** на указанную почту придет письмо при переходе платежа в статус «CONFIRMED»;
 - **По http:** MAPI будет отправлять POST-запрос при каждом изменении статуса платежа на URL, указанный в настройках терминала.
 - 2.2. Вызов метода GetState, который возвращает основные параметры и текущий статус платежа.

1. Параметры приема платежей

Параметры приема платежей настраиваются отдельно на каждый терминал.

Таблица 1.1.Параметры приема платежей

| Название параметра | Формат | Описание |
|-------------------------------------|---|---|
| TerminalKey | 20 символов (чувствительно к регистру) | Уникальный символьный ключ терминала. Устанавливается банком |
| Success URL | 250 символов (чувствительно к регистру) | URL на веб-сайте продавца, куда будет переведен покупатель в случае успешной оплаты * |
| Fail URL | 250 символов (чувствительно к регистру) | URL на веб-сайте продавца, куда будет переведен покупатель в случае неуспешной оплаты * |
| Success Add Card URL | 250 символов (чувствительно к регистру) | URL на веб-сайте продавца, куда будет переведен покупатель после успешной привязки карты * |
| Fail Add Card URL | 250 символов (чувствительно к регистру) | URL на веб-сайте продавца, куда будет переведен покупатель после неуспешной привязки карты * |
| Notification URL | 250 символов (чувствительно к регистру) | URL на веб-сайте продавца, куда будет отправлен POST запрос о статусе выполнения вызываемых методов. Только для методов Authorize, FinishAuthorize, Confirm, Cancel |
| Валюта терминала | 3 символа | Валюта, в которой будут происходить списания по данному терминалу, если иное не передано в запросе |
| Активность терминала | Рабочий / Неактивный / Тестовый | Определяет режим работы данного терминала |
| Password | 20 символов (чувствительно к регистру) | Используется для подписи запросов/ответов. Является секретной информацией, известной только продавцу и банку. Пароль находится в личном кабинете мерчанта https://oplata.tinkoff.ru |
| Отправлять нотификацию на Authorize | Да/Нет | Определяет, будет ли отправлена нотификация на выполнение метода Authorize (по умолчанию Нет) |

| Название параметра | Формат | Описание |
|---|--------|--|
| Отправлять нотификацию на FinishAuthorize | Да/Нет | Определяет, будет ли отправлена нотификация на выполнение метода FinishAuthorize (по умолчанию Да) |
| Отправлять нотификацию на Completed | Да/Нет | Определяет, будет ли отправлена нотификация на выполнение метода Payment (по умолчанию Да) |
| Отправлять нотификацию на Reversed | Да/Нет | Определяет, будет ли отправлена нотификация на выполнение метода Cancel (по умолчанию Да) |

* в URL можно указать необходимые параметры в виде \${<параметр>}, которые будут переданы на URL методом GET.

Таблица 1.2. Параметры Success URL и Fail URL

| Наименование | Описание |
|--------------|--|
| Success | Возможные значения: <ul style="list-style-type: none"> – true – запрос завершился успешно; – false – запрос не завершился. |
| ErrorCode | Код ошибки (0 – если ошибки не было) |
| OrderId | Номер заказа в системе Продавца |
| Message | Заголовок ошибки (заполняется только в случае ошибки) |
| Details | Детальное описание ошибки (заполняется только в случае ошибки) |

Например:

[http://tcsbank.ru/success.html?Success=\\${Success}&ErrorCode=\\${ErrorCode}&OrderId=\\${OrderId}&Message=\\${Message}&Details=\\${Details}](http://tcsbank.ru/success.html?Success=${Success}&ErrorCode=${ErrorCode}&OrderId=${OrderId}&Message=${Message}&Details=${Details})

2. Методы приема платежей

2.1. Общая информация

Прием платежей осуществляется вызовом методов с передачей параметров методом POST в формате JSON. Все методы и передаваемые параметры являются чувствительными к регистру.

Для POST запроса в заголовке должен присутствовать Content Type: application/json.

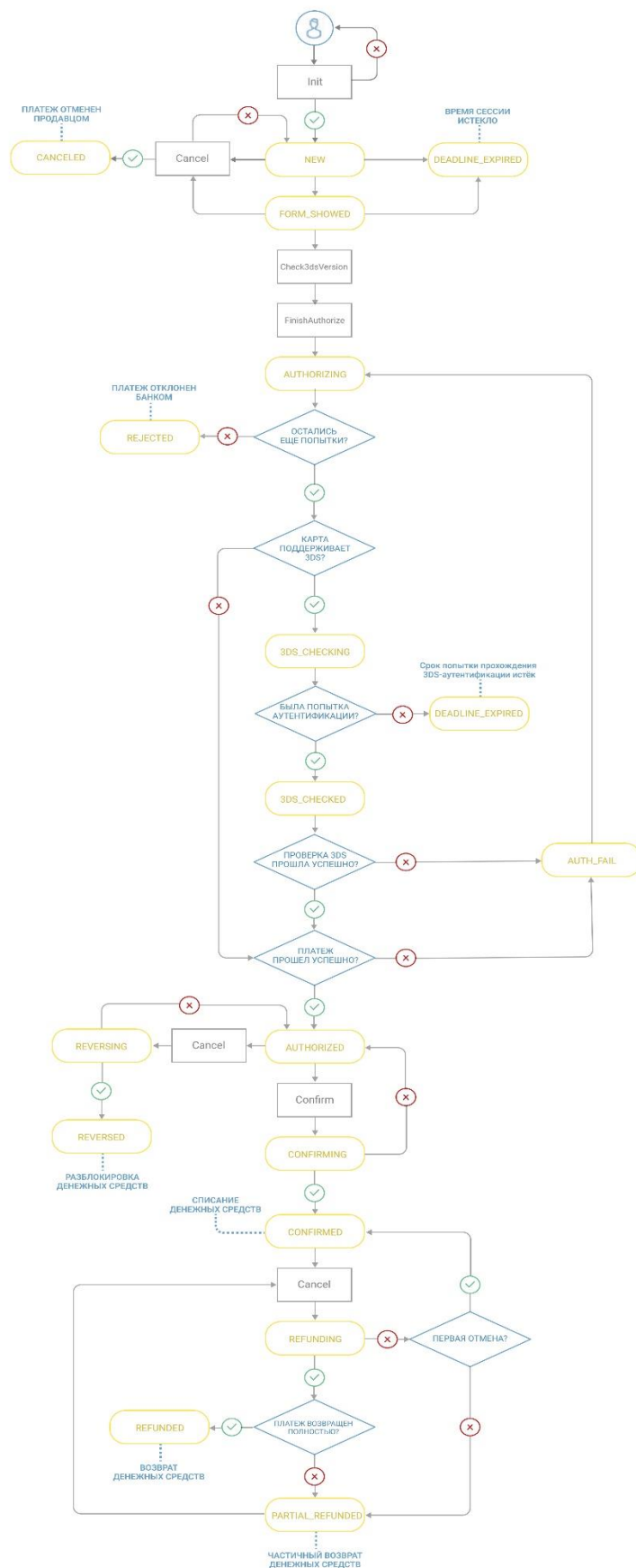
Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/>

Боевой URL: <https://securepay.tinkoff.ru/v2/>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

2.2. Схема проведения платежа

На схеме показаны статусы платежа и возможные методы, которые могут быть вызваны, если платеж находится в данном статусе.



2.3. Метод Init

Описание: Иницирует платежную сессию.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/Init/>

Боевой URL: <https://securepay.tinkoff.ru/v2/Init/>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 2.3.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|--|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| Amount* | Number | Да | Сумма в копейках |
| OrderId | String | Да | Номер заказа в системе Продавца |
| Token | String | Нет*** | Подпись запроса |
| IP | String | Нет | IP-адрес клиента |
| Description | String | Нет | Краткое описание***** |
| Currency | Number | Нет | Код валюты ISO 421. Если Currency передан и валюта разрешена для Продавца, транзакция будет инициирована в переданной валюте. Иначе будет использована валюта по умолчанию для данного терминала. В текущей версии допустимы только рубли - код 643 |
| CustomerKey | String | Да, если передан Recurrent (но не является обязательным при рекуррентных платежах через СБП) | Идентификатор покупателя в системе Продавца. Если передается и Банком разрешена автоматическая привязка карт к терминалу, то для данного покупателя будет осуществлена привязка карты. Тогда в нотификации на AUTHORIZED будет передан параметр CardId (подробнее см. |

| Наименование | Тип | Обязательность | Описание |
|-----------------|--------|-----------------------------|---|
| | | | 5.4. Метод GetCardList) |
| Recurrent | String | Для регистрации автоплатежа | Если передается и установлен в Y, то регистрирует платеж как рекуррентный. В этом случае после оплаты в нотификации на AUTHORIZED будет передан параметр RebillId для использования в методе Charge |
| PayType | Enum | Нет | Определяет тип проведения платежа – двухстадийная или одностадийная оплата. <ul style="list-style-type: none"> – "O" - одностадийная оплата; – "T"- двухстадийная оплата |
| Language | String | Нет | Язык платёжной формы. <ul style="list-style-type: none"> – ru - форма оплаты на русском языке; – en - форма оплаты на английском языке. По умолчанию (если параметр не передан) - форма оплаты на русском языке |
| NotificationURL | String | Нет | URL на веб-сайте продавца, куда будет отправлен POST запрос о статусе выполнения вызываемых методов (настраивается в Личном кабинете). Если параметр передан – используется его значение. Если нет – значение в настройках терминала. |
| SuccessURL | String | Нет | URL на веб-сайте продавца, куда будет переведен покупатель в случае успешной оплаты (настраивается в Личном кабинете). Если параметр передан – используется его значение. Если нет – значение в настройках терминала. |
| FailURL | String | Нет | URL на веб-сайте продавца, куда будет переведен покупатель в случае неуспешной оплаты (настраивается в Личном кабинете). Если параметр передан – используется его значение. Если нет – значение в настройках терминала. |

| Наименование | Тип | Обязательность | Описание |
|-----------------|------------------|--------------------------------|--|
| RedirectDueDate | Datetime | Нет | Срок жизни ссылки или динамического QR-кода СБП (если выбран данный способ оплаты)****. Если текущая дата превышает дату, переданную в данном параметре, ссылка для оплаты или возможность платежа по QR-коду становятся недоступными и платёж выполнить нельзя. Максимальное значение: 90 дней от текущей даты. Минимальное значение: 1 минута от текущей даты. Формат даты: YYYY-MM-DDTHH24:MI:SS+GMT Пример даты: 2016-08-31T12:28:00+03:00 |
| DATA** | Object | Нет | JSON объект, содержащий дополнительные параметры в виде «ключ»:«значение». Данные параметры будут переданы на страницу оплаты (в случае ее кастомизации). Максимальная длина для каждого передаваемого параметра: <ul style="list-style-type: none"> – Ключ – 20 знаков; – Значение – 100 знаков. Максимальное количество пар «ключ-значение» не может превышать 20 |
| Receipt | Object | Нет | JSON объект с данными чека |
| Shops | Array of objects | Обязательный для маркетплейсов | JSON объект с данными Маркетплейса |
| Descriptor | String | Нет | Динамический дескриптор точки |

* Параметр "Amount" должен быть равен сумме всех параметров "Amount", переданных в объекте Items. (см. ниже). Минимальная сумма операции с помощью СБП составляет 10 рублей

** Если у терминала включена опция привязки покупателя после успешной оплаты и передается параметр CustomerKey, то в передаваемых параметрах DATA могут присутствовать параметры метода AddCustomer. Если они присутствуют, то автоматически привязываются к покупателю.

Например, если указать: "DATA": {"Phone":"+71234567890", "Email":"a@test.com"}

к покупателю автоматически будут привязаны данные Email и телефон, и они будут возвращаться при вызове метода GetCustomer.

Параметр notificationEnableSource позволяет отправлять нотификации только если Source (также присутствует в параметрах сессии) платежа входит в перечень указанных в параметре. Возможные варианты: TinkoffPay, sbpqr, YandexPay.

Пример: notificationEnableSource=TinkoffPay

Для MCC 4814 обязательно передать значение в параметре "Phone".

Требования по заполнению параметра «Phone»:

- минимум 7 символов;
- максимум 20 символов;
- разрешены только цифры, исключение – первый символ может быть «+».

Для MCC 6051 и 6050 обязательно передать параметр "account" (номер электронного кошелька, не должен превышать 30 символов). Пример: "DATA": {"account": "123456789"}

Если используется функционал сохранения карт на платежной форме, то при помощи опционального параметра "DefaultCard" можно задать какая карта будет выбираться по умолчанию. Возможные варианты:

- Оставить платежную форму пустой. Пример: "DATA": {"DefaultCard": "none"};
- Заполнить данными передаваемой карты. В этом случае передается CardId. Пример: "DATA": {"DefaultCard": "894952"};
- Заполнить данными последней сохраненной карты. Применяется, если параметр "DefaultCard" не передан, передан с некорректным значением или в значении null.

По умолчанию возможность сохранения карт на платежной форме может быть отключена. Для активации обратитесь в службу технической поддержки.

При реализации подключения оплаты **Tinkoff Pay Web** на сайте мерчанта необходимо обязательно передавать следующие параметр в объекте Data:

```
"DATA": {
  "TinkoffPayWeb": "true",
  "Device": "Desktop",
  "DeviceOs": "iOS",
  "DeviceWebView": "true",
  "DeviceBrowser": "Safari"
}
```

где следует передать параметры устройства, с которого будет осуществлен переход

Таблица 2.3.1.1 Параметры запроса для Tinkoff Pay

| Наименование | Тип | Обязательность | Описание |
|---------------|---------|----------------|--|
| Device | String | Нет | Тип устройства <ul style="list-style-type: none"> • SDK (вызов из мобильных приложений) • Desktop (вызов из браузера с десктопа) • Mobile (вызов из браузера с мобильных устройств) |
| DeviceOs | String | Нет | ОС устройства |
| DeviceWebView | Boolean | Нет | Признак открытия в WebView |

| Наименование | Тип | Обязательность | Описание |
|---------------|---------|----------------|--|
| DeviceBrowser | String | Нет | Браузер |
| TinkoffPayWeb | Boolean | Нет | Признак проведения операции через Tinkoff Pay по API |

В случае если не удастся определить параметр гарантированно (режим инкогнито и пр.) – не передавать параметр.

