ОПИСАНИЕ ПРОТОКОЛА A2C_V2 Мультирасчеты

10.11.2025

БИЗНЕС

Версия документа: 1.12

Оглавление

История изменений

Версия	Описание	Дата
1.0	Документ создан	12.08.2024
1.1	Обновлен тип параметра PartnerId в методе Init	02.09.2024
1.2	Добавлен подраздел 3.1	24.09.2024
1.3	Добавлен метод CancelPayment	21.11.2024
1.4	Обновлена схема проведения платежа и список статусов	21.03.2025
1.5	Обновлено описание параметра LevelOfConfidence Обновлен список ошибок	19.05.2025
1.6	Добавлено описание параметра Amount в запросе метода CancelPayment Добавлено описание параметров OriginalAmount и NewAmount в ответе метода CancelPayment Обновлено описание статуса операции в методе CancelPayment В раздел 9 добавлены карты для проведения тестирования Выплат СБП	20.06.2025
1.7	Добавлен параметр ExternalRequestId в параметры запроса и ответа для метода CancelPayment	22.07.2025
1.8	Добавлен параметр Token в методах, где ранее ошибочно отсутствовал Исправлена опечатка в примерах запроса Добавлены минимальные значения для методов Init и CancelPayment	14.08.2025
1.9	Добавлен раздел Выплаты самозанятым	22.08.2025
1.10	Обновлены описание метода AddCard и список ошибок	03.09.2025
1.11	Исправления в параметрах метода AddCard	28.10.2025
1.12	Добавлены таблицы 2.4.3 и 2.4.4 с примерами заполнения	10.11.2025

Термины и сокращения

Термин	Определение
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.
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.
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 банка
Партнер	Продавец, являющийся ЮЛ.

1. Параметры выплат

Параметры выплат настраиваются отдельно на каждый терминал.

Таблица 1.1. Параметры выплат

Название параметра	Формат	Описание
TerminalKey	20 символов (чувствительно к регистру)	Уникальный символьный ключ терминала. Устанавливается банком
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://business.tbank.ru. Он совпадает с паролем от терминала интернет-эквайринга

Таблица 1.2. Параметры Success Add Card URL и Fail Add Card URL

Наименование	Описание
Success	Возможные значения: true – привязка карты завершилась успешно; false – привязка карты не завершилась
ErrorCode	Код ошибки (0 – если ошибки не было).
OrderId	Номер заказа в системе Площадки.
Message	Заголовок ошибки (заполняется только в случае ошибки).
Details	Детальное описание ошибки (заполняется только в случае ошибки).

Например:

https://securepay.tinkoff.ru/html/payForm/e2c/html/e2cSuccess.html?Success=${Success}&ErrorCode=${ErrorCode}&Message=${Message}&Details=${Details}&PaymentId=${PaymentId}&TranDate=${TranDate}&BackUrl=${BackUrl}

2. Методы приема платежей

2.1. Общая информация

Выдача средств осуществляется вызовом методов с передачей параметров методом GET или POST в зависимости от метода. Все методы и передаваемые параметры являются чувствительными к регистру. Порядок передачи параметров в запросе значения не имеет.

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

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2

Боевой URL: https://securepay.tinkoff.ru/e2c/v2

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала.

2.2. Выплаты самозанятым

Для взаимодействия с самозанятыми используется отдельная API-коллекция Самозанятые. Это единственный тип получателей, для которого требуются дополнительные настройки.

Преднастройка:

1. Выпустите токен для доступа к коллекции **Самозанятые** (можно выбрать всю коллекцию или отдельные методы) по инструкции.
2. Выпустите mTLS-сертификат по инструкции для сервиса **Моментальные выплаты** или **T-API**. Если у вас уже выпущен такой сертификат, можно использовать его.

Важно:

- отредактировать сертификат/токен не получится, для изменения выпустите новый сертификат/токен;
- токен и сертификат используются для подписи запросов только по методам из коллекции Самозанятые;
- по вопросам выпуска сертификата/токена обращайтесь в чат поддержки ЛК Бизнеса.

Порядок вызова методов и их описание:

1. **Отправить запрос на подключение самозанятого к Т-Банку** — банк отправляет запрос самозанятому в приложении «Мой налог» на привязку к Т-Банку. Если самозанятый отклонил привязку, нужно повторить вызов метода.
   Метод вызывается однократно, не требуется при каждой проверке статуса самозанятого.
   Проверку статуса самозанятого можно выполнить только после его привязки к Т Банку.
2. **Получить статус самозанятого** — возвращает информацию о статусе самозанятого, дате постановки на учет, выданных Т-Банку правах, регионе и виде деятельности.
3. **Запросить сумму доходов самозанятого за период** — метод возвращает данные из ФНС о доходе самозанятого за период.
4. **Зарегистрировать доход самозанятого** — метод отправляет данные о доходе самозанятого в ФНС и пробивает чек.
5. **Получить информацию о чеке** — метод возвращает информацию по конкретному чеку.
6. **Аннулировать доход самозанятого** — метод отменяет ранее пробитый чек.

