

ОПИСАНИЕ ПРОДУКТА БЕЗОПАСНАЯ СДЕЛКА

22.11.2021



Оглавление

ИСТОРИЯ ИЗМЕНЕНИЙ	3
ТЕРМИНЫ И СОКРАЩЕНИЯ	4
1. ОПИСАНИЕ ПРОДУКТА	5
2. НАСТРОЙКА ПОДКЛЮЧЕНИЯ	5
3. ВАРИАНТЫ ПОДКЛЮЧЕНИЯ	6
4. СТАТУСНАЯ МОДЕЛЬ СДЕЛКИ	8
4.1 Валидация сделки на стороне банка (схема 1:1)	9
4.2 Валидация сделки на стороне Сайта	10
5. АЛГОРИТМ ФОРМИРОВАНИЯ ПОДПИСИ ЗАПРОСА НА ВЫПЛАТЫ.....	11
5.1 С помощью параметра Token	11
5.2 С помощью RSA сертификата	12
5.3 С помощью сертификата КриптоПро.....	13
6. ФУНКЦИОНАЛ 1 СПИСАНИЕ: 1 ВЫПЛАТА.....	14
6.1 Для партнера, использующего форму оплаты Банка (не PCI DSS).....	14
6.2 Для партнера, использующего собственную форму оплаты (с PCI DSS).....	15
7. ФУНКЦИОНАЛ N СПИСАНИЙ : 1 ВЫПЛАТА	16
7.1. Для партнера, использующего форму оплаты Банка (не PCI DSS).....	16
7.2. Для партнера, использующего собственную форму оплаты (с PCI DSS).....	18
8. ФУНКЦИОНАЛ 1 СПИСАНИЕ : N ВЫПЛАТ	20
8.1. Для партнера, использующего форму оплаты Банка (не PCI DSS).....	20
8.2. Для партнера, использующего собственную форму оплаты (PCI DSS).....	21
9. МЕТОД CREATESPDEAL.....	24

История изменений

Версия	Описание	Дата
1.0	Документ создан	18.06.2021
1.1	Добавлен раздел 5 “Алгоритм формирования подписи запроса на выплаты”. Актуализирован п. 4 «Статусная модель Сделки» в части отмены сделки. Незначительные изменения оформления.	22.11.2021

Термины и сокращения

Термин	Определение
Банк	АО "Тинькофф Банк"
Продавец	Зарегистрированный пользователь Сайта (Приложения), размещающий там Объявления с предложением заключить Сделку в отношении Товара с использованием Сервиса
Покупатель	Зарегистрированный пользователь Сайта (Приложения), осуществляющий просмотр размещенного Продавцом Объявления, взаимодействие с Продавцом в отношении Товара, заключивший с Продавцом Сделку с использованием Сервиса.
Товар	Товар, услуга (работа), в отношении которого Продавец размещает на Сайте (Приложении) Объявление, являющийся предметом Сделки.
Объявление	Информационное сообщение с предложением о Товаре (включая контактную информацию, фотографии и любую сопутствующую информацию), размещаемое Продавцом на Сайте/Приложении, адресованное неопределенному кругу лиц.
Сайт (Приложение, Площадка)	Интернет ресурс, предоставляющий возможность размещения Продавцами Объявлений о продаже Товаров, а также предоставляющий Покупателям возможность поиска, просмотра предложений Продавцов с целью последующего приобретения Товара с использованием Сервиса.
Заказ	Запрос на Покупку размещенного Товара на Сайте/Приложении.
Покупка	Авторизация и списание средств с банковской карты Покупателя за покупку Товара.
Выплата Продавцу	Пополнение банковской карты Продавца за Покупку Товара, необходимо указать идентификатор Покупки. Осуществляется после подтверждение списания средств с банковской карты Покупателя (Покупки).
Сделка	Соглашение, заключенное между Покупателем и Продавцом, предметом которого является купля-продажа Товара с использованием функционала Сайта, и устанавливающее порядок выполнения сторонами Сделки (Покупателем и Продавцом) ее условий.