*** Обязательность подписи запроса (token) задается отдельно через настройки терминала. Описание алгоритма формирования подписи доступно [по ссылке](#).

**** В случае, если параметр RedirectDueDate не был передан, проверяется настроечный параметр платежного терминала REDIRECT_TIMEOUT, который может содержать значение срока жизни ссылки в часах. Если его значение больше нуля, то оно будет установлено в качестве срока жизни ссылки или динамического QR-кода. Иначе, устанавливается значение «по умолчанию» - 1440 мин. (1 сутки).

***** При оплате через СБП данная информация будет отображена в приложении мобильного банка покупателя. Максимально допустимое количество знаков при передаче назначения платежа в СБП – 140 символов.

Объект Receipt при ФФД 1.05

Таблица 2.3.2. Структура объекта Receipt

| Наименование | Тип | Обязательность | Описание |
|--------------|------------------|----------------|--|
| Items | Array of objects | Да | Массив, содержащий в себе информацию о товарах |
| FfdVersion | String | Нет | Версия ФФД. Возможные значения: <ul style="list-style-type: none"> "FfdVersion": "1.2" "FfdVersion": "1.05" По умолчанию версия ФФД - 1.05. |
| Email | String | Нет* | Электронная почта для отправки чека |
| Phone | String | Нет* | Телефон для отправки чека |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Taxation | Enum | Да | Система налогообложения. Перечисление с возможными значениями: <ul style="list-style-type: none"> – «osn» – общая СН; – «usn_income» – упрощенная СН (доходы); – «usn_income_outcome» – упрощенная СН (доходы минус расходы); – «envd» – единый налог на вмененный доход; – «esn» – единый сельскохозяйственный налог; – «patent» – патентная СН. |
| Payments | Object | Нет ** | Объект с информацией о видах суммы платежа. См. Структура объекта Payments |

* Должен быть заполнен параметр или “Phone”, или “E-mail”

** Если объект не передан, будет автоматически указана итоговая сумма чека с видом оплаты “Безналичный”.

Если передан объект receipt.Payments, то значение в Electronic должно быть равно итоговому значению Amount в методе Init. При этом сумма введенных значений по всем видам оплат, включая Electronic, должна быть равна сумме (Amount) всех товаров, переданных в объекте receipt.Items.

Таблица 2.3.3. Структура объекта Shops

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| ShopCode | String | Да | Присвоенный идентификатор точки на стороне банка |
| Amount | Number | Да | Сумма в копейках, которая относится к указанному ShopCode |
| Name | String | Нет | Наименование позиции Максимум 128 символов |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Fee | String | Нет | Сумма комиссии в копейках, удерживаемая из возмещения Партнера в пользу Маркетплейса. Если не передано, используется комиссия, указанная при регистрации. |

Таблица 2.3.4. Структура объекта Items

| Наименование | Тип | Обязательность | Описание |
|---------------|--------|----------------|---|
| Name | String | Да | Наименование товара Максимум 128 символов |
| Price | Number | Да | Цена в копейках |
| Quantity | Number | Да | Количество/вес: – целая часть не более 8 знаков; – дробная часть не более 3 знаков для Атол, не более 2 знаков для CloudPayments |
| Amount | Number | Да | Сумма в копейках. Целочисленное значение не более 10 знаков |
| PaymentMethod | Enum | Нет | Признак способа расчёта. Если значение не передано, по умолчанию в онлайн-кассу передается признак способа расчёта "full_payment". Возможные значения: – «full_prepayment» – предоплата 100%. – «prepayment» – предоплата. – «advance» – аванс. – «full_payment» – полный расчет. – «partial_payment» – частичный расчет и кредит. – «credit» – передача в кредит. – «credit_payment» – оплата кредита. |
| PaymentObject | Enum | Нет | Признак предмета расчёта. Если значение не передано, по умолчанию в онлайн-кассу отправляется признак предмета расчёта "commodity". Возможные значения: – «commodity» – товар. |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| | | | <ul style="list-style-type: none"> – «excise» – подакцизный товар. – «job» – работа. – «service» – услуга. – «gambling_bet» – ставка азартной игры. – «gambling_prize» – выигрыш азартной игры. – «lottery» – лотерейный билет. – «lottery_prize» – выигрыш лотереи. – «intellectual_activity» – предоставление результатов интеллектуальной деятельности. – «payment» – платеж. – «agent_commission» – агентское вознаграждение. – «composite» – составной предмет расчета. – «another» – иной предмет расчета. |
| Тех | Enum | Да | <p>Ставка налога.</p> <p>Перечисление со значениями:</p> <ul style="list-style-type: none"> – «none» – без НДС; – «vat0» – НДС по ставке 0%; – «vat10» – НДС чека по ставке 10%; – «vat20» – НДС чека по ставке 20%; – «vat110» – НДС чека по расчетной ставке 10/110; – «vat120» – НДС чека по расчетной ставке 20/120 |
| Ean13 | String | Нет | <p>Штрих-код в требуемом формате. В зависимости от типа кассы требования могут отличаться:</p> <ul style="list-style-type: none"> • АТОЛ Онлайн - шестнадцатеричное представление с пробелами. Максимальная длина – 32 байта (^[a-fA-F0-9]{2}\$)(^[a-fA-F0-9]{2}\\s){1,31}[a-fA-F0-9]{2}\$) Пример: 00 00 00 01 00 21 FA 41 00 23 05 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12 00 AB 00 • CloudKassir - длина строки: четная, от 8 до 150 байт, т.е. от 16 до 300 ASCII символов ['0' - '9' , 'A' - 'F'] шестнадцатеричного представления кода маркировки товара. Пример: 303130323930303030630333435 • OrangeData - строка, содержащая base64 кодированный массив от 8 до 32 байт |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|--|---|
| | | | <p>Пример: igQVAAADMTIzNDU2Nzg5MDEyMwAAAAAA Q==</p> <p>В случае передачи в запросе параметра Ean13 не прошедшего валидацию, возвращается неуспешный ответ с текстом ошибки в параметре message = "Неверный параметр Ean13".</p> |
| ShopCode | String | Нет | Код магазина. Для параметра ShopCode необходимо использовать значение параметра Submerchant_ID, полученного в ответ при регистрации магазинов через xml. Если xml не используется, передавать поле не нужно |
| AgentData | Object | Да, если используется агентская схема | Данные агента. Описание параметров объекта доступно по ссылке . |
| SupplierInfo | Object | Да, если передается значение AgentSign в объекте AgentData | Данные поставщика платежного агента. Описание параметров объекта доступно по ссылке . |

Таблица 2.3.5. Структура объекта Payments

| Наименование | Тип | Обязательность | Описание |
|----------------|------------|----------------|--|
| Cash | Number(14) | Нет | Вид оплаты "Наличные". Сумма к оплате в копейках не более 14 знаков. |
| Electronic | Number(14) | Да | Вид оплаты "Безналичный". |
| AdvancePayment | Number(14) | Нет | Вид оплаты "Предварительная оплата (Аванс)". |
| Credit | Number(14) | Нет | Вид оплаты "Постоплата (Кредит)" |
| Provision | Number(14) | Нет | Вид оплаты "Иная форма оплаты". |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "Amount": 140000,
  "OrderId": "21050",
```

```
"Description": "Подарочная карта на 1000 рублей",
"Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
"DATA": {
  "Phone": "+71234567890",
  "Email": "a@test.com"},
"Receipt": {
  "Email": "a@test.ru",
  "Phone": "+79031234567",
  "Taxation": "osn",
  "Items": [
    {
      "Name": "Наименование товара 1",
      "Price": 10000,
      "Quantity": 1.00,
      "Amount": 10000,
      "Tax": "vat10",
      "Ean13": "303130323930303030630333435"
    },
    {
      "Name": "Наименование товара 2",
      "Price": 20000,
      "Quantity": 2.00,
      "Amount": 40000,
      "Tax": "vat20"
    },
    {
      "Name": "Наименование товара 3",
      "Price": 30000,
      "Quantity": 3.00,
      "Amount": 90000,
      "Tax": "vat10"
    }
  ]
}
```

Пример запроса, если используется агентская схема:

```
{
  "TerminalKey": "TinkoffBankTest",
  "Amount": 50000,
  "OrderId": "21050",
  "Description": "Подарочная карта на 1000 рублей",
  "Token": "2ED30E046136931431B5251B7C9A1EAC68DAB082203BD42676BA14A851359DF4",
  "DATA": {"Phone": "+71234567890", "Email": "a@test.com"},
  "Receipt": {
    "Email": "a@test.ru",
    "Phone": "+79031234567",
    "Taxation": "osn",
    "Customer": "TestCustomer",
    "CustomerInn": "1234567890",
    "Payments": {
      "electronic": 50000,
      "cash": 90000,
      "advancePayment": 0,
      "credit": 0,
      "provision": 0
    }
  },
  "AgentData": {
    "AgentSign": "bank_paying_subagent",
    "OperationName": "Чек",
  }
}
```