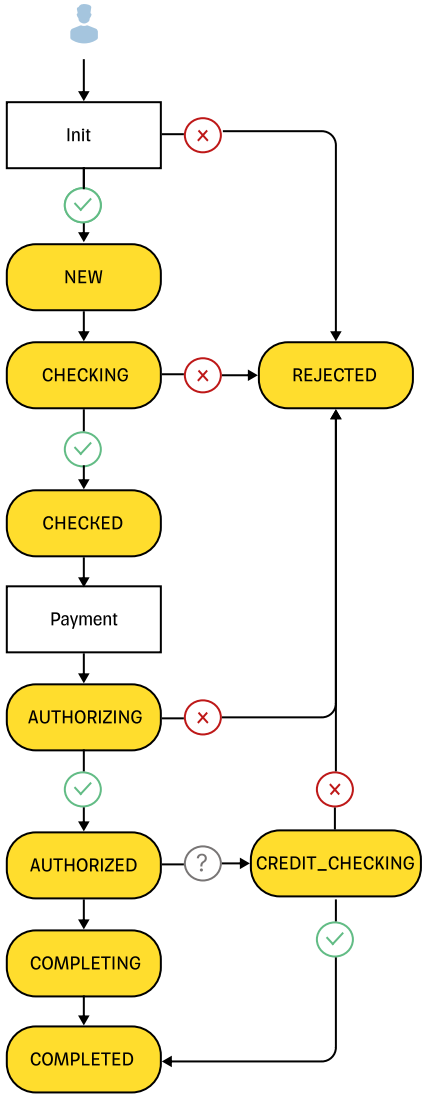
По вопросам, связанным с методами проверки самозанятого, обратитесь на почту: openapi@tbank.ru.

Методы из коллекции Самозанятые позволяют делать проверку по самозанятому (статус, лимит, пробитие/аннулирование чека), а сама выплата выполняется отдельными методами Init и Payment, и **эти процессы не связаны**. Например:

- если самозанятый не дал разрешение на привязку самозанятого к Т-Банку, выплата возможна, но она будет считаться выплатой физлицу, не самозанятому;
- если у самозанятого превышен лимит или он снялся с учета, выплату можно сделать как физлицу;
- если чек выбит, а выплата не прошла — чек можно аннулировать;
- если выплата прошла, но чек не выбит (например, сервис ФНС недоступен) — чек можно выбить позже.

2.3. Схема проведения платежа

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



Полный список возможных статусов операции:

- **NEW** — новая сессия,
- **AUTHORIZING** — авторизация,
- **CHECKING** — проверка данных,
- **CREDIT_CHECKING** — на стадии обработки,
- **COMPLETING** — операция выполняется,
- **REJECTED** — операция отклонена,
- **CHECKED** — проверка прошла успешно,
- **COMPLETED** — операция успешно выполнена.

2.4. Метод Init

Метод для схем PCI DSS \ Без PCI DSS

Описание: Инициирует платежную сессию.
Примечание: При выплате по СБП не требуется использовать метод Payment. Выплата происходит в рамках одного метода **Init**.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/Init/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/Init/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 2.4.1. Параметры запроса на выплату

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
OrderId	String	Да	Номер заказа в системе Площадки
IP	String	Нет	IP-адрес клиента
PaymentRecipientId	String	Да	Идентификатор получателя выплаты
DealId	Number	Да	Идентификатор сделки
LevelOfConfidence	String	Нет	Уровень проверки получателя выплаты. Параметр необходим только в том случае, если его передачу потребовал отдел рисков на этапе подключения. Возможные значения: "low", "moderate", "high"
FinalPayout	Boolean	Нет	Признак финальной выдачи. Если передан в значении true - сделка автоматически закроется после выплаты
DigestValue	String	Да**	Значение хеша в Base64
SignatureValue	String	Да**	Значение подписи в Base64
X509SerialNumber	String	Да**	Серийный номер сертификата

Token	String	Нет**	Подпись запроса
CardId	String	Да*	Идентификатор карты пополнения, привязанной с помощью метода AddCard.
CardData	String	Да*	Данные карты пополнения
Phone	String	Да*	Номер телефона получателя Формат: 11 цифр Пример: 70123456789
SbpMemberId	Number	Да*	Идентификатор банка-получателя в СБП Получить список идентификаторов банка можно через Метод GetSbpMembers
PartnerId	String	Да*	ID партнера магазина Мультирасчётов, которому предназначается выплата. Партнер должен быть зарегистрирован в системе Банка (подробнее в документации). Передается значение из shopCode, полученного при регистрации партнера.
Amount	Number	Да	Сумма в копейках. Минимальное значение — 100
Currency	Number	Нет	Код валюты ISO 4217 (например, 643). Если передан Currency, и он разрешен для Продавца, то транзакция будет инициирована в переданной валюте. Иначе будет использована валюта по умолчанию для данного терминала.
DATA	Object	Нет	JSON объект, содержащий дополнительные параметры в виде “ключ”:“значение”. При передаче параметра CustomerKey, переданные в Data параметры привяжутся к пользователю. Максимальная длина для каждого передаваемого параметра: – Ключ – 20 знаков; – Значение – 100 знаков. Максимальное количество пар «ключзначение» не может превышать 20.
senderAccountInfo	Object	Нет	Объект для передачи данных отправителя (необходим для выплат на иностранные карты)
recipientAccountInfo	Object	Нет	Объект для передачи данных получателя (необходим для выплат на иностранные карты)