1. Описание продукта

Сервис "Безопасная сделка" позволяет организовать взаиморасчеты между двумя физическими лицами при продаже услуг или товаров. При этом, взаимодействие между физическими лицами происходит не напрямую между собой, а при участии Площадки - юридического лица, которое организует размещение Товара на своей витрине, помогает Продавцу найти покупателя, дает возможность воспользоваться сервисом банка для оплаты за товар и для получения денежных средств продавцом после реализации товара, а также организует арбитраж, в случае такой необходимости. Площадка контролирует ход выполнения сделки, и дает команду Банку на списание с карты в момент покупки, а после успешного выполнения продавцом своих обязанностей (например, после исполнения оговоренных услуг или работ, либо после успешной доставки товара), инициирует выплату на карту продавцу.

Площадка берет свою комиссию за осуществление "Безопасной сделки".

2. Настройка подключения

Для подключения продукта "Безопасная сделка" необходимо подать заявку своему персональному менеджеру или указать это в комментарии при подаче короткой заявки на oplata.tinkoff.ru. Так же, необходимо указать максимальный срок сделки (срок, после которого при отсутствии подтверждения успешного завершения сделки со стороны площадки денежные средства возвращаются на карту покупателя).

3. Варианты подключения

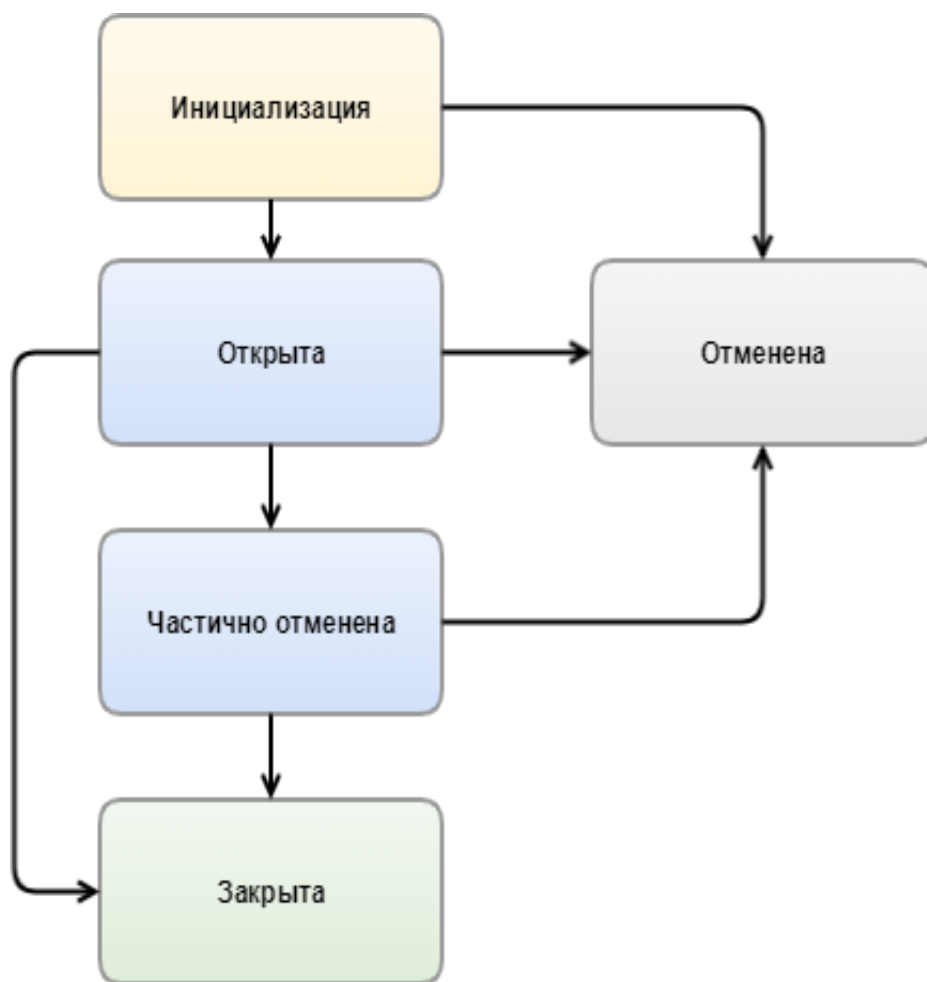
Ниже представлена матрица возможных реализаций подключений данного продукта на основании протоколов Банка, а также указание минимальных необходимых методов для реализации данного продукта.