```
"Phones": ["+19221210697", "+19098561231"],
"ReceiverPhones": ["+29221210697", "+29098561231"],
"TransferPhones": ["+39221210697"],
"OperatorName": "Tinkoff",
"OperatorAddress": "г.Москва",
"OperatorInn": "7710140679"
},
"SupplierInfo": {
  "Phones": ["+49221210697", "+49098561231"]
},
"Items": [
  {
    "AgentData": {
      "AgentSign": "paying_agent",
      "OperationName": "Позиция чека",
      "Phones": ["+790912312398"],
      "ReceiverPhones": ["+79221210697", "+79098561231"],
      "TransferPhones": ["+79221210697"],
      "OperatorName": "Tinkoff",
      "OperatorAddress": "г. Тольятти",
      "OperatorInn": "7710140679"
    },
    "SupplierInfo": {
      "Phones": ["+79221210697", "+79098561231"],
      "Name": "ООО Вендор товара",
      "Inn": "7710140679"
    },
    "Name": "Наименование товара 1",
    "Price": 10000,
    "Quantity": 1.00,
    "Amount": 10000,
    "Tax": "vat10",
    "Ean13": "303130323930303030630333435",
    "ShopCode": "12345",
    "MeasurementUnit": "шт"
  },
  {
    "Name": "Наименование товара 2",
    "Price": 20000,
    "Quantity": 2.00,
    "Amount": 40000,
    "Tax": "vat20"
  },
  {
    "Name": "Наименование товара 3",
    "Price": 30000,
    "Quantity": 3.00,
    "Amount": 90000,
    "Tax": "vat10"
  }
]
}
```

Объект Receipt при ФФД 1.2

Атрибуты, предусмотренные в протоколе для отправки чеков по маркируемым товарам, не являются обязательными для товаров без маркировки. Если используется ФФД 1.2, но реализуемый товар - не подлежит маркировке, то поля можно не отправлять или отправить со значением null.

Таблица 2.3.6. Структура объекта Receipt

| Наименование | Тип | Обязательность | Описание |
|----------------------|------------------|----------------|---|
| FfdVersion | String | Да | Версия ФФД. Возможные значения: <ul style="list-style-type: none"> • "FfdVersion": "1.2" • "FfdVersion": "1.05" |
| ClientInfo | Object | Нет* | Информация по покупателю |
| Taxation | Enum | Да | Система налогообложения. Перечисление с возможными значениями: <ul style="list-style-type: none"> – «osn» – общая СН; – «usn_income» – упрощенная СН (доходы); – «usn_income_outcome» – упрощенная СН (доходы минус расходы); – «envd» – единый налог на вмененный доход; – «esn» – единый сельскохозяйственный налог; – «patent» – патентная СН. |
| Email | String | Нет** | Электронный адрес для отправки чека покупателю |
| Phone | String | Нет** | Телефон покупателя для отправки чека |
| Customer | String | Нет | Идентификатор/Имя покупателя |
| CustomerInn | String | Нет | ИНН покупателя |
| Items | Array of objects | Да | Массив, содержащий в себе информацию о товарах |
| Payments | Object | Нет *** | Объект с информацией о видах суммы платежа. См. Структура объекта Payments |
| OperatingCheckProps | Object | Нет | Операционный реквизит чека (тег 1270) |
| SectoralCheckProps | Object | Нет | Отраслевой реквизит чека (тег 1261) |
| AddUserProp | Object | Нет | Дополнительный реквизит пользователя (тег 1084) |
| AdditionalCheckProps | String | Нет | Дополнительный реквизит чека (БСО) (тег 1192) |

* Обязателен для товаров с маркировкой

** Должен быть заполнен параметр или "Phone", или "E-mail"

*** Если объект не передан, будет автоматически указана итоговая сумма чека с видом оплаты "Безналичный".

Если передан объект receipt.Payments, то значение в Electronic должно быть равно итоговому значению Amount в методе Init. При этом сумма введенных значений по всем видам оплат, включая Electronic, должна быть равна сумме (Amount) всех товаров, переданных в объекте receipt.Items.

Таблица 2.3.7. Структура объекта ClientInfo

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Birthdate | String | Нет | Дата рождения покупателя в формате ДД.ММ.ГГГГ |
| Citizenship | String | Нет | Числовой код страны, гражданином которой является покупатель. Код страны указывается в соответствии с Общероссийским классификатором стран мира ОКСМ |
| DocumentCode | String | Нет | <p>Числовой код вида документа, удостоверяющего личность.</p> <p>Может принимать только значения:</p> <p>21 - Паспорт гражданина Российской Федерации</p> <p>22 - Паспорт гражданина Российской Федерации, дипломатический паспорт, служебный паспорт, удостоверяющие личность гражданина Российской Федерации за пределами Российской Федерации;</p> <p>26 - Временное удостоверение личности гражданина Российской Федерации, выдаваемое на период оформления паспорта гражданина Российской Федерации</p> <p>27 - Свидетельство о рождении гражданина Российской Федерации (для граждан Российской Федерации в возрасте до 14 лет)</p> <p>28 - Иные документы, признаваемые документами, удостоверяющими личность гражданина Российской Федерации в соответствии с законодательством Российской Федерации</p> <p>31 - Паспорт иностранного гражданина</p> <p>32 - Иные документы, признаваемые документами, удостоверяющими личность иностранного гражданина в соответствии с законодательством Российской Федерации и международным договором Российской Федерации</p> <p>33 - Документ, выданный иностранным государством и признаваемый в соответствии с международным договором Российской Федерации в качестве документа, удостоверяющего личность лица без гражданства.</p> <p>34 - Вид на жительство (для лиц без гражданства)</p> <p>35 - Разрешение на временное проживание (для лиц без гражданства)</p> |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| | | | <p>36 - Свидетельство о рассмотрении ходатайства о признании лица без гражданства беженцем на территории Российской Федерации по существу</p> <p>37 - Удостоверение беженца</p> <p>38 - Иные документы, признаваемые документами, удостоверяющими личность лиц без гражданства в соответствии с законодательством Российской Федерации и международным договором Российской Федерации</p> <p>40 - Документ, удостоверяющий личность лица, не имеющего действительного документа, удостоверяющего личность, на период рассмотрения заявления о признании гражданином Российской Федерации или о приеме в гражданство Российской Федерации</p> |
| DocumentData | String | Нет | Реквизиты документа, удостоверяющего личность (например: серия и номер паспорта) |
| Address | String | Нет | Адрес покупателя (клиента), грузополучателя. Максимум 256 символов |

Таблица 2.3.8. Структура объекта OperatingCheckProps

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Name | String | Да | Идентификатор операции (тег 1271) Принимает значения «О» до определения значения реквизита ФНС России. |
| Value | String | Да | Данные операции (тег 1272) |
| Timestamp | String | Да | Дата и время операции в формате ДД.ММ.ГГГГ ЧЧ:ММ:СС (тег 1273) |

Таблица 2.3.9. Структура объекта SectoralCheckProps

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| FederalId | String | Да | Идентификатор ФОИВ (тег 1262). Максимальное количество символов – 3 |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| Date | String | Да | Дата документа основания в формате ДД.ММ.ГГГГ (тег 1263) |
| Number | String | Да | Номер документа основания (тег 1264) |
| Value | String | Да | Значение отраслевого реквизита (тег 1265) |

Таблица 2.3.10. Структура объекта AddUserProp

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| Name | String | Да | Наименование дополнительного реквизита пользователя (тег 1085) |
| Value | String | Да | Значение дополнительного реквизита пользователя (тег 1086) |

Таблица 2.3.11. Структура объекта Items

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|--|--|
| AgentData | Object | Да, если используется агентская схема | Данные агента Описание параметров объекта доступно по ссылке |
| SupplierInfo | Object | Да, если передается значение AgentSign в объекте AgentData | Данные поставщика платежного агента. Описание параметров объекта доступно по ссылке |
| Name | String | Да | Наименование товара Максимум 128 символов |
| Price | Number | Да | Цена в копейках |
| Quantity | Number | Да | Количество/вес: – целая часть не более 8 знаков; – дробная часть не более 3 знаков для Атол, не более 2 знаков для CloudPayments Значение «1», если передан объект markCode |

| Наименование | Тип | Обязательность | Описание |
|---------------|--------|----------------|--|
| Amount | Number | Да | Сумма в копейках. Целочисленное значение не более 10 знаков |
| Tax | Enum | Да | Ставка налога. Перечисление со значениями: <ul style="list-style-type: none"> – «none» – без НДС; – «vat0» – НДС по ставке 0%; – «vat10» – НДС чека по ставке 10%; – «vat20» – НДС чека по ставке 20%; – «vat110» – НДС чека по расчетной ставке 10/110; – «vat120» – НДС чека по расчетной ставке 20/120 |
| PaymentMethod | String | Да | Признак способа расчёта. Возможные значения: <ul style="list-style-type: none"> • «full_prepayment» – предоплата 100%. • «prepayment» – предоплата. • «advance» – аванс. • «full_payment» – полный расчет. • «partial_payment» – частичный расчет и кредит. • «credit» – передача в кредит. • «credit_payment» – оплата кредита. |
| PaymentObject | String | Да | Значения реквизита "признак предмета расчета" (тег 1212) таблица 101 Возможные значения соответствуют: <ul style="list-style-type: none"> – «commodity» – товар. – «excise» – подакцизный товар. – «job» – работа. – «service» – услуга. – «gambling_bet» – ставка азартной игры. – «gambling_prize» – выигрыш азартной игры. – «lottery» – лотерейный билет. – «lottery_prize» – выигрыш лотереи. – «intellectual_activity» – предоставление результатов интеллектуальной деятельности. – «payment» – платеж. – «agent_commission» – агентское вознаграждение. – «contribution» - Выплата – «property_rights» - Имущественное право – «unrealization» - Внереализационный доход – «tax_reduction» - Иные платежи и взносы |

| Наименование | Тип | Обязательность | Описание |
|--------------------|--------|----------------|--|
| | | | <ul style="list-style-type: none"> – «trade_fee» - Торговый сбор – «resort_tax» - Курортный сбор – «pledge» - Залог – «income_decrease» - Расход – «ie_pension_insurance_without_payments» - Взносы на ОПС ИП – «ie_pension_insurance_with_payments» - Взносы на ОПС – «ie_medical_insurance_without_payments» - Взносы на ОМС ИП – «ie_medical_insurance_with_payments» - Взносы на ОМС – «social_insurance» - Взносы на ОСС – «casino_chips» - Платеж казино – «agent_payment» - Выдача ДС – «excisable_goods_without_marking_code» - АТНМ – «excisable_goods_with_marking_code» - АТМ – «goods_without_marking_code» - ТНМ – «goods_with_marking_code» - ТМ«another» – иной предмет расчета. |
| UserData | String | Нет | Дополнительный реквизит предмета расчета. |
| Excise | Number | Нет | Сумма акциза в рублях с учетом копеек, включенная в стоимость предмета расчета. Целая часть не более 8 знаков; дробная часть не более 2 знаков; значение не может быть отрицательным; |
| CountryCode | String | Нет | Цифровой код страны происхождения товара в соответствии с Общероссийским классификатором стран мира (3 цифры) |
| DeclarationNumber | String | Нет | Номер таможенной декларации (32 цифры максимум) |
| MeasurementUnit | String | Да | Единицы измерения. Передавать в соответствии с ОК 015-94 (МК 002-97)) |
| MarkProcessingMode | String | Нет* | Режим обработки кода маркировки. Должен принимать значение равное «0». |

| Наименование | Тип | Обязательность | Описание |
|-------------------|------------------|----------------|--|
| MarkCode | Object | Нет* | Код маркировки в машиночитаемой форме, представленный в виде одного из видов кодов, формируемых в соответствии с требованиями, предусмотренными правилами, для нанесения на потребительскую упаковку, или на товары, или на товарный ярлык Подробнее см. в таблице 2.3.10 |
| MarkQuantity | Object | Нет** | Реквизит «дробное количество маркированного товара» |
| SectoralItemProps | Array of objects | Нет*** | Отраслевой реквизит предмета расчета Подробнее см. в таблице 2.3.11 |

* Включается в чек в случае, если предметом расчета является товар, подлежащий обязательной маркировке средством идентификации (соответствующий код в поле paymentObject).

** Передается только в случае, если расчет осуществляется за маркированный товар (соответствующий код в поле paymentObject) и значение в поле measurementUnit равно «0»

Состоит из двух параметров, где

- Numerator – Числитель дробной части предмета расчета. Значение должно быть строго меньше значения реквизита «знаменатель».
- Denominator - Знаменатель дробной части предмета расчета. Значение равно количеству товара в партии (упаковке), имеющей общий код маркировки товара.

Пример:

```
{
  "numerator": "1",
  "denominator": "2"
}
```

*** Необходимо указывать только для товаров подлежащих обязательной маркировке средством идентификации и включение данного реквизита предусмотрено НПА отраслевого регулирования для соответствующей товарной группы.

Таблица 2.3.12. Структура объекта MarkCode

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| MarkCodeType | String | Да | Тип штрих кода Возможные значения: <ul style="list-style-type: none"> • Unknown - код товара, формат которого не идентифицирован, как один из реквизитов • ean8 - код товара в формате EAN-8. • ean13 - код товара в формате EAN-13 • itf14 - код товара в формате ITF-14 |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| | | | <ul style="list-style-type: none"> gs1O - код товара в формате GS1, нанесенный на товар, не подлежащий маркировке gs1m - код товара в формате GS1, нанесенный на товар, подлежащий маркировке short - код товара в формате короткого кода маркировки, нанесенный на товар, fur - контрольно-идентификационный знак мехового изделия. egais2O - код товара в формате ЕГАИС-2.О. - egais3O - код товара в формате ЕГАИС-3.О. |
| value | String | Да | Код маркировки. |

Таблица 2.3.13. Структура объекта SectorallItemProps

| Наименование | Тип | Обязательность | Описание |
|--------------|---------------|----------------|---|
| FederalId | Enum (string) | Да | Идентификатор ФОИВ (федеральный орган исполнительной власти). |
| Date | String | Да | Дата нормативного акта ФОИВ |
| Number | String | Да | Номер нормативного акта ФОИВ |
| Value | String | Да | Состав значений, определенных нормативного актом ФОИВ. |

Пример запроса с маркировкой:

```
{
  "TerminalKey": "TinkoffBankTest",
  "Amount": 90000,
  "OrderId": "{{timestamp}}",
  "Description": "RS_TESTING",
  "CustomerKey": "a.grigorash",
  "Receipt": {
    "FfdVersion": "1.2",
    "ClientInfo": {
      "Birthdate": "21.11.1995",
      "Citizenship": "643",
      "DocumentCode": "40",
      "DocumentData": "4507 443564",
      "Address": "г. Краснодар ул. Привокзальная 1"
    }
  },
  "Taxation": "osn",
}
```

```
"Email": "ext.test.qa@tinkoff.ru",
"Phone": "88005553535",
"Customer": "Клиент",
"CustomerInn": "516974792202",
"Items": [
  {
    "AgentData": {
      "AgentSign": "paying_agent",
      "OperationName": "Позиция чека",
      "Phones": [
        "+790912312398"
      ],
      "ReceiverPhones": [
        "+79221210697",
        "+79098561231"
      ],
      "TransferPhones": [
        "+79221210697"
      ],
      "OperatorName": "Tinkoff",
      "OperatorAddress": "г. Тольятти",
      "OperatorInn": "7710140679"
    },
    "SupplierInfo": {
      "Phones": [
        "+79221210697",
        "+79098561231"
      ],
      "Name": "ООО Вендор товара",
      "Inn": "7710140679"
    },
    "Name": "Тестовый товар",
    "Price": 100000,
    "Quantity": 1,
    "Amount": 90000,
    "Tax": "vat20",
    "PaymentMethod": "full_prepayment",
    "PaymentObject": "goods_with_marking_code",
    "UserData": "Данные пользователя ext.test.qa@tinkoff.ru",
    "Excise": 12.2,
    "CountryCode": "056",
    "DeclarationNumber": "12345678901",
    "MeasurementUnit": "шт",
    "MarkProcessingMode": "O",
    "MarkCode": {
      "MarkCodeType": "EAN8",
      "Value": "12345678"
    },
    "MarkQuantity": {
      "Denominator": "2",
      "Numerator": "1"
    },
    "SectoralItemProps": [
      {
        "Number": "123/43",
        "Date": "21.11.2020",
        "Value": "test value SectoralItemProps",
        "FederalId": "001"
      }
    ]
  }
]
```



```
]
}
```

Пример запроса без маркировки:

```
{
  "TerminalKey": "TinkoffBankTest",
  "Amount": 90000,
  "OrderId": "{{timestamp}}",
  "Description": "RS_TESTING",
  "CustomerKey": "a.grigorash",
  "Receipt": {
    "FfdVersion": "1.2",
    "Taxation": "osn",
    "Email": "ext.test.qa@tinkoff.ru",
    "Phone": "88005553535",
    "Customer": "Клиент",
    "CustomerInn": "516974792202",
    "Items": [
      {
        "AgentData": {
          "AgentSign": "paying_agent",
          "OperationName": "Позиция чека",
          "Phones": ["+790912312398"],
          "ReceiverPhones": ["+79221210697", "+79098561231"],
          "TransferPhones": ["+79221210697"],
          "OperatorName": "Tinkoff",
          "OperatorAddress": "г. Тольятти",
          "OperatorInn": "7710140679"
        },
        "SupplierInfo": {
          "Phones": ["+79221210697", "+79098561231"],
          "Name": "ООО Вендор товара",
          "Inn": "7710140679"
        },
        "Name": "Тестовый товар",
        "Price": 100000,
        "Quantity": 1,
        "Amount": 90000,
        "Tax": "vat20",
        "PaymentMethod": "full_prepayment",
        "PaymentObject": "commodity",
        "MeasurementUnit": "шт"
      }
    ]
  }
}
```

Ответ

Таблица 2.3.14. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| Amount | Number | Да | Сумма в копейках |
| OrderId | String | Да | Номер заказа в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| Status | String | Да | Статус транзакции |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |
| PaymentURL | String | Нет | Ссылка на платежную форму |

Пример ответа:

```
{
  "Success": true,
  "ErrorCode": "O",
  "TerminalKey": "TinkoffBankTest",
  "Status": "NEW",
  "PaymentId": "13660",
  "OrderId": "21050",
  "Amount": 100000,
  "PaymentURL": "https://securepay.tinkoff.ru/rest/Authorize/1B63Y1"
}
```

Статус платежа:

при успешном сценарии: NEW;

при неуспешном: REJECTED.

2.4. Метод Authorize

Описание: Метод вызывается только при использовании терминала банка. Вызов происходит автоматически при переадресации покупателя на страницу PaymentURL, указанную в ответе на Init. Статус платежа выставляется в FORM_SHOWED.

2.5. Метод FinishAuthorize

Описание: Подтверждает инициированный платеж передачей карточных данных. При использовании одностадийного проведения осуществляет списание денежных средств с карты покупателя. При двухстадийном проведении осуществляет блокировку указанной суммы на карте покупателя.

2.6. Оплата картой на платежной форме банка

Вызывается формой оплаты, доступной по адресу PaymentURL, при заполнении покупателем карточных данных и нажатии кнопки «Оплатить».

Статус платежа:

- при успешном сценарии и одностадийном проведении платежа: CONFIRMED;
- при успешном сценарии и двухстадийном проведении платежа: AUTHORIZED;
- при неуспешном: REJECTED.

Переадресация покупателя:

- в случае успешного проведения платежа на Success URL;
- в случае неуспешного проведения платежа на Fail URL.

2.7 Метод Confirm

Описание: Осуществляет списание заблокированных денежных средств. Используется при двухстадийном проведении платежа.

Применяется только к платежам в статусе AUTHORIZED. Статус транзакции перед разблокировкой выставляется в CONFIRMING. Сумма списания может быть меньше или равна сумме авторизации.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/Confirm>

Боевой URL: <https://securepay.tinkoff.ru/v2/Confirm>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 2.7.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|--------------------------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| Token | String | Да | Подпись запроса |
| IP | String | Нет | IP-адрес клиента |
| Amount | Number | Нет | Сумма в копейках (если не передан, используется Amount, переданный в методе Init) |
| Receipt | Object | Нет | JSON объект с данными чека |
| Shops | Object | Обязательный для маркетплейсов | JSON объект с данными Маркетплейса |
| Route | Enum | Нет | Способ платежа. При проведении платежа «в Рассрочку» необходимо передавать значение TCB При проведении платежа «Долями» необходимо передавать значение BNPL |
| Source | Enum | Нет | Источник платежа. При проведении платежа «в Рассрочку» необходимо передавать значение Installment При проведении платежа «Долями» необходимо передавать значение BNPL |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "PaymentId": "10063",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ
Формат ответа: JSON
Таблица 2.7.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Да | Номер заказа в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| Status | String | Да | Статус транзакции |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |
| Params* | Object | Нет | Детали |

* для платежей в «Рассрочку»

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|------------------------------------|
| Route | Enum | Нет | Способ платежа (TCB) |
| Source | Enum | Нет | Источник платежа (Installment) |
| CreditAmount | Number | Нет | Сумма выданного кредита в копейках |

Пример ответа:

```
{
  "Success": true,
  "ErrorCode": "O",
  "TerminalKey": "TinkoffBankTest",
  "Status": "CONFIRMED",
  "PaymentId": "2164657",
  "OrderId": "PAYMENT117539"
}
```

```
}
```

Пример ответа для платежа «в Рассрочку»:

```
{  
  "Success": true,  
  "ErrorCode": "O",  
  "TerminalKey": "TinkoffBankTest",  
  "Status": "CONFIRMED",  
  "PaymentId": "454622",  
  "OrderId": "PAYMENT114569"  
  "Params": {  
    "Route": "TCB",  
    "Source": "Installment",  
    "CreditAmount": 1000000  
  }  
}
```

2.8. Метод Charge

Описание: Осуществляет рекуррентный (повторный) платеж — безакцептное списание денежных средств со счета банковской карты Покупателя.

Для возможности его использования Покупатель должен совершить хотя бы один платеж в пользу Продавца, который должен быть указан как рекуррентный (см. параметр Recurrent в 2.3. Метод Init), фактически являющийся первичным. По завершении оплаты в нотификации на AUTHORIZED или CONFIRMED будет передан параметр RebillId.

В дальнейшем для совершения рекуррентного платежа Продавец должен вызвать метод Init, а затем без переадресации на PaymentURL вызвать метод Charge для оплаты по тем же самым реквизитам и передать параметр RebillId, полученный при совершении первичного платежа.

Метод Charge работает по одностадийной и двухстадийной схеме оплаты. Чтобы перейти на двухстадийную схему нужно переключить терминал в Личном кабинете <https://oplata.tinkoff.ru>, а также написать обращение на acq_help@tinkoff.ru с просьбой переключить схему рекуррентов.

Для использования рекуррентных платежей по одностадийной схеме необходима следующая последовательность действий:

1. Совершить родительский платеж путем вызова Init с указанием дополнительных параметров Recurrent=Y и CustomerKey.
2. Переадресовать Покупателя на PaymentUrl.
3. После оплаты заказа Покупателем в нотификации на статус AUTHORIZED или CONFIRMED будет передан параметр RebillId, который необходимо сохранить.
4. Спустя некоторое время для совершения рекуррентного платежа необходимо вызвать метод Init со стандартным набором параметров (параметры Recurrent и CustomerKey здесь не нужны).
5. Получить в ответ на Init параметр PaymentId, при этом переадресацию пользователя на PaymentUrl производить не надо.
6. Вызвать метод Charge с параметром RebillId, полученным в п.3, и параметром PaymentId, полученным в п.5.

Для использования рекуррентных платежей по двухстадийной схеме необходима следующая последовательность действий:

1. Совершить родительский платеж путем вызова Init с указанием дополнительных параметров Recurrent=Y и CustomerKey.
2. Переадресовать Покупателя на PaymentUrl.
3. После оплаты заказа Покупателем в нотификации на статус AUTHORIZED или CONFIRMED будет передан параметр RebillId, который необходимо сохранить.
4. Спустя некоторое время для совершения рекуррентного платежа необходимо вызвать метод Init со стандартным набором параметров (параметр Recurrent и CustomerKey здесь не нужны).
5. Получить в ответ на Init параметр PaymentId, при этом переадресацию пользователя на PaymentUrl производить не надо.
6. Вызвать метод Charge с параметром RebillId, полученным в п.3, и параметром PaymentId, полученным в п.5. При успешном сценарии операция перейдет в статус AUTHORIZED. Денежные средства будут заблокированы на карте покупателя.
7. Вызвать метод Confirm для подтверждения платежа.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/Charge>

Боевой URL: <https://securepay.tinkoff.ru/v2/Charge>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 2.8.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|---|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка, полученный в ответе на вызов метода Init |
| RebillId | String | Да | Идентификатор рекуррентного платежа (см. параметр Recurrent в методе Init) |
| Token | String | Да | Подпись запроса |
| IP | String | Нет | IP-адрес клиента |
| SendEmail | bool | Нет | true – если покупатель хочет получать уведомления на почту |
| InfoEmail | String | Нет (Обязателен при передаче SendEmail) | Адрес почты покупателя |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "PaymentId": "10063",
  "RebillId": "145919",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 2.8.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Да | Номер заказа в системе Продавца |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| Status | String | Да | Статус транзакции |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| Amount | Number | Да | Сумма списания в копейках |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "OrderId": "100668",
  "Success": true,
  "Status": "CONFIRMED",
  "PaymentId": "63100",
  "Amount": 444,
  "ErrorCode": "O",
}
```

2.9. Метод Cancel

Описание: Отменяет платежную сессию. В зависимости от статуса платежа переводит его в следующие состояния:

Таблица 2.9.1. Статусы платежа

| Начальный статус | Статус после проведения операции |
|------------------|---|
| NEW | CANCELED |
| AUTHORIZED | PARTIAL_REVERSED – если отмена не на полную сумму |
| AUTHORIZED | REVERSED – если отмена на полную сумму |
| CONFIRMED | PARTIAL_REFUNDED – если отмена не на полную сумму |
| CONFIRMED | REFUNDED – если отмена на полную сумму |

Для платежей «в Рассрочку» отмена доступна только из статуса AUTHORIZED.

Для платежей «Долями» если операция в статусе CONFIRMED или PARTIAL_REFUNDED будет осуществлен частичный либо полный возврат

Если платеж находился в статусе AUTHORIZED производится отмена холдирования средств на карте клиента. При переходе из статуса CONFIRMED – возврат денежных средств на карту клиента.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/Cancel>

Боевой URL: <https://securepay.tinkoff.ru/v2/Cancel>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 2.9.2. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| Token | String | Да | Подпись запроса |
| IP | String | Нет | IP-адрес клиента |
| Amount | Number | Нет* | Сумма отмены в копейках. |

| Наименование | Тип | Обязательность | Описание |
|-------------------|--------|----------------|---|
| Receipt | Object | Нет** | JSON объект с данными чека (Описание см. в методе Init). Данные переданные в данном запросе могут отличаться от переданных в Init, если отмена производится только по части товаров |
| Shops | Object | Нет | JSON объект с данными Маркетплейса |
| Route | Enum | Нет | Способ платежа. При проведении платежа «в Рассрочку» необходимо передавать значение TCB При проведении платежа «Долями» необходимо передавать значение BNPL |
| Source | Enum | Нет | Источник платежа. При проведении платежа «в Рассрочку» необходимо передавать значение Installment При проведении платежа «Долями» необходимо передавать значение Bnpl |
| QrMemberId | String | Нет | Код банка в классификации СБП, в который необходимо выполнить возврат См параметр MemberId в Методе QrMembersList |
| ExternalRequestId | String | Нет*** | Идентификатор операции на стороне мерчанта |

* в случае отмены статуса NEW поле Amount, даже если оно проставлено, игнорируется. Отмена производится на полную сумму.

** в случае полной отмены структура чека не передается. В случае частичной отмены необходимо передавать те товары, которые нужно отменить.

*** Если поле заполнено, то перед проведением возврата проверяется запрос на отмену с таким ExternalRequestId:

- если такой запрос уже есть, то в ответе вернется текущее состояние платежной операции;
- если такого запроса нет, то произойдет отмена платежа.

Если поле не передано или пустое (""), то запрос будет обработан без проверки ранее созданных возвратов.

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest"
  "PaymentId": "10063"
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 2.9.3. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|-------------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| Status | String | Да | Статус транзакции |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| OrderId | String | Да | Номер заказа в системе Продавца |
| OriginalAmount | Number | Да | Сумма в копейках до операции отмены |
| NewAmount | Number | Да | Сумма в копейках после операции отмены |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |
| ExternalRequestId | String | Нет | Идентификатор операции на стороне мерчанта |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "Success": true,
  "Status": "REVERSED",
  "ErrorCode": "O",
  "PaymentId": "2167708",
  "OrderId": "PAYMENT117558",
  "OriginalAmount": 1000,
  "NewAmount": 0
}
```

2.10. Метод GetState

Описание: Возвращает статус платежа.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/GetState>

Боевой URL: <https://securepay.tinkoff.ru/v2/GetState>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 2.10.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| Token | String | Да | Подпись запроса |
| IP | String | Нет | IP-адрес клиента |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "PaymentId": "10063",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 2.10.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Да | Номер заказа в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| Status | String | Да | Статус транзакции |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Amount | Number | Да | Сумма операции в копейках |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |
| Params* | Object | Нет | Детали |

*для платежей в «Рассрочку»

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|------------------------------------|
| Route | Enum | Нет | Способ платежа (TCB) |
| Source | Enum | Нет | Источник платежа (Installment) |
| CreditAmount | Number | Нет | Сумма выданного кредита в копейках |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "OrderId": "21057",
  "Success": true,
  "Status": "NEW",
  "PaymentId": "10063",
  "ErrorCode": "O",
  "Amount": 1000
}
```

Таблица 2.10.3. Возможные статусы транзакции

| Статус | Промежуточный? | Значение |
|----------------|----------------|--|
| NEW | Нет | Платеж зарегистрирован в шлюзе, но его обработка в процессинге не начата |
| CANCELED | Нет | Платеж отменен Продавцом |
| PREAUTHORIZING | Да | Проверка платежных данных Покупателя |
| FORM_SHOWED | Нет | Покупатель переправлен на страницу оплаты |

| Статус | Промежуточный? | Значение |
|------------------|----------------|--|
| AUTHORIZING | Да | Покупатель начал аутентификацию |
| 3DS_CHECKING | Нет | Покупатель начал аутентификацию по протоколу 3-D Secure. Сессии, находящиеся в статусе 3DS_CHECKING более 36 часов, будут автоматически закрываться. |
| 3DS_CHECKED | Да | Покупатель завершил аутентификацию по протоколу 3-D Secure |
| AUTH_FAIL | Да | Не пройдена проверка по протоколу 3-D Secure |
| PAY_CHECKING | Да | Платеж обрабатывается |
| AUTHORIZED | Нет | Средства заблокированы, но не списаны |
| REVERSING | Да | Начало отмены блокировки средств |
| REVERSED | Нет | Денежные средства разблокированы |
| CONFIRMING | Да | Начало списания денежных средств |
| CONFIRM_CHECKING | Да | Платеж обрабатывается |
| CONFIRMED | Нет | Денежные средства списаны |
| REFUNDING | Да | Начало возврата денежных средств |
| ASYNC_REFUNDING | Да | Обработка возврата денежных средств по QR |
| PARTIAL_REFUNDED | Нет | Произведен частичный возврат денежных средств |
| REFUNDED | Нет | Произведен возврат денежных средств |
| REJECTED | Нет | Платеж отклонен Банком |
| UNKNOWN | Да | Статус не определен |
| DEADLINE_EXPIRED | Нет | Платежная сессия закрыта в связи с превышением срока отсутствия активности по текущему статусу |

2.1 1. Метод CheckOrder

Описание: Возвращает статус заказа.

Запрос

Тестовый URL *: <https://rest-api-test.tinkoff.ru/v2/CheckOrder>

Боевой URL: <https://securepay.tinkoff.ru/v2/CheckOrder>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 2.1 1.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | Number | Да | Номер заказа в системе Продавца |
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "OrderId": "21057",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 2.1 1.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Да | Номер заказа в системе Продавца |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|----------|
| Payments | Object | Да | Детали |

Таблица 2.1 1.3. Структура объекта Payments

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| Amount | Number | Нет | Сумма операции в копейках |
| Status | String | Да | Статус транзакции |
| RRN | String | Нет | RRN операции |
| Success | String | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | Number | Да | Код ошибки, «О» - если успешно |
| Message | String | Да | Краткое описание ошибки |

Пример ответа:

```
{
  "Success": true,
  "ErrorCode": "O",
  "Message": "OK",
  "OrderId": "21057",
  "TerminalKey": "TinkoffBankTest",
  "Payments": [
    {
      "Status": "REJECTED",
      "PaymentId": "10063",
      "Rrn": 1234567,
      "Amount": 555,
      "Success": false,
      "ErrorCode": "1051",
      "Message": "Недостаточно средств на карте"
    },
    {
      "Status": "AUTH_FAIL",
      "PaymentId": "1005563",
      "Rrn": 1234567,
      "Amount": 555,
      "Success": false,
      "ErrorCode": "76",
      "Message": "Операция по иностранной карте недоступна."
    },
    {
      "Status": "NEW",
      "PaymentId": "100553363",

```

```
"Rrn": 1234567,  
"Amount": 555,  
"Success": true,  
"ErrorCode": "O",  
"Message": "ok"  
}  
]  
}
```

2.12. Метод SendClosingReceipt

Описание: Метод позволяет отправить закрывающий чек в кассу.