***** условная обязательность передачи:

- **CardId** для выплаты на привязанную карту
- или **CardData** для выплаты по зашифрованным данным карты
- или набор параметров **Phone** и **SbpMemberId** для выплаты по СБП
- или **PartnerId** для выплат Партнеру

****** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Таблица 2.4.2. Параметры CardData

Наименование	Тип	Обязательность	Описание
PAN	Number	Да	Номер карты
ExpDate	Number	Нет	Месяц и год срока действия карты в формате MMYY
CardHolder	String	Нет	Имя и фамилия держателя карты (как на карте)
CVV	String	Нет	Код защиты (с обратной стороны карты)

Объект CardData собирается в виде списка «ключ=значение» (разделитель «;»), зашифровывается открытым ключом (X509 RSA 2048), получившееся бинарное значение кодируется в Base64.

Открытый ключ генерируется в Т-Бизнес. Для получения открытого ключа для шифрования CardData можете написать на acq_help@tbank.ru или обратиться к сотруднику, который вам помогал с процессом выпуска терминала.

Пример значения элемента формы CardData:

PAN=4300000000000777;ExpDate=0523;CardHolder=IVAN PETROV;CVV=111

Привязка карты, переданной в объекте CardData, должна происходить в момент вызова Init к указанному CustomerKey (если он был передан)

Таблица 2.4.3. Параметры объекта senderAccountInfo

Наименование		Тип	Обязательность	Описание
addressInfo		Object	Обязателен для нерезидентов	Блок адресных данных отправителя
	address	String	Да	Адрес отправителя
	apartment	String	Да	Номер квартиры отправителя

	building	String	Да	Номер дома отправителя
	city	String	Да	Город отправителя
	country	String	Да	Код страны по стандарту ISO
	street	String	Да	Улица отправителя
	postalCode	String	Да	Почтовый индекс отправителя
personInfo		Object	Обязателен для нерезидентов	Блок персональных данных отправителя
	firstName	String	Да	Имя
	lastName	String	Да	Фамилия
	middleName	String	Да	Первая буква отчества. При отсутствии отчества передается пробел
	dateOfBirth	String	Да	Дата рождения
	citizenship	String	Да	Гражданство
passportInfo		Object	Обязателен для нерезидентов	Блок паспортных данных
	series	String	Да	Серия паспорта
	number	String	Да	Номер паспорта
	issueDate	String	Да	Дата выдачи паспорта
accountNumber		String	Да	Номер счета

Пример заполнения объекта senderAccountInfo:


```
"senderAccountInfo": {
    "addressInfo": {
        "address": "100000 Russia Moscowovie Moscow Titushkino 1  111",
        "apartment": "111",
        "building": "1",
        "city": "Moscow",
        "country": "643",
        "street": "Titushkino",
        "postalCode": "100000"
    },
```

```
    "personInfo": {
        "firstName": "Nikolay",
        "lastName": "Petrov",
        "middleName": "A",
        "dateOfBirth": "21.12.2000",
        "citizenship": "RUS"
    },
    "passportInfo": {
        "series": "00",
        "number": "000000",
        "issueDate": "12.12.2023"
    },
    "accountNumber": "5546329130144633"
}
```


Таблица 2.4.4. Параметры объекта recipientAccountInfo

Наименование		Тип	Обязательность	Описание
personInfo		Object	Да	Блок персональных данных получателя
	firstName	String	Да	Имя
	lastName	String	Да	Фамилия
	middleName	String	Да	Первая буква отчества. При отсутствии отчества передается пробел
	dateOfBirth	String	Да	Дата рождения
	citizenship	String	Да	Гражданство

Пример заполнения объекта senderAccountInfo:


```
{
"TerminalKey": "TerminalKeyE2C",
"recipientAccountInfo": {
    "personInfo": {
        "firstName": "Darja",
        "lastName": "Shulgan",
        "middleName": "A",
        "dateOfBirth": "21.12.2000",
        "citizenship": "RUS"
    }
}
```

Пример запроса для выплат на карту:


```
{
"TerminalKey": "TerminalKeyE2C",
"OrderId": " TestOrder ",
"CardId": "11111",
"Amount": 1751,
"PaymentRecipientId": "+7 999123456”,
"DealId": "112233",
"LevelOfConfidence": "moderate”,
"FinalPayout": true,
"Token": "17b0ac0170ec685cd9e3fc57948d21aeb45778c2c7da54140e9f258cb0a68ba1",
 "senderAccountInfo": {
   "addressInfo": {
      "address": "Moscow 5 259",
      "apartment": "259",
      "building": "5",
      "city": "Moscow ",
      "country": "643",
      "street": "Moscow ",
      "postalCode": "215010"
   },
   "personInfo": {
      "firstName": " Отправитель ",
      "lastName": " Тест ",
      "middleName": " Тестович ",
      "dateOfBirth": "01.01.1990",
      "citizenship": "RUS"
   },
   "passportInfo": {
      "series": "00 11",
      "number": "123456",
      "issueDate": "01.01.2010"
   },
 },
"recipientAccountInfo": {
   "personInfo": {
      "firstName": " Получатель ",
      "lastName": " Тест",
      "middleName": " Тестович",
      "dateOfBirth": "01.01.2000",
      "citizenship": "BLR"
   }
}
```