Тип операции	Покупка		Выплата Продавцу	
Тип подключения	NOT PCI DSS Если используется стандартная платежная форма на стороне Тинькофф Банка	PCI DSS Если используется платежная форма на стороне Площадки	NOT PCI DSS Если используется стандартная платежная форма на стороне Тинькофф Банка	PCI DSS Если используется платежная форма на стороне Площадки
API	Полное описание протокола.			
	Init - инициализация Сделки на стороне банка, OrderId - идентификатор Сделки. Confirm - списание средств с банковской карты Покупателя, открытие Сделки. Cancel - частичный или полный возврат денежных средств Покупателю. Если не было списания, то денежные средства расхольдируются. Отмена/частичная отмена Сделки.	Init - создание Сделки на стороне банка. FinishAuthorize - запрос на Покупку (авторизация средств на банковской карте Покупателя). Confirm - списание средств с банковской карты Покупателя, открытие Сделки. Cancel - частичный или полный возврат денежных средств Покупателю. Если не было списания, то денежные средства расхольдируются. Отмена/частичная отмена Сделки.	AddCard - получение url на привязку карты Продавца на стороне банка. Init - проверка возможности закрытия Сделки, пополнения карты продавца. Передается OrderId ранее созданной Сделки. Payment - закрытие Сделки, запрос на пополнение карты Продавца.	Init - проверка возможности закрытия Сделки, пополнения карты продавца. Передается OrderId ранее созданной Сделки. Payment - закрытие Сделки, запрос на пополнение карты Продавца.
	поддержка Apple Pay	поддержка Apple Pay	привязка карты на стороне банка	привязка карты на стороне банка

Тип операции	Покупка		Выплата Продавцу	
Дополнительные возможности API	поддержка Google Pay привязка карты на стороне банка, рекуррентные платежи	поддержка Google Pay привязка карты на стороне банка, рекуррентные платежи	подпись запроса с помощью Token *	подпись запроса с помощью Token *

*Для выбора подписи запросов с помощью Token при подключении сообщите об этом своему персональному менеджеру или укажите это в комментарии при подаче короткой заявки на oplata.tinkoff.ru. Для переключения подписи с УНЭП на подпись запроса с помощью Token напишите на acq_help@tinkoff.ru

4. Статусная модель Сделки



Статус сделки	Какие выполнены операции	Описание этапа
Открыта	Есть успешная операция покупки (Платеж в статусе AUTHORIZED или CONFIRMED), нет других операций.	Денежные средства закодированы или списаны с карты покупателя. В реестр попадет только после операции Confirm
Частично отменена	Есть хотя бы одна операция Cancel на часть суммы сделки	Была хотя бы одна операция отмены\возврата на часть суммы сделки
Закрыта	Есть операция Payout	Денежные средства выплачены на карту продавца
Отменена	Есть одна или несколько операций Cancel, на всю сумму сделки	Сделка полностью отменяется (с закрытием аккумулятора) в тот момент, когда сумма отмен/возвратов по сделке достигает суммы оплат по сделке

4.1 Валидация сделки на стороне банка (схема 1:1)

На стороне банка реализованы следующие проверки Сделки:

- запрос на проверку возможности закрытия Сделки/запрос на завершение Сделки может быть только по ранее созданной Сделке (указан OrderId ранее созданной Сделки).
- запрос на проверку возможности закрытия Сделки/запрос на завершение Сделки может быть только по открытым или частично отмененным Сделкам (совершено списание денежных средств с банковской карты Покупателя, есть положительная сумма остатка Сделки).
- запрос на завершение Сделки может быть осуществлен только один раз в рамках одной Сделки.
- запрос на проверку возможности закрытия Сделки/запрос на завершение Сделки может быть только в пределах установленных лимитов на Сделку*.
- сумма завершения Сделки должна быть меньше либо равна сумме ее остатка.
- запрос на частичный или полный возврат денежных средств Покупателю не может быть осуществлен после закрытия Сделки.

Дополнительная логика, реализованная на стороне Банка:

- банк автоматически подтверждает списание денежных средств с карты Покупателя по истечению 4-х дней по картам ПС Visa Electron и 6-и дней по картам других ПС (MasterCard, МИР, Visa).
- Банк автоматически отменяет Заказ по истечению х-дней**, если не поступило запроса на завершение Сделки.

4.2 Валидация сделки на стороне Сайта

При реализации продукта на своей стороне рекомендуется повторить логику валидации, осуществляемую на стороне банка во избежании низкой конверсии.