Условия работы метода:

1. Закрывающий чек может быть отправлен если платежная сессия по первому чеку находится в статусе CONFIRMED.
2. В платежной сессии был передан объект Receipt.
3. В объекте Receipt был передан хотя бы один объект Receipt.Items.PaymentMethod = "full_prepayment" или "prepayment" или "advance"

Запрос

Тестовый URL *: <https://rest-api-test.tinkoff.ru/v2/SendClosingReceipt/>

Боевой URL: <https://securepay.tinkoff.ru/cashbox/SendClosingReceipt/>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 2.12.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| Token | String | Да | Подпись запроса |
| Receipt | String | Да | Массив данных чека см. Структура объекта Receipt |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "PaymentId": "700000105771",
  "Receipt": {
    "Email": "test@tinkoff.ru",
    "Taxation": "osn",
    "Payments": {
      "electronic": 5000,
      "cash": 0,
      "advancePayment": 5000,
      "credit": 0,
      "provision": 0
    },
    "Items": [
```

```
{
  "AgentData": {
    "AgentSign": "bank_paying_agent",
    "OperationName": "Позиция чека 1 ",
    "Phones": [
      "+823456781012141611"
    ],
    "ReceiverPhones": [
      "+923456781012141611",
      "+133456781012141611"
    ],
    "TransferPhones": [
      "+136456781012141611"
    ],
    "OperatorName": "Tinkoff",
    "OperatorAddress": "г. Тольятти",
    "OperatorInn": "7710140679"
  },
  "SupplierInfo": {
    "Phones": [
      "88001007755",
      "+74959565555"
    ],
    "Name": "СПАО \\"Ингосстрах\\"",
    "Inn": "7705042179"
  },
  "Name": "Название товара 1 ",
  "Price": 10000,
  "Quantity": 1,
  "Amount": 10000,
  "Tax": "vat20",
  "PaymentMethod": "full_payment",
  "PaymentObject": "lottery_prize"
}
],
},
"Token": "9e2e9b311e3006be4c2411e294c1b4ff3a71b25b315695a2fb494b6e19fae35c"
}
```

Ответ

Формат ответа: JSON

Таблица 2.12.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |

Пример ответа:

```
{  
  "Success": true,  
  "ErrorCode": "O",  
  "Message": null  
}
```

2.13. Оплата картой на платежной форме банка

Вызывается формой оплаты, доступной по адресу PaymentURL, при заполнении покупателем

карточных данных и нажатии кнопки «Оплатить».

Статус платежа:

- при успешном сценарии и одностадийном проведении платежа: CONFIRMED;
- при успешном сценарии и двухстадийном проведении платежа: AUTHORIZED;
- при неуспешном: REJECTED.

Переадресация покупателя:

- в случае успешного проведения платежа на Success URL;
- в случае неуспешного проведения платежа на Fail URL.

2.14. Оплата через Tinkoff Pay Web на сайте мерчанта

Оплата доступна на мобильных устройствах и десктопах, проводится последовательным вызовом методов:

- /TinkoffPay/terminals/{terminalKey}/status
- /init
- /TinkoffPay/transactions/{paymentId}/versions/{version}/link либо /TinkoffPay/{paymentId}/QR

2.14.1. Метод TinkoffPay/terminals/{terminalKey}/status

Описание: определение возможности проведения платежа Tinkoff Pay на терминале и устройстве

Запрос

Тестовый URL: <https://rest-api-test.tinkoff.ru/v2/TinkoffPay/terminals/{terminalKey}/status>

Боевой URL: <https://securepay.tinkoff.ru/v2/TinkoffPay/terminals/{terminalKey}/status>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: GET

Таблица 2.14.1.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |

Ответ

Таблица 2.14.1.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |
| Params | Object | Да | Параметры ответа |

Таблица 2.14.1.3. Структура объекта Params

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Allowed | bool | Да | Наличие возможности проведения оплаты Tinkoff Pay по API, SDK |
| Version | String | Нет* | Версия Tinkoff Pay, доступная на терминале: 1.0 (e-invoice) 2.0 (Tinkoff Pay) |

*Обязательное при Allowed = true

Если версия в ответе метода 1.0, то оплата может быть проведена только с мобильного устройства.

Пример ответа:

```
{
  "Success":true,
  "ErrorCode":"O",
  "Params":{
    "Allowed":true,
    "Version":"1.0"
  }
}
```

2.14.2. Метод TinkoffPay/transactions/{paymentId}/versions/{version}/link

Описание: получение Link для безусловного редиректа на мобильных устройствах

Запрос

Тестовый URL: <https://rest-api-test.tinkoff.ru/v2/TinkoffPay/transactions/{paymentId}/versions/{version}/link>

Боевой URL: <https://securepay.tinkoff.ru/v2/TinkoffPay/transactions/{paymentId}/versions/{version}/link>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: GET

Таблица 2.14.2.1 Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| paymentId | Number | Да | Идентификатор платежа |
| version | String | Да | Версия Tinkoff Pay: 1.0 (e-invoice) 2.0 (Tinkoff Pay) |

Ответ

Таблица 2.14.2.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |
| Params* | Object | Да | Параметры ответа |

Таблица 2.14.2.3. Структура объекта Params

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| RedirectUrl | String | Да | Link для перехода в приложение MB на мобильных устройствах |
| WebQR | String | Нет | URL для получения QR |

Пример ответа:

```
{
  "Success": true,
  "ErrorCode": "O",
  "Params": {
    "RedirectUrl": "https://www.tinkoff.ru/tpay/2000000000000037338",
    "WebQR": "https://securepay.tinkoff.ru/v2/TinkoffPay/2332790853/QR"
  }
}
```

Устаревший ответ для версии 1.0(e-invoice):

```
{
  "Success": true,
  "ErrorCode": "O",
  "Params": {
    "RedirectUrl": "tinkoffbank://Main/Elnoicing?billId=5000015507&providerId=e-invoicing"
  }
}
```

2.14.3. Метод GET /v2/TinkoffPay/{paymentId}/QR

Описание: получение QR для десктопов

Запрос

Тестовый URL: <https://rest-api-test.tinkoff.ru/v2/TinkoffPay/{paymentId}/QR>

Боевой URL: <https://securepay.tinkoff.ru/v2/TinkoffPay/{paymentId}/QR>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: GET

Таблица 2.14.3.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|-----------------------|
| paymentId | Number | Да | Идентификатор платежа |

Ответ

Формат ответа: imageSVG

Таблица 2.14.3.2. Параметры ответа

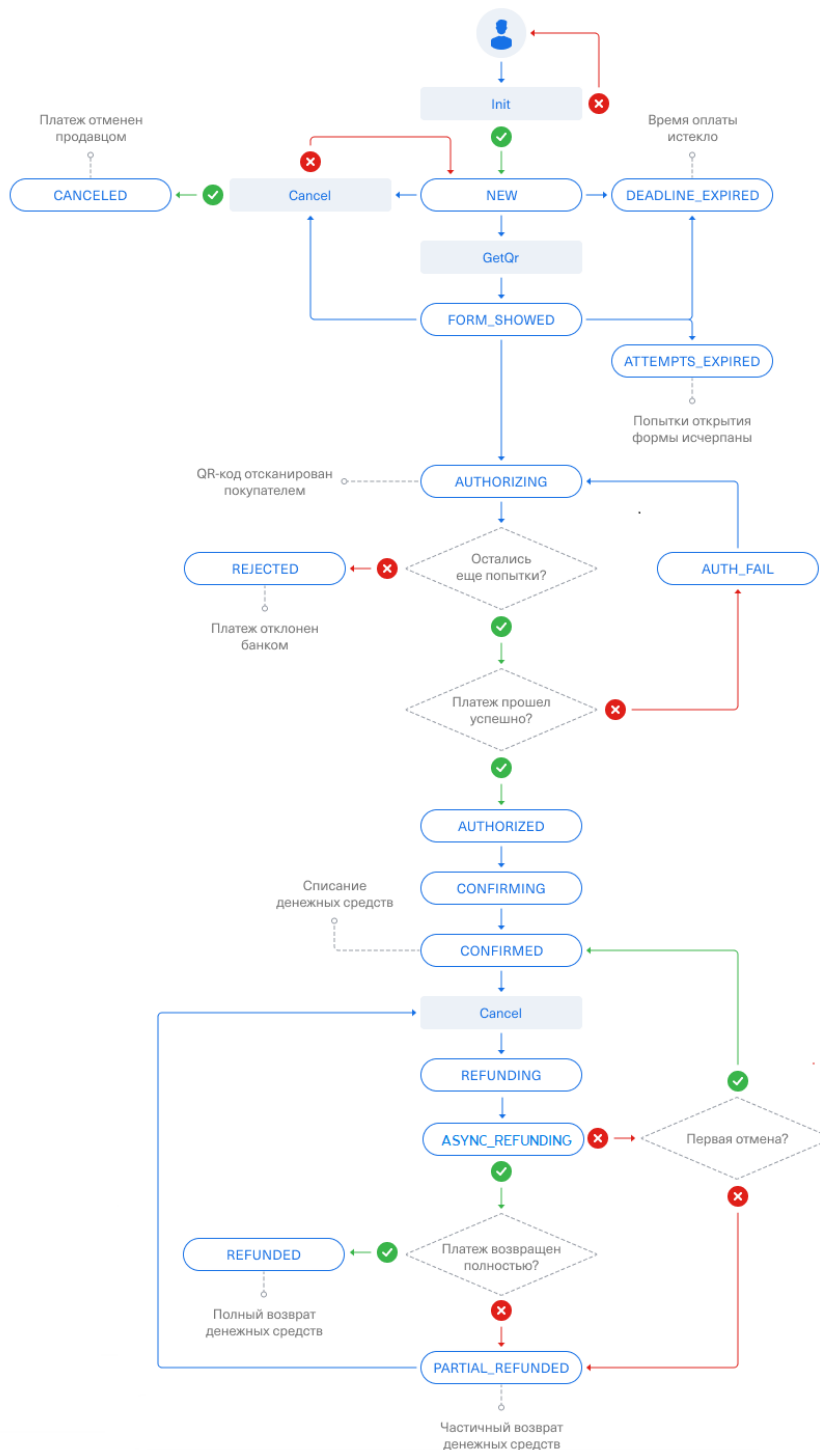
| Наименование | Тип | Обязательность | Описание |
|--------------|----------|----------------|-----------------------|
| QR | imageSVG | Нет | WebQR для Tinkoff Pay |

3. Методы оплаты по СБП

В этом разделе перечислены методы для взаимодействия с СБП.

Внимание! Тестирование оплаты через Систему быстрых платежей возможно на prod окружении и только на demo терминале. URL отправки запросов <https://securepay.tinkoff.ru/v2/>

3.1. Схема проведения платежа при оплате по QR



3.2. Метод GetQr

Описание: Регистрирует QR и возвращает информацию о нем от СБП. Должен быть вызван после вызова метода Init.

Запрос

Боевой URL: <https://securepay.tinkoff.ru/v2/GetQr>

Метод: POST

Таблица 3.2.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| DataType | String | Нет | Тип возвращаемых данных PAYLOAD – В ответе возвращается только Payload - ссылка на оплату (по умолчанию) IMAGE – В ответе возвращается SVG изображение QR |
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "PaymentId": "10063",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 3.2.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Да | Номер заказа в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |

| | | | |
|-----------|--------|-----|--|
| Data | String | Да | В зависимости от параметра DataType в запросе это: Payload - информация, которая должна быть закодирована в QR (ссылка на оплату) или SVG изображение QR, в котором уже закодирован Payload |
| PaymentId | Number | Да | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "OrderId": "21057",
  "Success": true,
  "Data": "https://qr.nspk.ru/AS1000670LSS7DN18SJQDNP4B05KLJL2?type=01&bank=100000000001&sum=10000&cur=RUB&crc=C",
  "PaymentId": 10063,
  "ErrorCode": "O"
}
```

3.3. Метод QrMembersList

Описание: Список банков, куда может быть осуществлен возврат платежа по СБП, совершенного по QR.

Запрос

Боевой URL: <https://securepay.tinkoff.ru/v2/QrMembersList>

Метод: POST

Таблица 3.3.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "PaymentId": "10063",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 3.3.2. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|-----------------|----------------|---|
| Members | Array of Member | Нет | Массив списка участников. Возвращается только если возврат возможен |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |

Объект Member

Таблица 3.3.3. Параметры Member

| Наименование | Тип | Обязательность | Описание |
|--------------|---------|----------------|--|
| MemberId | String | Да | Банк участник |
| MemberName | String | Да | Наименование банка участника |
| IsPayee | Boolean | Да | true - если данный участник был получателем указанного платежа, false - в противном случае |

Пример ответа:

```
{
  "Members": [
    {
      "MemberId": "10000000",
      "MemberName": "АО\Тинькофф Банк\"",
      "IsPayee": true
    },
    {
      "MemberId": "10000000",
      "MemberName": "ПАО\Сбербанк\"",

```

```

    "IsPayee": false
  }
],
"Success": true,
"ErrorCode": "0",
"Message": "OK"
}

```

3.4. Метод AddAccountQr

Описание: Иницирует привязку счета покупателя к магазину в СБП и возвращает информацию о нём. В ответ будет отправлена нотификация о привязке счета (см. [п. 4.5 Нотификации о статусе привязки счета](#)).