Пример запроса для выплат СБП:


```
{
   "TerminalKey":  "TerminalKeyE2C",
```

```
  "OrderId": "testSBP_10",
  "Phone": "79998887766",
  "SbpMemberId": "100000000004",
  "FinalPayout": "true",
  "Amount": 100,
  "DealId": "9043456",
  "PaymentRecipientId": "79066589133",
  "Token": "e24fd85c4c20e85d2eab5f65e8b2066c83970b"
}
```


Пример запроса для выплат Партнеру:


```
{
  "TerminalKey": "TerminalKeyE2C",
  "OrderId": "TestOrder",
  "Amount": 1751,
  "PartnerId": "523456”,
  "DealId": "112233",
  "FinalPayout": true,
  "Token": "cb70d75fff815c433b17297593e66eb33d0dd90ec5933b24272820cc564e7dca"
}
```


Ответ

Формат ответа: **JSON**

Таблица 2.4.5 Параметры ответа при выплате на карту

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
Amount	Number	Да	Сумма в копейках
OrderId	String	Да	Номер заказа в системе Площадки
Success	bool	Да	Успешность операции
Status	String	Да	Статус транзакции
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
CardId	String	Нет	Идентификатор карты в системе Банка
MaskedFio	String	Нет	Маскированные данные ФИО получателя (для выплат по СБП)
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Примеры ответа:


```
{
  "Success": true,
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "Status": "CHECKED",
  "PaymentId": "2353039",
  "OrderId": " PaymentTestN ",
  "Amount": 100000
}
```


Пример ответа при выплате по СБП:


```
{
  "TerminalKey": "TerminalKeyE2C",
  "OrderId": "Номер заказа",
  "PaymentId": "4443333222211",
  "Amount": "10000",
  "Success": "true",
  "Status": "COMPLITING",
  "ErrorCode": "0",
  "MaskedFio": "Иван И."
}
```


Статус платежа
при выплате на карту или выплате Партнеру:

- при успешном сценарии: **CHECKED**
- при неуспешном: **REJECTED**

при выплате по СБП:

- при успешном сценарии: **COMPLETED**
- при неуспешном: **REJECTED**
- временный статус обработки: **COMPLITING**

2.5. Метод Payment

Метод для схем PCI DSS \ Без PCI DSS

Описание: Пополняет карту.
Примечание: При выплате по СБП не требуется использовать метод **Payment**. Выплата происходит в рамках одного метода Init.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/Payment/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/Payment/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 2.5.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата
Token	String	Нет*	Подпись запроса

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:


```
{
   "TerminalKey": "TerminalKeyE2C",
   "PaymentId": "700000085140",
   "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
   "SignatureValue": "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERB-
SlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
   "X509SerialNumber": "2613832945"
}
```

Ответ

Формат ответа: **JSON**

Таблица 2.5.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
OrderId	String	Да	Номер заказа в системе Площадки
Success	bool	Да	Успешность операции (true/false)
Status	String	Да	Статус транзакции
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:


```
{
  "Success": "true",
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "Status": "COMPLETED",
  "PaymentId": "10063",
  "OrderId": "21050"
}
```


Статус операции:

- при успешном сценарии и одностадийном проведении платежа: **COMPLETED**
- при неуспешном: **REJECTED**
- операция обрабатывается: **CREDIT_CHECKING***

*операция будет находиться во временном статусе CREDIT_CHECKING в течении первых 10-20 минут.

В течение этого времени можно вызвать метод GetState для уточнения конечного статуса выплаты, отличного от CREDIT_CHECKING, а именно: COMPLETED/REJECTED/CHECKED

2.6. Метод GetState

Метод для схем PCI DSS \ Без PCI DSS

Описание: Возвращает статус платежа.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/GetState/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/GetState/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 2.6.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
IP	String	Нет	IP-адрес клиента
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата
Token	String	Нет*	Подпись запроса

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:


```
{
  "TerminalKey": "TerminalKeyE2C",
  "PaymentId": "700000085101",
  "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
  "SignatureValue":
"rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
  "X509SerialNumber": "2613832945"
}
```

Ответ

Формат ответа: **JSON**

Таблица 2.6.2. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
OrderId	String	Да	Номер заказа в системе Площадки
Success	Boole	Да	Успешность операции (true/false)
Status	String	Да	Статус транзакции
Amount	Number	Нет	Сумма отмены в копейках
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:

```
{
  "Success": "true",
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "Status": "NEW",
  "PaymentId": "10063",
  "OrderId": "21057"
}
```

Возможные статусы транзакции:

Статус	Промежуточный?	Значение
NEW	Нет	Платеж зарегистрирован в шлюзе, но его обработка в процессинге не начата
CHECKING	Да	Платеж на этапе проверки данных

CHECKED	Нет	Данные проверены
COMPLETING	Да	Начало зачисления денежных средств
COMPLETED	Нет	Денежные средства зачислены на карту получателя
REJECTED	Нет	Платеж отклонен Банком
CREDIT_CHECKING*	Да	На стадии обработки. В течение 60 минут операции присвоится конечный статус согласно схеме

* операция обрабатывается и будет находиться во временном статусе CREDIT_CHECKING в течении первых 10-20 минут. В течение этого времени можно вызвать метод GetState для уточнения конечного статуса выплаты: COMPLETED/REJECTED/CHECKED.