* лимиты согласовываются с персональным менеджером.

** настройка на стороне Банка по требованиям Сайта. По умолчанию составляет 60 дней от даты авторизации денежных средств на карте Покупателя.

5. Алгоритм формирования подписи запроса на выплаты.

5.1 С помощью параметра Token

Инструкция по формированию Token доступна [по ссылке](#)

В запросе на выплаты не участвуют параметры X509SerialNumber, DigestValue, SignatureValue.

Пример запроса в методе Init:

```
{
  "TerminalKey": "testRegressBank",
  "Amount": "1751",
  "OrderId": "autoOrd1615285401068DELb",
  "CardId": "70000000857",
  "Token": "76956ae79e4b33e9833ab09c344c638e14d772d0e423aec4db65039622b2f1b6"
}
```

5.2 С помощью RSA сертификата

Для формирования подписи запроса необходимо:

1) Собрать массив всех передаваемых параметров в виде пар Ключ-Значение (кроме параметра DigestValue, SignatureValue, X509SerialNumber).

Например:

```
[{"TerminalKey","TestB"}, {"PaymentId","20150"}]
```

2) Сортировать по Ключам:

```
[{"PaymentId","20150"}, {"TerminalKey","TestB"}]
```

3) Конкатенировать значения:

20150TestB

4) Вычислить хэш-сумму по алгоритму SHA256 и записать значение в DigestValue (должно получиться значение в Base64).

5) Декодировать DigestValue из Base64, подписать получившееся значение с помощью RSA ключа* и записать в SignatureValue (должно получиться значение в Base64).

*Инструкция по получению RSA ключа доступна по [ссылке](#).

Ниже представлены примеры реализации работы с библиотекой:

Язык программирования	Ссылка
Java	https://acdn.tinkoff.ru/static/documents/rsa-crypto-lib-java-mapi.zip

5.3 С помощью сертификата КриптоПро

Для подписи запросов/ответов возможно пользоваться усиленной неквалифицированной электронной подписи (УНЭП) на алгоритмах ГОСТ и предоставить сертификат ключа проверки данной подписи в АО «Тинькофф Банк».

Для формирования подписи запроса необходимо:

1) Собрать массив всех передаваемых параметров в виде пар Ключ-Значение (кроме параметра DigestValue, SignatureValue, X509SerialNumber).

Например:

```
[{"TerminalKey","TestB"}, {"PaymentId","20150"}]
```

2) Сортировать по Ключам:

```
[{"PaymentId","20150"}, {"TerminalKey","TestB"}]
```

3) Конкатенировать значения:

20150TestB

4) Вычислить хэш-сумму по ГОСТ Р 34.11-2012 256 и записать значение в DigestValue (должно получиться значение в Base64).

5) Декодировать DigestValue из Base64, подписать получившееся значение по ГОСТ Р 34.10-2012 256 и записать в SignatureValue. (должно получиться значение в Base64).

Для получения УНЭП можно обратиться к вашему менеджеру по взаимодействию, написав письмо с темой «Получение УНЭП_Наименование организации».

В тексте письма указать:

Наименование системы: EACQ (тест/прод).

Цель использования сертификата в системе: подпись методов протокола Интернет-Эквайринга.

Ниже представлены примеры реализации работы с библиотекой КриптоПро (CryptoPro):

Язык программирования	Ссылка	Версия Крипто Про
C#	Поддержка 2012 ГОСТ	КриптоПРО CSP
Java	Поддержка 2012 ГОСТ	КриптоПРО JCP
PHP	Поддержка 2012 ГОСТ	КриптоПРО CSP

6. Функционал 1 списание: 1 выплата

6.1 Для партнера, использующего форму оплаты Банка (не PCI DSS)

- Площадка в первом платеже в рамках сделки вызывает метод /v2/Init. Order_id – это идентификатор сделки.
- В ответ на запрос площадка получается ссылку на форму оплаты. Оплата проходит стандартно.
- После успешного холдирования средств Покупателя Площадке отправляется http(s)-нотификация о результатах выполнения данного действия.
- После успешного оказания услуги Покупателю Площадка отправляет /v2/confirm для списания средств с Покупателя. Запрос формируется стандартно без дополнительных параметров в контексте Сделки.
- Для выплаты продавцу площадка вызывает метод /e2c/Init, передав в него order_id ранее созданной сделки. В метод Payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init.
- В метод Payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init