Запрос

Боевой URL: <https://securepay.tinkoff.ru/v2/AddAccountQr>

Метод: POST

Таблица 3.4.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала |
| Description | String | Да | Подробное описание деталей заказа |
| DataType | String | Нет | Тип возвращаемых данных PAYLOAD – В ответе возвращается только Payload (по-умолчанию) IMAGE – В ответе возвращается SVG изображение QR |
| DATA | Object | Нет | JSON объект, содержащий дополнительные параметры в виде “ключ:” значение”. Данные параметры будут переданы на страницу оплаты (в случае ее кастомизации). Максимальная длина для каждого передаваемого параметра: Ключ – 20 знаков; Значение – 100 знаков. Максимальное количество пар «ключ-значение» не может превышать 20 |

| | | | |
|-----------------|----------|-----|---|
| RedirectDueDate | Datetime | Нет | Срок жизни ссылки или динамического QR-кода СБП (если выбран данный способ оплаты)*. Если текущая дата превышает дату, переданную в данном параметре, ссылка для оплаты или возможность платежа по QR-коду становятся недоступными и платёж выполнить нельзя. Максимальное значение: 90 дней от текущей даты. Минимальное значение: 1 минута от текущей даты. Формат даты: YYYY-MM-DDTHH24:MI:SS+GMT Пример даты: 2016-08-31T12:28:00+03:00 |
| Token | String | Да | Подпись запроса |

*В случае, если параметр RedirectDueDate не был передан, проверяется настроечный параметр платежного терминала REDIRECT_TIMEOUT, который может содержать значение срока жизни ссылки в часах. Если его значение больше нуля, то оно будет установлено в качестве срока жизни ссылки или динамического QR-кода. Иначе, устанавливается значение «по умолчанию» - 1440 мин. (1 сутки).

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "RedirectDueDate": "2016-08-31T12:28:00+03:00",
  "DataType": "payload",
  "Description": "Подписка в пользу АО \"Кофеек\", регулярность раз в месяц, сумма 100 рублей",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

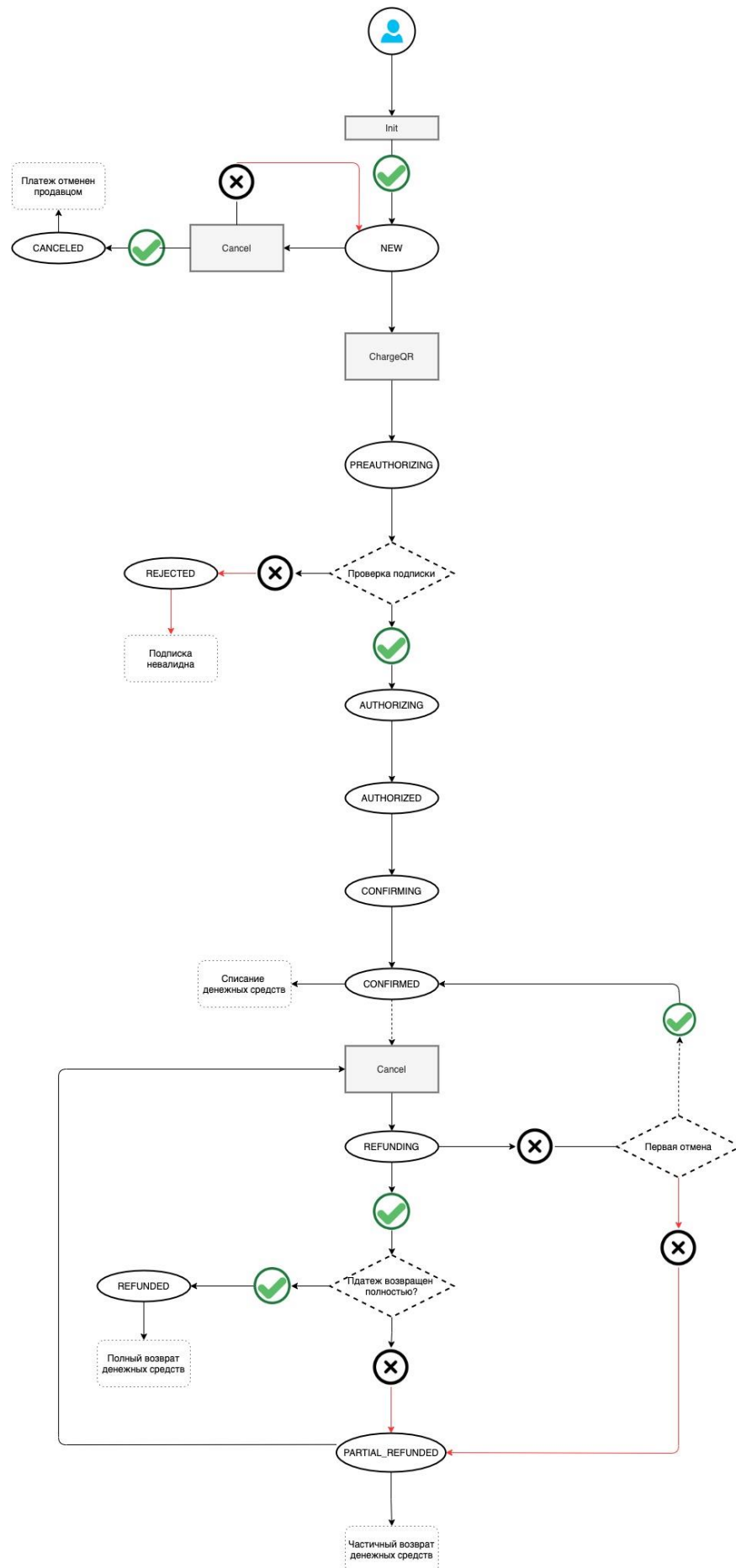
Таблица 3.4.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| TerminalKey | String | Да | Идентификатор терминала |
| Data | String | Да | В зависимости от параметра DataType в запросе это: Payload - информация, которая должна быть закодирована в QR или SVG изображение QR в котором уже закодирован Payload |

| | | | |
|------------|--------|-----|--|
| RequestKey | String | Да | Идентификатор запроса на привязку счета |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно, Код ошибки, «3013» - "Рекуррентные платежи недоступны", Код ошибки, «3001» - "Оплата через QrPay недоступна" |
| Message | String | Нет | Краткое описание ошибки |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "RequestKey": "AB1000670LSS7DN18SJQDNP4B05KLJL2",
  "Data": "https://qr.nspk.ru/AS1000670LSS7DN18SJQDNP4B05KLJL2?type=01&bank=100000000001&sum=10000&cur=RUB&crc=C08B",
  "Success": true,
  "Message": "OK",
  "ErrorCode": "O"
}
```

3.5. Метод ChargeQr

Проведение платежа по привязанному счету по QR через СБП.

Для возможности его использования Покупатель должен совершить успешную привязку счета с помощью метода AddAccountQr. После вызова метода будет отправлена нотификация на Notification URL о привязке счета (см. [п. 4.5 Нотификации о статусе привязки счета](#)), в которой будет указан AccountToken.

Для совершения платежа по привязанному счету Продавец должен вызвать метод Init, в котором поля Recurrent= Y и DATA= {"QR": "true"}, а затем вызвать метод ChargeQr для оплаты по тем же самым реквизитам и передать параметр AccountToken, полученный после привязки счета по QR в нотификации.

ChargeQr выполняет внутренние проверки и возвращает промежуточный результат.

Негативный - отказ на стороне Банка-эквайера, для уточнения причины см. [код ошибки](#).

Позитивный - платеж передан в обработку НСПК и Банка-эмитента.

Конечным результатом обработки будет изменение статуса на CONFIRMED или CANCELED, который можно получить методом GetState или в автоматически отправляемой нотификации.

Запрос

Боевой URL: <https://securepay.tinkoff.ru/v2/ChargeQr>

Метод: POST

Таблица 3.5.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|---|--|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| AccountToken | String | Да | Идентификатор привязки счета, назначаемый Банком Плательщика |
| Token | String | Да | Подпись запроса |
| IP | String | Нет | IP-адрес клиента |
| SendEmail | bool | Нет | true – если покупатель хочет получать уведомления на почту |
| InfoEmail | String | Нет (Обязателен при передаче SendEmail) | Адрес почты покупателя |

Пример запроса:

```
{
```

```

"TerminalKey": "TinkoffBankTest",
"PaymentId": "10063",
"Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
"AccountToken": "70LSS7DN18SJQRS10006DNPKLJL24B05"
}

```

Ответ

Формат ответа: JSON

Таблица 3.5.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Да | Номер заказа в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| Status | String | Да | Статус транзакции |
| Amount | Number | Да | Сумма списания в копейках |
| Currency | Number | Да | Код валюты ISO 421. |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Да | Код ошибки, «0» - если успешно, Код ошибки, «3013» - "Рекуррентные платежи недоступны", Код ошибки, «3015» - "Неверный статус AccountToken" |
| Message | String | Нет | Краткое описание ошибки |

Пример ответа:

```

{
  "TerminalKey": "TinkoffBankTest",
  "OrderId": "21057",
  "Success": true,
  "Status": "OK",
  "Amount": 100,
  "Currency": "643",

```

```
"PaymentId": "10063",
"ErrorCode": "O"
}
```

3.6. Метод GetAddAccountQrState

Описание: возвращает статус привязки счета Покупателя по магазину

Таблица 3.6.1. Статусы привязки счета

| Наименование | Описание |
|--------------|---|
| NEW | получен запрос на привязку счёту |
| PROCESSING | в обработке |
| ACTIVE | привязка счета успешна |
| INACTIVE | привязка счета неуспешна/деактивирована |

Запрос

Боевой URL: <https://securepay.tinkoff.ru/v2/GetAddAccountQrState>

Метод: POST

Таблица 3.6.2. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| RequestKey | String | Да | Идентификатор запроса на привязку счета |
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6",
  "RequestKey": "AB1000670LSS7DN18SJQDNP4BO5KLJL2",
}
```

Ответ

Формат ответа: JSON

Таблица 3.6.3. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|----------------|--------|----------------|--|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| RequestKey | String | Да | Идентификатор запроса на привязку счета |
| Status | String | Да | Статус привязки |
| BankMemberId | String | Нет | Идентификатор Банка Плательщика, который будет совершать оплату по привязанному счету - заполнен, если статус ACTIVE, INACTIVE |
| BankMemberName | String | Нет | Наименование Банка, заполнен если BankMemberId передан |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно, Код ошибки, «3012» - «Привязка счета не найдена» |
| Message | String | Нет | Краткое описание ошибки |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "RequestKey": "AB1000670LSS7DN18SJQDNP4B05KLJL2",
  "Status": "NEW",
  "Success": true,
  "Message": "OK",
  "ErrorCode": "O"
}
```

3.7. Метод GetAccountQrList

Описание: Возвращает список привязанных счетов покупателей по магазину

Запрос

Боевой URL: <https://securepay.tinkoff.ru/v2/GetAccountQrList>

Метод: POST

Таблица 3.7.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|-------------------------|
| TerminalKey | String | Да | Идентификатор терминала |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|-----------------|
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: Массив JSON

Таблица 3.7.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|----------------|--------|----------------|---|
| RequestKey | String | Да | Идентификатор запроса на привязку счета |
| Status | String | Да | Статус привязки |
| AccountToken | String | Нет | Идентификатор привязки счета, назначаемый Банком Плательщика |
| BankMemberId | String | Нет | Идентификатор Банка Плательщика, который будет совершать оплату по привязаному счету - заполнен, если статус ACTIVE, INACTIVE |
| BankMemberName | String | Нет | Наименование Банка, заполнен если BankMemberId передан |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |

Пример ответа:

```
{
  "AccountTokens": [
    {
      "RequestKey": "AB1000670LSS7DN18SJQDNP4B05KLJL2",
      "Status": "ACTIVE",
      "AccountToken": "70LSS7DN18SJQRS10006DNP4B05KLJL24B05",
      "BankMemberId": "1000000000004",
      "BankMemberName": "Промсвязьбанк"
    }
  ],
}
```

```
{
  "RequestKey": "AB1000670LSS7DN18SJQDNP4B05KLJL3",
  "Status": "ACTIVE",
  "AccountToken": "70LSS7DN18SJQRS10006DNP4B05KLJL24B33",
  "BankMemberId": "1000000000004",
  "BankMemberName": "Промсвязьбанк"
},
{
  "TerminalKey": "TinkoffBankTest",
  "Success": true,
  "ErrorCode": "O",
  "Message": "OK"
}
```

3.8. Метод GetQRState

Описание: Возвращает статус возврата платежа по СБП.

Запрос

Боевой URL: <https://securepay.tinkoff.ru/v2/GetQrState>

Метод: POST

Таблица 3.8.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка, полученный в ответе на вызов метода Init |
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "PaymentId": "2304882",
  "Token": "c0ad1dfc4e94ed44715c5ed0e84f8ec439695b9ac219a7a19555a075a3c3ed24"
}
```

Ответ

Формат ответа: JSON

Таблица 3.8.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|------|----------------|---|
| Success | Bool | Да | Успешность прохождения запроса (true/false) |

| | | | |
|-----------------|--------|--|---|
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Status | String | Да, если не произошло ошибки при получении статуса | Статус платежа |
| QrCancelCode | String | Нет | Код ошибки возврата, полученный от СБП |
| QrCancelMessage | String | Нет | Дополнительное описание ошибки, произошедшей при возврате по QR |
| OrderId | String | Нет | Номер заказа в системе Продавца |
| Amount | Number | Нет | Сумма отмены в копейках |
| Message | String | Нет | Краткое описание ошибки, произошедшей при запросе статуса |

Пример ответа:

```
{
  "Status": "CONFIRMED",
  "QrCancelCode": "IO5043",
  "QrCancelMessage": "У покупателя нет расчетного счета в этом банке. Попробуйте вернуть деньги на счет этого покупателя в другом б",
  "Success": true,
  "ErrorCode": "O",
  "Message": "OK"
}
```

3.9 Метод SbpPayTest

Описание: Тестовая платежная сессия с предопределенным статусом по СБП.