2.7. Метод GetSbpMembers

Метод для схем PCI DSS \ Без PCI DSS

Описание: Получение списка идентификаторов банков, участвующих в СБП.

Запрос

Боевой URL: https://securepay.tinkoff.ru/a2c/sbp/GetSbpMembers

Метод: POST

Таблица 2.7.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
Token	String	Нет*	Уникальный идентификатор транзакции в системе Банка
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:

```
{
  "TerminalKey": "TerminalKeyE2C",
  "Token": "19b67a3be6500f17ccb607e5f874f614"
}
```

Ответ

Формат ответа: **JSON**

Таблица 2.7.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
Success	Boolean	Да	Успешность прохождения запроса (true/false)
ErrCode	String	Да	Код ошибки, «0» - если успешно

Message	String	Нет	Краткое описание ошибки
Members	Object	Да	

Таблица 2.7.3. Параметры объекта Members

Наименование	Тип	Обязательность	Описание
MemberId	String	Да	Код участника в СБП
MemberName	String	Нет	Наименование участника
MemberNameRus	String	Да	Наименование участника на русском

Пример ответа:


```
{
  "Success": true,
  "ErrorCode": "0",
  "Members": [
      {
          "MemberId": "100000000004",
          "MemberName": "T-Bank",
          "MemberNameRus": "Т-Банк"
      },
      {
          "MemberId": "100000000004",
          "MemberName": "T-Bank",
          "MemberNameRus": "Т-Банк"
      },
      {
          "MemberId": "100000000004",
          "MemberName": "T-Bank",
          "MemberNameRus": "Т-Банк"
      }
  ]
}
```

2.8. Метод CancelPayment

Метод для схем PCI DSS \ Без PCI DSS

Описание: Отмена выплаты Партнеру.
Примечание: Метод применяется только по успешной выплате Партнеру в рамках открытой сделки.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/CancelPayment

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/CancelPayment

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 2.8.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
PaymentId	Number	Да	Идентификатор сессии выплаты ЮЛ, по которой необходимо сделать возврат
DealId	String	Да	Идентификатор сделки
Amount	Number	Нет*	Сумма отмены в копейках Минимальное значение — 1
ExternalRequestId	String	Нет**	Идентификатор операции на стороне мерчанта
Token	String	Нет***	Уникальный идентификатор транзакции в системе Банка
DigestValue	String	Да***	Значение хеша в Base64
SignatureValue	String	Да***	Значение подписи в Base64
X509SerialNumber	String	Да***	Серийный номер сертификата

* Если параметр **Amount** не указан, производится полная отмена выплаты Партнеру. Отмена производится на сумму остатка по сессии.

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

- если такой запрос уже есть, то в ответе вернется текущее состояние платежной операции;

- если такого запроса нет, то произойдет отмена выплаты.

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

*** Передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:


```
{
 "TerminalKey": "TerminalKeyE2C",
 "PaymentId": "4713907670",
 "DealId": "112233",
 "Amount": 3000,
 "ExternalRequestId": "1234567890"
 "DigestValue": "cb70d75fff815c433b17297593e66eb33d0dd90ec5933b24272820cc564e7dca",
 "SignatureValue": "cb70d75fff815c433b17297593e66eb33d0dd90ec5933b24272820cc564e7dca",
 "X509SerialNumber": "2613832945"
}
```


Ответ

Формат ответа: **JSON**

Таблица 2.8.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
OrderId	String	Да	Номер заказа в системе Площадки
Success	Boolean	Да	Успешность прохождения запроса (true/false)
Status	String	Да	Статус транзакции
PaymentId	Number	Да	Уникальный идентификатор транзакции в системе Банка
OriginalAmount	Number	Да	Сумма в копейках до операции отмены
NewAmount	Number	Да	Сумма в копейках после операции отмены
ExternalRequestId	String	Нет	Идентификатор операции на стороне мерчанта
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:


```
{
  "Success": true,
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "Status": "PART_CREDIT_CANCELED",
  "ExternalRequestId": "1234567890",
  "PaymentId": "4713907670",
  "OriginalAmount": 10000,
  "NewAmount": 7000,
  "OrderId": "2124144154"
}
```


Статус операции:

- при обработке запроса — **CREDIT_CANCELING** или **CANCEL_CHECKING**
- при успешном выполнении сессия переходит в один из статусов: **CREDIT_CANCELED** — при отмене на полную сумму, **PART_CREDIT_CANCELED** — в случае частичной отмены
- при неуспешном выполнении сессия получает статус **COMPLETED** , если по ней не было отмен, либо **PART_CREDIT_CANCELED**, если имелись частичные отмены

3. Алгоритм формирования подписи запроса

3.1. 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 e2c/v2/Init:


```
{
 "TerminalKey": "TerminalKeyE2C",
 "Amount": "1751",
 "OrderId": "autoOrd1615285401068DELb",
 "CardId": "70000000857",
 "Token": "76956ae79e4b33e9833ab09c344c638e14d772d0e423aec4db65039622b2f1b6"
}
```


3.2. 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","TerminalKeyE2C"},{"PaymentId","20150"}]

2) Сортировать по Ключам:

[{"PaymentId","20150"},{"TerminalKey","TerminalKeyE2C"}]

3) Конкатенировать значения:

20150TerminalKeyE2C

4) Вычислить хэш-сумму по алгоритму SHA256 и получить результат в бинарном виде.