6.2 Для партнера, использующего собственную форму оплаты (с PCI DSS)

- Площадка в первом платеже в рамках сделки вызывает метод /v2/Init. Order_id – это идентификатор сделки.
- Далее необходимо вызвать метод /v2/FinishAuthorize, передав карточные данные покупателя.
- После успешного холдирования средств Покупателя Площадке отправляется http(s)-нотификация о результатах выполнения данного действия.
- После успешного оказания услуги Покупателю Площадка отправляет /v2/confirm для списания средств с Покупателя. Запрос формируется стандартно без дополнительных параметров в контексте Сделки.
- Для выплаты продавцу площадка вызывает метод /e2c/Init, передав в него order_id ранее созданной сделки. В метод Payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init.
- В метод Payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init.

7. Функционал N списаний : 1 выплата

7.1. Для партнера, использующего форму оплаты Банка (не PCI DSS)

- Площадка в первом платеже в рамках сделки в методе /v2/Init помимо параметров платежа дополнительно передает в секции DATA параметр StartSpAccumulation со значением true.
- В ответ на запрос площадка получается ссылку на форму оплаты. Оплата проходит стандартно.
- После успешного холдирования средств Покупателя Площадке отправляется http(s)-нотификация о результатах выполнения данного действия. Нотификация будет содержать дополнительный параметр идентификатора сделки – SpAccumulationId. Пример см. ниже.
- После успешного оказания услуги Покупателю Площадка отправляет /v2/confirm для списания средств с Покупателя. Запрос формируется стандартно без дополнительных параметров в контексте Сделки.
- Площадка в последующие платежи в метод /v2/Init передает в DATA параметр SpAccumulationId, полученный в первом платеже, и параметр StartSpAccumulation со значением false. После успешного холдирования средств Покупателя Площадке отправляется http(s)-нотификация с тем же SpAccumulationId, который был сформирован в ранее. По всем платежам с переданным параметром SpAccumulationId идет накопление суммы для выплаты (в накопление входит первый платеж). Выплата продавцу не производится. Ограничений по количеству платежей для накопления суммы выплат нет. Одна площадка может инициировать сразу несколько накоплений.
- Для выплаты продавцу площадка в метод /e2c/Init передает необходимую сумму, которая не должна превышать сумму списаний по платежам и в объект DATA передать параметр SpAccumulationId. В метод Payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init.
- Для формирования нового идентификатора для накопления суммы площадка в методе Init передает в DATA параметр StartSpAccumulation=true и не передает параметр SpAccumulationId.

Пример передачи параметров в объекте Data: "DATA":

```
{  
  "StartSpAccumulation": "false",  
  "SpAccumulationId": "756439112"  
}
```

Наименование	Тип	Обязательность	Описание
StartSpAccumulation	Boolean	Нет	Признак управления накоплением: если передано значение true, то начинается накопление суммы для выплаты продавцу
SpAccumulationId	String	Нет	Идентификатор созданного накопителя

7.2. Для партнера, использующего собственную форму оплаты (с PCI DSS)

- Площадка в первом платеже в методе /v2/Init передает в DATA параметр StartSpAccumulation=true для начала накопления суммы выплаты продавцу. В ответе метода /v2/FinishAuthorize и в http(s)-нотификации возвращается параметр SpAccumulationId – идентификатор накопителя. Если по операции проводилась проверка 3DS, то параметр SpAccumulationId вернется в ответе метода Submit3DSAuthorization. Выплата продавцу не производится.
- Площадка в последующие платежи в метод /v2/Init передает в DATA параметр SpAccumulationId, полученный в первом платеже в /v2/FinishAuthorize, и параметр StartSpAccumulation=false. В ответе /v2/FinishAuthorize и http(s)-нотификации по этим платежам возвращается тот же SpAccumulationId, который был сформирован в п. 1. По всем платежам с переданным параметром SpAccumulationId идет накопление суммы для выплаты (в накопление входит первый платеж). Выплата продавцу не производится. Ограничений по количеству платежей для накопления суммы выплат нет. Одна площадка может инициировать сразу несколько накоплений.
- Для выплаты продавцу площадка в метод /e2c/Init передает необходимую сумму, которая не должна превышать сумму списаний по платежам и в объект DATA передать параметр SpAccumulationId. В метод Payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init.
- Для формирования нового идентификатора для накопления суммы площадка в методе Init передает в DATA параметр StartSpAccumulation=true и не передает параметр SpAccumulationId.