Запрос

Боевой URL *: <https://securepay.tinkoff.ru/v2/SbpPayTest>

*отправка запросов осуществляется с использованием терминалов с приставкой DEMO

Метод: POST

Таблица 3.9.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| PaymentId | String | Да | Уникальный идентификатор транзакции в системе Банка |

| Наименование | Тип | Обязательность | Описание |
|-------------------|---------|----------------|--|
| TerminalKey | String | Да | Идентификатор терминала |
| Token | String | Да | Подпись запроса |
| IsDeadlineExpired | Boolean | Нет | <p>Признак эмуляции отказа проведения платежа Банком по таймауту. По умолчанию не используется (эмуляция не требуется).</p> <ul style="list-style-type: none"> • false – эмуляция не требуется • true – требуется эмуляция (не может быть использован вместе с IsRejected = true) |
| IsRejected | Boolean | Нет | <p>Признак эмуляции отказа Банка в проведении платежа</p> <p>По умолчанию не используется (эмуляция не требуется)</p> <ul style="list-style-type: none"> • false – эмуляция не требуется • true – требуется эмуляция (не может быть использован вместе с IsDeadlineExpired = true) |

Пример запроса:

```
{
  "PaymentId": "10063",
  "TerminalKey": "TinkoffBankTest",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 3.9.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|---------|----------------|---|
| Success | Boolean | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |

Пример ответа:

```
{  
  "Success": true,  
  "ErrorCode": "O",  
  "Message": "ОК"  
}
```

3.10. Тесты по QR

3.10.1 Сценарий “Платеж-успех”

- 1) Инициировать начало платежной сессии – вызывать метод **Init**.
- 2) Запросить формирование Динамического QR-кода **GetQr**.
- 3) Отобразить Динамический QR-код на странице Покупателю.
- 4) Вызвать метод **SbpPayTest**, передавая в нем внутренний идентификатор платежной сессии Банка (PaymentId).
- 5) Запросить текущий статус платежа вызывая метод **GetState**.
- 6) Получить ответ со статусом CONFIRMED.

3.10.2 Сценарий “Платеж - отказ по таймауту”

- 1) Инициировать начало платежной сессии – вызывать метод **Init**.
- 2) Запросить формирование Динамического QR-кода **GetQr**.
- 3) Отобразить Динамический QR-код на странице Покупателю.
- 4) Вызвать метод **SbpPayTest**, передавая в нем внутренний идентификатор платежной сессии Банка (PaymentId) и параметр IsDeadlineExpired = true.
- 5) Запросить текущий статус платежа вызывая метод **GetState**.
- 6) Получить ответ со статусом DEADLINE_EXPIRED.

3.10.3 Сценарий “Платеж – отказ, отклонен Банком”

- 1) Инициировать начало платежной сессии – вызывать метод **Init**.
- 2) Запросить формирование Динамического QR-кода **GetQr**.
- 3) Отобразить Динамический QR-код на странице Покупателю.
- 4) Вызвать метод **SbpPayTest**, передавая в нем внутренний идентификатор платежной сессии Банка (PaymentId) и параметр IsRejected = true.
- 5) Запросить текущий статус платежа вызывая метод **GetState**.
- 6) Получить ответ со статусом REJECTED.

3.10.4 Сценарий “Возврат – успех

- 1) Инициировать возврат (не отмену) методом **Cancel** тестового платежа по QR-коду СБП, выполненного успешно в тесте 3.10.1
- 2) Запросить текущий статус платежа вызывая метод **GetState**.
- 3) Получить ответ со статусом REFUNDED.

4. Нотификация продавца об операциях

Нотификации – это уведомления магазину о статусе выполнения платежа. На основании этих уведомлений магазин должен предоставлять услугу/товар покупателю.

4.1. Нотификации по электронной почте

Тинькофф Оплата может присылать письма с уведомлениями об успешных платежах. Настроить нотификации на электронную почту можно в личном кабинете. Уведомления на почту можно комбинировать с уведомлениями, отправляемыми по http(s).

4.2. Нотификации по http(s)

Описание: Тинькофф Оплата может уведомлять магазин об успешных/ошибочных платежах и изменении статуса платежа. Для этого в настройках терминала необходимо указать URL, на который будут отправляться POST-запросы со статусами.

При совершении операций Authorize, FinishAuthorize, Confirm, Cancel на адрес Notification URL высылается уведомление POST-запросом с информацией об операции. При использовании одностадийного проведения платежа при обращении к методу FinishAuthorize нотификация отправляется на сайт Продавца на адрес Notification URL синхронно и ожидает ответа в течение 10 секунд. После получения ответа или неполучения его за заданное время сервис переадресует Покупателя на Success URL или Fail URL в зависимости от результата платежа.

В случае успешной обработки нотификации Продавец должен вернуть ответ с телом сообщения: OK (без тегов и заглавными английскими буквами).

Если тело сообщения отлично от OK, любая нотификация считается неуспешной, и сервис будет повторно отправлять нотификацию раз в час в течение 24 часов. Если нотификация за это время так и не доставлена, она складывается в дампы.

Вышесказанное так же действительно и при вызове метода Charge за исключением того, что данный метод не осуществляет переадресации Покупателя.

Если в NotificationURL используются порты, допустимо использование 443 порта (HTTPS).

Актуальный список внешних сетей*, используемых Тинькофф Оплата, для отправки нотификаций:

91.194.226.0/23

91.218.132.0/22

212.233.80.0/22

*Для корректной работы нотификаций необходимо добавить данные сети в исключения сетевых фильтров или других видов защиты в случае их использования.

URL: Notification URL

Метод: POST

Таблица 4.2.1. Параметры нотификации

| Наименование | Тип | Описание |
|--------------|--------|---|
| TerminalKey | String | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Номер заказа в системе Продавца |

| Наименование | Тип | Описание |
|--------------|--------|--|
| Success | bool | Успешность прохождения запроса (true/false) |
| Status | String | Статус транзакции |
| PaymentId | String | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Код ошибки, «О» - если успешно |
| Amount | Number | Текущая сумма транзакции в копейках |
| CardId | Number | Если разрешена автоматическая привязка карт Покупателей к терминалу, и при вызове метода Init был передан параметр CustomerKey - в этот параметр будет передан идентификатор привязанной карты |
| Pan | String | Замаскированный номер карты/Замаскированный номер телефона |
| ExpDate | String | Срок действия карты |
| RebillId | String | Если при вызове метода Init платеж был помечен как рекуррентный - после подтверждения оплаты в этот параметр будет передан идентификатор рекуррентного платежа |
| Token | String | Подпись запроса. Формируется по такому же принципу, как и в случае запросов в банк |
| DATA | String | Дополнительные параметры платежа, переданные при создании заказа** |

* Параметры, передаваемые в нотификации, могут изменяться.

** для платежей «в Рассрочку» обязательно передаются параметры

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|------------------------------------|
| Route | Enum | Нет | Способ платежа (TCB) |
| Source | Enum | Нет | Источник платежа (Installment) |
| CreditAmount | Number | Нет | Сумма выданного кредита в копейках |

Рекомендуется проверять данные, полученные в нотификации:

1. Сеть, с которой будут приходить уведомления: 91.218.132.1, 91.218.133.1 и из списка приведенного выше.
2. Проверить Token.
3. Проверить сумму платежа.
4. Проверить идентификатор терминала.
5. Проверить номер заказа в системе Продавца.

Таблица 4.2.2. Статусы платежей, по которым приходят http(s)-нотификации

| Status | Описание |
|------------------|---|
| AUTHORIZED | Деньги захолдированы на карте клиента. Ожидается подтверждение операции* |
| CONFIRMED | Операция подтверждена |
| REVERSED | Операция отменена |
| REFUNDED | Произведён возврат |
| PARTIAL_REFUNDED | Произведён частичный возврат |
| REJECTED | Списание денежных средств закончилась ошибкой |
| 3DS_CHECKING | Автоматическое закрытие сессии, которая превысила срок пребывания в статусе 3DS_CHECKING (более 36 часов)** |

* Операция может быть подтверждена:

- Через Личный Кабинет;
- Запросом Confirm;
- Автоматически, если у магазина настроена одностадийная схема оплаты для магазина в Личном Кабинете.

По неподтвержденным операциям возмещение не производится. Узнать статус платежа можно с помощью вызова метода GetState.

**Для получения нотификаций по 3DS_CHECKING клиент должен реализовать на своей стороне возможность получения нотификаций со статусом DEADLINE_EXPIRED, а также написать на почту acq_help@tinkoff.ru с просьбой включить отправку нотификаций об автозакрытии сессий в статусе 3DS_CHECKING.

Пример http(s)-нотификации:

```
{
  "TerminalKey": "TinkoffBankTest",
  "OrderId": "test2",
  "Success": true,
  "Status": "CONFIRMED",
  "PaymentId": "2006896",
  "ErrorCode": "O",
  "Amount": 102120,
  "CardId": 867911,
  "Pan": "430000**0777",
  "ExpDate": "1122",
  "Token": "d0815e288f121255d5d6b77831fb486cc5e9f91914a3f58a99b6118b54676d84"
}
```

Ответ на HTTP(s)-нотификацию:

В случае успешной обработки нотификации Продавцу необходимо вернуть ответ HTTP CODE = 200 и с телом сообщения: ОК (без тегов и заглавными английскими буквами).

PHP. Пример ответа на http(s)-нотификацию

```
<?php
echo «ОК»;
?>
```

Java. Пример ответа на http(s)-нотификацию

```
@POST
@Path("/ok")
public Response NotifyResponse() {
    return Response.status(200).entity("OK").build();
}
```

Если ответ «ОК» не получен, нотификация считается неуспешной, и сервис будет повторно отправлять данную нотификацию раз в час в течение 24 часов.

Если нотификация за это время не доставлена, она будет сложена в архив.

При получении нотификации и перед её обработкой настоятельно рекомендуем проверить подпись.

Проверка токенов:

Для формирования подписи запроса для нотификации необходимо:

1. Собрать массив всех передаваемых параметров в виде пар Ключ-Значение (кроме параметра Token).

Например:

```
[["CardHolder","IVAN
IVANOV"],["TerminalKey","TinkoffBankTest"],["OrderId","201709"],["Success","true"],["Status","AUTHORIZED"],["PaymentId","8742591"],["ErrorCode","0"],["Amount","9855"],["CardId","322264"],["Pan","430000*****0777"],["ExpDate","1122"],["RebillId","101709"]]
```

2. Добавить в массив пару (Password, значение). Password – пароль для терминала, указан в Личном кабинете <https://oplata.tinkoff.ru>:

```
[["CardHolder","IVAN
IVANOV"],["TerminalKey","TinkoffBankTest"],["OrderId","201709"],["Success","true"],["Status","AUTHORIZED"],["PaymentId","8742591"],["ErrorCode","0"],["Amount","9855"],["CardId","322264"],["Pan","430000*****0777"],["ExpDate","1122"],["RebillId","101709"],["Password","Dfsfh56dgKI"]]
```

3. Отсортировать массив по Ключам по алфавиту:

```
[["Amount","9855"],["CardId","322264"],["CardHolder","IVANIVANOV"],["ErrorCode","0"],["ExpDate","1122"],["OrderId","201709"],["Pan","430000*****0777"],["Password","Dfsfh56dgKI"],["PaymentId","8742591"],["RebillId","101709"],["Status","AUTHORIZED"],["Success","true"],["TerminalKey","TinkoffBankTest"]]
```

4. Конкатенировать значения всех пар:

```
9855IVAN
IVANOV32226401122201709430000*****0777Dfsfh56dgKI8742591101709AUTHORIZEDtrue1321054611234DEMO
```

5. Вычислить SHA-256 от полученного в п.4. значения:

```
f481cbb9b3e92291209386d183b8a03e811c77d80216c20b298632a1208e0498
```

Пример генерации токена:

```
private static final String PASSWORD_KEY = "Password"; private static final String PASSWORD_VALUE = "12345678";
private String generateToken(final Map<String, String> parameters) throws UnsupportedOperationException,
NoSuchAlgorithmException { final Map<String, String> sortedParameters = new TreeMap<String, String>(parameters);
if (sortedParameters.containsKey(TOKEN)) {
    sortedParameters.remove(TOKEN);
}
sortedParameters.put(PASSWORD_KEY, PASSWORD_VALUE); final String paramString =
Joiner.on("").skipNulls().join(sortedParameters.values()); return
calculateSha256(paramString);
}
```

Сравнение токенов

```
private boolean checkToken(final Map<String,String> params, final String expectedToken) {
    final String actualToken = params.get(TOKEN);
    return !(expectedToken == null || !expectedToken.equals(actualToken));
}
```

4.3. Нотификация о фискализации

Описание: Если используется подключенная онлайн касса, по результату фискализации будет отправлена нотификация с фискальными данными.

URL: Notification URL

Метод: POST

Таблица 4.3.1. Параметры нотификации

| Наименование | Тип | Описание |
|-------------------------|---------|--|
| TerminalKey | String | Идентификатор терминала, выдается Продавцу Банком |
| OrderId | String | Номер заказа в системе Продавца |
| Success | bool | Успешность прохождения запроса (true/false) |
| Status | String | Для нотификации о фискализации значение всегда RECEIPT |
| PaymentId | String | Уникальный идентификатор транзакции в системе Банка |
| ErrorCode | String | Код ошибки, если произошла ошибка |
| ErrorMessage | String | Описание ошибки, если произошла ошибка |
| Amount | Number | Текущая сумма транзакции в копейках |
| FiscalNumber | Integer | Номер чека в смене |
| ShiftNumber | Integer | Номер смены |
| ReceiptDatetime | Date | Дата и время документа из ФН |
| FnNumber | String | Номер ФН |
| EcrRegNumber | String | Регистрационный номер ККТ |
| FiscalDocumentNumber | Integer | Фискальный номер документа |
| FiscalDocumentAttribute | Integer | Фискальный признак документа |
| Receipt | JSON | Состав чека |
| Type | String | Признак расчета |

| Наименование | Тип | Описание |
|------------------|--------|--|
| Token | String | Подпись запроса. Формируется по такому же принципу, как и в случае запросов в банк |
| Ofd | String | Наименование оператора фискальных данных |
| Url | String | URL адрес с копией чека |
| QrCodeUrl | String | URL адрес с QR кодом для проверки чека в ФНС |
| CalculationPlace | String | Место осуществления расчетов |
| CashierName | String | Имя кассира |
| SettlePlace | String | Место нахождения (установки) ККМ |

4.4 Нотификация о статусе привязки счета

Описание: После привязки счета по QR, магазину отправляется статус привязки и токен. Нотификация будет приходить по статусам ACTIVE и INACTIVE.

Тинькофф Оплата может уведомлять магазин об успешных/ошибочных привязках счета. Для этого в настройках терминала необходимо указать URL, на который будут отправляться POST-запросы со статусами.

Таблица 4.5.1. Статусы привязки счета

| Наименование | Описание |
|--------------|---|
| ACTIVE | привязка счета успешна |
| INACTIVE | привязка счета неуспешна/деактивирована |

Запрос

URL: Notification URL

Метод: POST

Таблица 4.5.2. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|------------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| RequestKey | String | Да | Идентификатор запроса на привязку счета |
| Status | String | Да | Статус привязки |
| AccountToken | String | Нет | Идентификатор привязки счета - Token, назначаемый Банком |
| BankMemberId | String | Нет | Идентификатор Банка Плательщика, который будет совершать оплату по привязаному счету - заполнен, если статус ACTIVE |
| BankMemberName | String | Нет | Наименование Банка, заполнен если BankMemberId передан |
| NotificationType | String | Да | Тип нотификации, всегда константа «LINKACCOUNT» |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» если успешно |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| Message | String | Нет | Краткое описание ошибки |
| Token | String | Да | Подпись запроса. Формируется по такому же принципу, как и в случае запросов в банк |

Пример http(s)-нотификации:

```
{
  "TerminalKey": "TinkoffBankTest",
  "RequestKey": " AB1000670LSS7DN18SJQDNP4B05KLJL2 ",
  "Token": " 871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6 ",
  "Status": "ACTIVE",
  "AccountToken": " 70LSS7DN18SJQRS10006DNP4B05KLJL24B05",
  "BankMemberId": "1000000000004",
  "BankMemberName": "Промсвязьбанк",
  "NotificationType": "LINKACCOUNT",
  "Success": true,
  "Message": "OK",
  "ErrorCode": "O"
},
```

Ответ на HTTP(s)-нотификацию

В случае успешной обработки нотификации Продавцу необходимо вернуть ответ HTTP CODE = 200 и с телом сообщения: ОК (без тегов и заглавными английскими буквами).

PHP. Пример ответа на http(s)-нотификацию

```
<?php
echo «ОК»;
?>
```

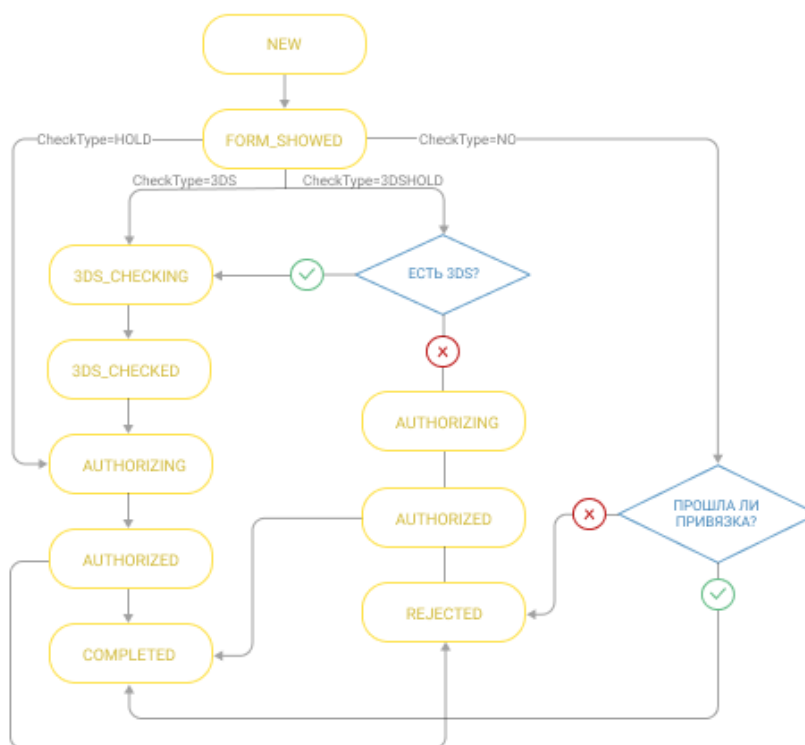
Java. Пример ответа на http(s)-нотификацию

```
{
  @POST
  @Path("/ok")
  public Response NotifyResponse() {
    return Response.status(200).entity("OK").build();
  }
}
```

Если ответ «ОК» не получен, нотификация считается неуспешной, и сервис будет повторно отправлять данную нотификацию раз в час в течение 24 часов.

Если нотификация за это время не доставлена, она будет сложена в архив.

В результате привязки карты к параметру CustomerKey будет привязана CardId.



последующего отображения Покупателю замаскированного номера карты, по которой будет совершен рекуррентный платеж.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/AddCustomer>

Боевой URL: <https://securepay.tinkoff.ru/v2/AddCustomer>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 5.1.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| CustomerKey | String | Да | Идентификатор покупателя в системе Продавца |
| Token | String | Да | Подпись запроса |
| IP | String | Нет | IP-адрес запроса |
| Email | String | Нет | Email клиента |
| Phone | String | Нет | Телефон клиента |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "CustomerKey": "Customer 1",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 5.1.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Платежный ключ, выдается Продавцу при заведении терминала |
| CustomerKey | String | Да | Идентификатор покупателя в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» – если успешно |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---------------------------|
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "CustomerKey": "Customer1",
  "Success": true,
  "ErrorCode": "O"
}
```

5.2. Метод GetCustomer

Описание: Возвращает данные покупателя, зарегистрированного в терминале Продавца.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/GetCustomer>

Боевой URL: <https://securepay.tinkoff.ru/v2/GetCustomer>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 5.2.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| CustomerKey | String | Да | Идентификатор покупателя в системе Продавца |
| IP | String | Нет | IP-адрес запроса |
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "CustomerKey": "Customer1",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 5.2.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Платежный ключ, выдается Продавцу при заведении терминала |
| CustomerKey | String | Да | Идентификатор покупателя в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - успешно |
| Email | String | Нет | Email клиента |
| Phone | String | Нет | Телефон клиента (+ 7 1 234567890) |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "CustomerKey": "Customer 1",
  "Success": true,
  "ErrorCode": "O"
}
```

5.3. Метод RemoveCustomer

Описание: Удаляет сохраненные данные покупателя.

Запрос

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/RemoveCustomer>

Боевой URL: <https://securepay.tinkoff.ru/v2/RemoveCustomer>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Метод: POST

Таблица 5.3.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| CustomerKey | String | Да | Идентификатор покупателя в системе Продавца |
| Token | String | Да | Подпись запроса |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|------------------|
| IP | String | Нет | IP-адрес запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "CustomerKey": "Customer 1",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: JSON

Таблица 5.3.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Платежный ключ, выдается Продавцу при заведении терминала |
| CustomerKey | String | Да | Идентификатор покупателя в системе Продавца |
| Success | bool | Да | Успешность прохождения запроса (true/false) |
| ErrorCode | String | Да | Код ошибки, «О» - если успешно |
| Message | String | Нет | Краткое описание ошибки |
| Details | String | Нет | Подробное описание ошибки |

Пример ответа:

```
{
  "TerminalKey": "TinkoffBankTest",
  "CustomerKey": "Customer 1",
  "Success": true,
  "ErrorCode": "О"
}
```

5.4. Метод GetCardList

Описание: Возвращает список привязанных карт покупателя. В том числе показывает удаленные карты.

Тестовый URL*: <https://rest-api-test.tinkoff.ru/v2/GetCardList>

Боевой URL: <https://securepay.tinkoff.ru/v2/GetCardList>

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tinkoff.ru с просьбой добавить ваши IP в WL. После чего отправлять запросы с боевых терминалов.

Запрос

Метод: POST

Таблица 5.4.1. Параметры запроса

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|---|
| TerminalKey | String | Да | Идентификатор терминала, выдается Продавцу Банком |
| CustomerKey | String | Да | Идентификатор покупателя в системе Продавца |
| IP | String | Нет | IP-адрес запроса |
| Token | String | Да | Подпись запроса |

Пример запроса:

```
{
  "TerminalKey": "TinkoffBankTest",
  "CustomerKey": "Customer 1",
  "Token": "871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6"
}
```

Ответ

Формат ответа: Массив JSON

Таблица 5.4.2. Параметры ответа

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| CardId | String | Да | Идентификатор карты в системе Банка |
| Pan | String | Да | Номер карты 411111*****1111 |
| Status | String | Да | Статус карты: А – активная, I – не активная, D - удаленная |
| RebillId | String | Нет | Идентификатор рекуррентного платежа (см. параметр Recurrent в методе Init) |

| Наименование | Тип | Обязательность | Описание |
|--------------|--------|----------------|--|
| CardType | Enum | Да | Тип карты: <ul style="list-style-type: none"> – карта списания (0); – карта пополнения (1); – карта пополнения и списания (2). |
| ExpDate | String | Нет | Срок действия карты |

Пример ответа:

```
[
{
  "CardId": "881900",
  "Pan": "518223*****0036",
  "Status": "D",
  "RebillId": "",
  "CardType": 0,
  "ExpDate": "1122"
},
{
  "CardId": "882263",
  "Pan": "448744*****4487",
  "Status": "A",
  "RebillId": "",
  "CardType": 0,
  "ExpDate": "0619"
}
]
```

6. Коды ошибок, передаваемые на FailURL

Подробный список ошибок с описанием представлен в отдельном документе по следующей [ссылке](#).

7. Список тестовых карт

Описание: Вы можете использовать любой срок действия для тестовой карты. Можно произвести несколько тестовых привязок с разными сроками действия и потом с помощью метода GetCardList посмотреть, какие карты привязаны. Тестовые карты используются при проведении операций на тестовой среде.

Таблица 7.1 Список тестовых карт для оплаты через протокол 3ds2.0

| Поведение карты | TransStatus* | Описание | PAN |
|--|--------------|--|--|
| <u>Ошибка оплаты</u> Недостаточно средств | Нет | Нет | 22013820000000831 expDate: 12/25 cvv: 123 |
| <u>Успешная оплата</u> 3ds2.0 Frictionless Flow | Нет | AUTHENTICATION_SUCCESSFUL Успешное прохождение аутентификации без ввода пароля | 22013820000000013 expDate: 12/25 cvv: 123 |
| <u>Успешная оплата</u> 3ds2.0 Challenge Flow | C | CHALLENGE_REQUIRED Требуется полное прохождение 3DS с редиректом на acsURL. Открытие формы для ввода одноразового пароля (OTP) | 22013820000000047 expDate: 12/25 cvv: 123 Метод аутентификации на ACS: Static Passcode. Ввести верный пароль 1qwezxc |
| <u>Ошибка оплаты</u> 3ds2.0 Restricted | R | ACCOUNT_VERIFICATION_REJECTED Эмитент отклонил аутентификацию | 22013820000000005 expDate: 12/25 cvv: 123 |
| <u>Ошибка оплаты</u> Frictionless Flow Not Authenticated | N | NOT_AUTHENTICATED Карта не поддерживает 3DS | 22013820000000021 expDate: 12/25 cvv: 123 |

| Поведение карты | TransStatus* | Описание | PAN |
|---|--------------|---|--|
| <u>Успешная оплата</u> Card not Enrolled (Attempt) | A | ATTEMPTS_PROCESSING_PERFORMED Эмитент недоступен или не поддерживает 3DS v2. Платежная система разрешает провести Pay, но эмитент мог отклонить авторизацию | 2201382000000039 expDate: 12/25 cvv: 123 |

Таблица 7.2 Список тестовых карт для оплаты через протокол 3ds1.0

| Поведение карты | TranStatus* | Описание | PAN |
|--|-------------|----------|--|
| <u>Успешная оплата</u> | Нет | - | 5586200071492158 expDate: 12/25 cvv: 123 |
| <u>Ошибка оплаты</u> Недостаточно средств при сумме авторизации > 1000 рублей | Нет | - | 5586200071499591 expDate: 12/25 cvv: 123 |

Таблица 7.2 Список тестовых карт для оплаты без 3ds

| Поведение карты | TransStatus* | Описание | PAN |
|--|--------------|----------|--|
| <u>Успешная оплата</u> | Нет | - | 2200770239097761 expDate: 12/25 cvv: 123 |
| <u>Ошибка оплаты</u> Недостаточно средств | Нет | - | 4249170392197566 expDate: 12/25 cvv: 123 |

| Поведение карты | TransStatus* | Описание | PAN |
|---|--------------|----------|--|
| <u>Ошибка оплаты</u> Ошибка при списании | Нет | - | 5586200071492075 expDate: 12/25 cvv: 123 |

* Описание параметра TransStatus находится в описании параметров ответа cres (JSON/JWE cres объект)

8. Правила расчета возмещений по операционному реестру

При наличии РКО от Тинькофф Банка выплаты производятся в календарные дни. При отсутствии – в дни работы расчетно-кассового центра по графику Центробанка.

Возмещение считается за один календарный день.

Таблица 8.1 Типы операций и их влияние на подсчет реестра

| Тип операции | Пояснение | Плюс/минус |
|------------------------|---|------------|
| Debit | Операция оплаты | Плюс |
| Credit | Операция возврата | Минус |
| Fee | Комиссия по операции оплаты (в том числе неуспешной) | Минус |
| CancelRefund | Отмена возврата | Плюс |
| Chargeback | Опротестование операции эмитентом | Минус |
| 2Chargeback | Арбитражное опротестование эмитентом | Минус |
| Chargeback_Reversal | Отмена опротестования операции эмитентом | Плюс |
| 2Chargeback_Reversal | Отмена арбитражного опротестования операции эмитентом | Плюс |
| CR_Chargeback | Возврат операции Refund от эмитента (карта или договор закрыты) | Плюс |
| Representment | Обратное опротестование Chargeback Тинькофф | Плюс |
| Representment_Reversal | Отмена обратного опротестования 2Chargeback Тинькофф | Минус |
| AUTH_FAIL | Неуспешная авторизация. Сама операция в расчете не участвует. Участвует только комиссия за них. | - |
| CreditClientCorrection | FeeColl (Ручное урегулирование операции с банком-эмитентом по договоренности или при списании с Тинькофф по клирингу) | Минус |

| Тип операции | Пояснение | Плюс/минус |
|-----------------------|---|------------|
| DebitClientCorrection | FeeColl (Ручное урегулирование операции с банком-эмитентом по договоренности или при списании с Тинькофф по клирингу) | Плюс |
| CreditCorrection | Списание с ТСП претензии клиента банка Тинькофф | Минус |
| DebitCorrection | Зачисление ТСП претензии клиента банка Тинькофф | Плюс |