5) Закодировать получившееся в пункте 4 бинарное значение в Base64 и записать значение в DigestValue

6) Подписать получившееся в пункте 4 бинарное значение с помощью RSA ключа* алгоритмом RSA-SHA256, закодировать результат в BASE64 и записать в SignatureValue

*Инструкция по получению RSA ключа доступна по ссылке.

Ниже представлены примеры реализации работы с библиотекой:

Язык	Ссылка
Java	https://cdn.t-static.ru/static/documents/rsa-crypto-lib-java-mapi.zip

3.3. С помощью сертификата КриптоПро:

Для формирования подписи запроса необходимо:

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

[{"TerminalKey","TerminalKeyE2C"},{"PaymentId","20150"}]

2) Сортировать по Ключам:

[{"PaymentId","20150"},{"TerminalKey","TerminalKeyE2C"}]

3) Конкатенировать значения:

20150TerminalKeyE2C

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 получится значение в Base64).

*Инструкция по получению сертификата КриптоПро описана в п.7.

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

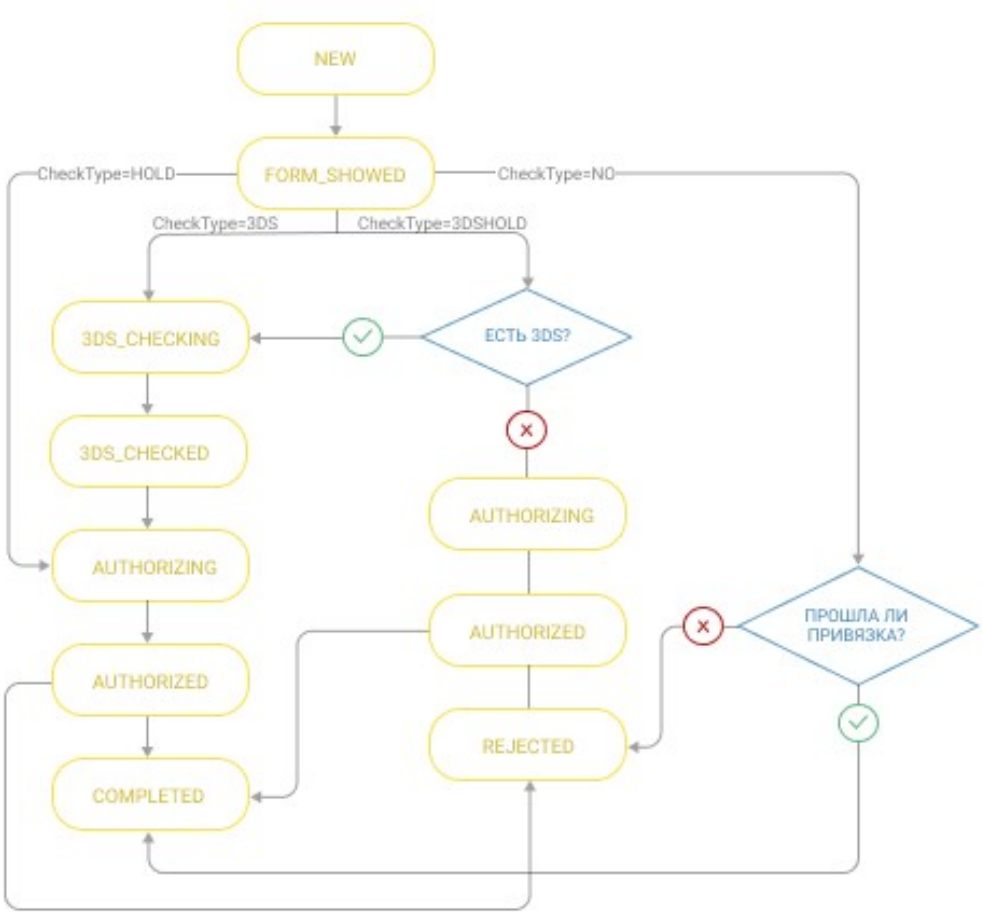
Язык	Ссылка	Версия КриптоПро
C#	Поддержка 2012 ГОСТ	КриптоПРО CSP
Java	Поддержка 2012 ГОСТ	КриптоПРО JCP
PHP	Поддержка 2012 ГОСТ	КриптоПРО CSP

4. Методы работы с привязанными картами и клиентами

4.1. Статусная схема привязки карт

Необходимо обратить внимание, что для корректной работы методов банком должна быть разрешена привязка карт и клиентов к терминалу Продавца.

В результате привязки карты к параметру CustomerKey будет привязана CardId.



Описание статусов:

- NEW — новая сессия;
- FORM_SHOWED — показ формы привязки карты;
- 3DS_CHECKING — отправка пользователя на проверку 3DS
- 3DS_CHECKED — пользователь успешно прошел проверку 3DS;
- AUTHORIZING — отправка платежа на 0 руб;
- AUTHORIZED — платеж на 0 руб прошел успешно;
- COMPLETED — привязка успешно завершена;
- REJECTED — привязка отклонена.