Пример передачи параметров в объекте Data: "DATA":

```
{  
  "StartSpAccumulation": "false",  
  "SpAccumulationId": "756439112"  
}
```

Наименование	Тип	Обязательность	Описание
StartSpAccumulation	Boolean	Нет	Признак управления накоплением: если передано значение true, то начинается накопление суммы для выплаты продавцу
SpAccumulationId	String	Нет	Идентификатор созданного накопителя

Пример http(s)-нотификации с параметром SpAccumulationId:

```
{
  «TerminalKey»:«1510572937960»,
  «OrderId»:«test2»,
  «Success»:true,
  «Status»:«CONFIRMED»,
  «PaymentId»:2006896,
  «ErrorCode»:«0»,
  «Amount»:102120,
  «CardId»:867911,
  «Pan»:«430000**0777»,
  «ExpDate»:«1122»,
  «SpAccumulationId»: 756439112,
  «Token»:«d0815e288f121255d5d6b77831fb486cc5e9f91914a3f58a99b6118b54676d84»
}
```

Параметр SpAccumulationId учитывается при формировании Token в нотификации.

8. Функционал 1 списание : N выплат

8.1. Для партнера, использующего форму оплаты Банка (не PCI DSS)

- Площадка в первом платеже в рамках сделки в методе /v2/Init помимо параметров платежа дополнительно передает в секции DATA параметр StartSpAccumulation со значением '1N'
- В ответ на запрос площадка получает ссылку на форму оплаты. Оплата проходит стандартно.
- После успешного холдирования средств Покупателя Площадке отправляется http(s)-нотификация о результатах выполнения данного действия. Нотификация будет содержать дополнительный параметр идентификатора сделки – SpAccumulationId. Пример см. ниже.
- После успешного оказания услуги Покупателю Площадка отправляет /v2/confirm для списания средств с Покупателя. Запрос формируется стандартно без дополнительных параметров в контексте Сделки.
- Для выплаты продавцу площадка в метод /e2c/Init передает необходимую сумму, которая не должна превышать сумму списаний по платежам и в объект DATA передать параметр SpAccumulationId. Параметр StartSpAccumulation не передается. Если выплата является последней в сделке, в секции DATA должен быть также передан параметр SpFinalPayout со значением true. Параметр OrderId выплаты может быть любой и не обязан совпадать со значением OrderId в операции списания в Сделке (как в сделках 1 к 1).
- В метод /e2c/payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init
- Для повторных выплат, повторяются пп. 5-6. Повторные выплаты возможны в рамках остатка на балансе сделки. Повторные выплаты возможны только, если выполнены оба условия одновременно:
 - а. в предыдущих выплатах не был передан параметр SpFinalPayout со значением true,
 - б. не был достигнут нулевой баланс сделки.

После передачи параметра SpFinalPayout со значением true, либо достижения нулевого баланса сделка закрывается.

- Для открытия новой сделки Площадка в методе /v2/Init передает в секции DATA параметр StartSpAccumulation='1N' и не передает параметр SpAccumulationId. Одновременно может быть открыто любое количество сделок.

8.2. Для партнера, использующего собственную форму оплаты (PCI DSS)