4.2. Метод AddCustomer

Метод для схем PCI DSS \ Без PCI DSS

Описание: Регистрирует покупателя в терминале Продавца.

Возможна автоматическая привязка покупателя и карты, по которой был совершен платеж при передаче параметра CustomerKey в методе Init. Это можно использовать для сохранения и последующего отображения Покупателю замаскированного номера карты, по которой будет совершен рекуррентный платеж.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/AddCustomer/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/AddCustomer/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.2.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
IP	String	Нет	IP-адрес запроса
Email	String	Нет	Email клиента
Phone	String	Нет	Телефон клиента (+71234567890)
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата
Token	String	Нет*	Подпись запроса

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:


```
{
  "TerminalKey": "TerminalKeyE2C",
  "CustomerKey": "TestCustomer20",
  "IP": "192.168.40.74",
  "Email": "autotest@test.ru",
  "Phone": "+71234567890",
  "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
  "SignatureValue":
"rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
  "X509SerialNumber": "2613832945"
}
```


Формат ответа: **JSON**

Ответ

Таблица 4.2.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Площадке при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» в случае успеха
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:


```
{
  "Success": "true",
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "CustomerKey": "Customer1"
}
```

4.3. Метод GetCustomer

Метод для схем PCI DSS \ Без PCI DSS

Описание: Возвращает данные покупателя, сохраненные для терминала Продавца.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/GetCustomer/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/GetCustomer/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.3.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
IP	String	Нет	IP-адрес запроса
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата
Token	String	Нет*	Подпись запроса

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:

```
{
   "TerminalKey": "TerminalKeyE2C",
   "CustomerKey": "TestCustomer1",
   "IP": "192.168.40.74",
   "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
   "SignatureValue":
"rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
   "X509SerialNumber": "2613832945"
}
```

Формат ответа: **JSON**

Ответ

Таблица 4.3.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Площадке при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» в случае успеха
Email	String	Нет	Email клиента
Phone	String	Нет	Телефон клиента (+71234567890)
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:

```
{
  "Success": true,
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "CustomerKey": "Customer1"
}
```

4.4. Метод RemoveCustomer

Метод для схем PCI DSS \ Без PCI DSS

Описание: Удаляет сохраненные данные покупателя.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/RemoveCustomer/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/RemoveCustomer/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.4.1. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
IP	String	Нет	IP-адрес запроса
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата
Token	String	Нет*	Подпись запроса

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:


```
{
   "TerminalKey": "TerminalKeyE2C",
   "CustomerKey": "TestCustomer11",
   "IP": "192.168.40.74",
   "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
   "SignatureValue":
   "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
   "X509SerialNumber": "2613832945"
}
```

Формат ответа: **JSON**

Ответ

Таблица 4.4.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Площадке при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» в случае успеха
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:

```
{
    "Success": true,
    "ErrorCode": "0",
    "TerminalKey": "TerminalKeyE2C",
    "CustomerKey": "Customer1"
}
```

4.5. Метод GetCardList

Метод для схем PCI DSS \ Без PCI DSS

Описание: Возвращает список привязанных карт у покупателя. В том числе показывает удаленные карты. Не возвращает список счетов, привязанных по номеру телефона через СБП

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/GetCardList/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/GetCardList/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.5.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
IP	String	Нет	IP-адрес запроса
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата
Token	String	Нет*	Подпись запроса

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:


```
{
   "TerminalKey": "TerminalKeyE2C",
   "CustomerKey": "TestCustomer1",
   "IP": "192.168.40.74",
   "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
   "SignatureValue":
"rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
   "X509SerialNumber": "2613832945"
}
```

Формат ответа: **Массив JSON**

Ответ

Таблица 4.5.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
Pan	String	Да	Номер карты 411111******1111
CardId	String	Да	Идентификатор карты в системе Банка
Status	String	Да	Статус карты: A – активная, D – не активная, E – срок действия карты истек
RebillId	Number	Да	Идентификатор рекуррентного платежа
ExpDate	String	Нет	Срок действия карты

Пример ответа:


```
[
    {
        "CardId": "894952",
        "Pan": "532130******5598",
        "Status": "A",
        "RebillId": "130802844",
        "CardType": 0,
        "ExpDate": "0423"
    },
    {
        "CardId": "894955",
        "Pan": "518223******0036",
        "Status": "A",
        "RebillId": "13816414",
        "CardType": 0,
        "ExpDate": "1122"
    }
]
```

4.6. Метод AddCard

Метод для схем PCI DSS \ Без PCI DSS

Описание: Добавляет привязанную карту к покупателю. В случае успешной привязки переадресует клиента на Success Add Card URL в противном случае на Fail Add Card URL.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/AddCard/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/AddCard/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.6.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
CheckType	String	Нет	Если CheckType не передается, автоматически проставляется значение NO. Возможные значения: 1. NO – сохранить карту без проверок. RebillID для рекуррентных платежей не возвращается; 2. HOLD – при сохранении сделать списание на 0 руб.* RebillID возвращается для терминалов без поддержки 3DS. 3. 3DS – при сохранении карты выполнить проверку 3DS и выполнить списание на 0 р. В этом случае RebillID будет только для 3DS карт. Карты, не поддерживающие 3DS, привязаны не будут 4. 3DSHOLD – при привязке карты выполняем проверку, поддерживает карта 3DS или нет. Выполняется списание на 0р.
PayForm	String	Нет	Название шаблона формы привязки