- Площадка в первом платеже в рамках сделки в методе /v2/Init помимо параметров платежа дополнительно передает в секции DATA параметр StartSpAccumulation со значением '1N'
- Для холдирования средств Площадка вызывает метод /v2/FinishAuthorize согласно протоколу без каких-либо дополнительных параметров, в контексте Сделки.
- После холдирования средств Площадке будет возвращен идентификатор сделки SpAccumulationId.
- Если платеж проводится без проверки 3DS, то идентификатор сделки SpAccumulationId возвращается в ответе метода /v2/FinishAuthorize.
- Если платеж проводится с проверкой 3DS, то идентификатор сделки SpAccumulationId возвращается в ответе метода v2/Submit3DSAuthorization для 3DS 1.0 или в ответе метода /v2/Submit3DSAuthorizationV2 для 3DS 2.0
- Также параметр SpAccumulationId возвращается в http(s)-нотификациях соответствующих запросов.
- После успешного оказания услуги Покупателю Площадка отправляет /v2/confirm для списания средств с Покупателя. Запрос формируется стандартно без дополнительных параметров в контексте Сделки.
- Для выплаты продавцу площадка в метод /e2c/Init передает необходимую сумму, которая не должна превышать сумму списаний по платежам и в объект DATA передать параметр SpAccumulationId. Параметр StartSpAccumulation не передается. Если выплата является последней в сделке, в секции DATA должен быть также передан параметр SpFinalPayout со значением true. Параметр OrderId выплаты может быть любой и не обязан совпадать со значением OrderId в операции списания в Сделке (как в сделках 1 к 1).
- В метод /e2c/payment передается PaymentId, полученный в ответ на вызов метода /e2c/Init. Выплата продавцу производится на сумму, которая была отправлена в методе /e2c/Init
- Для повторных выплат, повторяются пп. 5-6 Повторные выплаты возможны в рамках остатка на балансе сделки. Повторные выплаты возможны только, если выполнены оба условия одновременно:
 - а. в предыдущих выплатах не был передан параметр SpFinalPayout со значением true
 - б. в предыдущих выплатах не был достигнут нулевой баланс сделки.После передачи параметра SpFinalPayout со значением true, либо достижения нулевого баланса сделка закрывается.
- Для открытия новой сделки Площадка в методе /v2/Init передает в секции DATA параметр StartSpAccumulation='1N' и не передает параметр SpAccumulationId. Одновременно может быть открыто любое количество сделок

Пример передачи параметров в секции DATA на примере запроса /v2/Init

```
{
  "TerminalKey": "1234567890",
  "Amount": "5000",
  "OrderId": "1234567891",
  "Description": "Безопасная сделка",
  "DATA":
  {
    "Phone": "+71234567890",
    "Email": "a@test.com", "StartSpAccumulation": "1N"
  }
}
```

Пример передачи параметров в секции DATA на примере запроса /e2c/Init

Боевой URL: <https://securepay.tinkoff.ru/e2c/Init>

Метод: POST

Content-Type : application/x-www-form-urlencoded

request body:

TerminalKey:1573803282696E2C

OrderId:11-1595939854

IP:192.168.40.74

CardId:70000000804

CustomerKey:testCustomerKey

Amount:4000

Currency:643

DATA:Email=autotest@test.ru|SpAccumulationId=261|SpFinalPayout=true DigestValue:qfeohMmr-sEvr4QPB8CeZETb+W6VDEGnMrf+oVjySaMU=

SignatureValue:rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMe112ERB-

SIOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g== X509SerialNumber:1234567

Наименование	Тип	Обязательность	Описание
StartSpAccumulation	Boolean	Нет	Признак управления накоплением: если передано значение true, то начинается накопление суммы для выплаты продавцу
SpAccumulationId	String	Нет	Идентификатор созданного накопителя

Пример http(s)-нотификации с параметром SpAccumulationId:

```
{
  «TerminalKey»:«1510572937960»,
  «OrderId»:«test2»,
  «Success»:true,
  «Status»:«CONFIRMED»,
  «PaymentId»:2006896,
  «ErrorCode»:«0»,
  «Amount»:102120,
  «CardId»:867911,
  «Pan»:«430000**0777»,
  «ExpDate»:«1122»,
  «SpAccumulationId»: 756439112,
  «Token»:«d0815e288f121255d5d6b77831fb486cc5e9f91914a3f58a99b6118b54676d84»
}
```

Параметр SpAccumulationId учитывается при формировании Token в нотификации.

9. Метод createSpDeal

Описание: Метод создает новую сделку

Запрос

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

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

Метод: POST

Таблица 9.1 Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком
Type	String	Да	Тип сделки (N1/1N)
Token	String	Да	Подпись запроса

Пример запроса:

```
{
  "TerminalKey": "testRegressBank",
  "Type": "N1",
  "Token": "30797e66108934dfa3d841b856fdad227c6b9c46d6a39296e02dc800d86d181e"
}
```

Таблица 9.2 Параметры ответа

Наименование	Тип	Обязательность	Описание
SpAccumulationId	String	Да	Идентификатор сделки
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «00» - если успешно

Пример ответа:

```
{
  "SpAccumulationId": "123",
  "Success": true,
  "ErrorCode": "00"
}
```