ResidentState	Boolean	Нет	Признак резидентности добавляемой карты: Возможные значения: • true - Карта РФ, • false - Карта не РФ, • null - Не специфицируется (универсальная карта)
DigestValue	String	Да**	Значение хеша в Base64
SignatureValue	String	Да**	Значение подписи в Base64
X509SerialNumber	String	Да**	Серийный номер сертификата
Token	String	Нет**	Подпись запроса

***** при CheckType = 3DS, для успешной работы метода AttachCard необходимо передать срок действия карты в CardData (параметр ExpDate).

****** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:


```
{
  "TerminalKey": "TerminalKeyE2C",
  "CustomerKey": "TestCustomer10",
  "IP": "192.168.40.74",
  "CheckType": "NO",
  "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
  "SignatureValue": "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERB-
SlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
  "X509SerialNumber": "2613832945"
}
```


Формат ответа: **JSON**

Ответ

Таблица 4.6.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
PaymentId	Number	Да	Уникальный идентификатор транзакции в системе Банка
PaymentURL	String	Да	Идентификатор страницы выплаты.
TerminalKey	String	Да	Платежный ключ, выдается Площадке при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки

RequestKey	String	Да	Идентификатор запроса на привязку карты
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» в случае успеха
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:


```
{
   "Success": "true",
   "ErrorCode": "0",
   "TerminalKey": "TerminalKeyE2C",
   "CustomerKey": "Customer1",
   "PaymentURL": "https://securepayments.tinkoff.ru/addcard/02a94edd-046c-441b-90b6-cda01f337401",
   "RequestKey": "8de92934-26c9-474c-a4ce424f2021d24d"
}
```

4.7. Метод AttachCard

Метод для схемы PCI DSS

Описание: Добавляет привязанную карту к покупателю. Метод необходимо вызывать после AddCard. В случае успешной привязки переадресует клиента на Success Add Card URL в противном случае на Fail Add Card URL.

Для прохождения 3DS второй версии перед вызовом метода должен быть вызван /v2/check3dsVersion и выполнен 3DS Method, который является обязательным при прохождении 3DS по протоколу версии 2.0.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/AttachCard/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/AttachCard/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.7.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Площадке при заведении терминала
RequestKey	String	Да	Идентификатор запроса на привязку карты
CardData	String	Да	Данные карты привязки*
DATA	Object	Нет***	Ключ = значение дополнительных параметров через “|”, например, Email = a@test.ru|Phone = +71234567890. Если ключи или значения содержат в себе спец символы, то получившееся значение должно быть закодировано функцией urlencode. Максимальная длина для каждого передаваемого параметра: Ключ – 20 знаков, Значение – 100 знаков. Максимальное количество пар «ключ-значение» не может превышать 20
DigestValue	String	Да**	Значение хеша в Base64
SignatureValue	String	Да**	Значение подписи в Base64
X509SerialNumber	String	Да**	Серийный номер сертификата

Token	String	Нет**	Подпись запроса

* алгоритм шифрования описан на стр. 10 документации.
** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

*** для 3DS второй версии в параметрах DATA необходимо передавать следующие параметры

Таблица 4.7.2. Параметры объекта DATA для 3DS v2

Наименование	Тип	Обязательность	Описание
threeDSCompInd	String	Да	Идентификатор выполнения 3DS Method 'Y' - метод выполнялся 'N' - метод не выполнялся
javaEnabled	String	Нет	Поддерживает ли браузер пользователя Java: true/false. По умолчанию значение "false"
language	String	Да	Язык браузера по формату IETF BCP47 Рекомендация по получению значения в браузере (из глобального объекта navigator): navigator.language
colorDepth	String	Нет	Глубина цвета в битах Допустимые значения: 1/4/8/15/16/24/32/48 Рекомендация по получению значения в браузере (из глобального объекта screen): screen.colorDepth По умолчанию значение будет 48
timezone	String	Да	Time-zone пользователя в минутах. Пример для UTC +5 hours: "timezone": "-300" Рекомендация по получению значения в браузере: вызов метода getTimezoneOffset()
screen_height	String	Да	Высота экрана в пикселях. Рекомендация по получению значения в браузере (из глобального объекта screen): screen.height
screen_width	String	Да	Ширина экрана в пикселях Рекомендация по получению значения в браузере (из глобального объекта screen): screen.width

cresCallbackUrl	String	Да	URL который будет использоваться для получения результата (CRES) после завершения Challenge Flow (аутентификаци с дополнительным переходом на страницу ACS)

Для 3DS Version 2 в HttpHeaders запроса обязательно должны присутствовать заголовки: “User-Agent” и “Accept”.

Пример запроса для 3DS v1:


```
{
"TerminalKey": "TerminalKeyE2C",
"CardData": "b3tSlUYwsf3Erdv5ReB7WpWK3/NBWLIwDiSLjQGOcBxAOMgs7ALd7ediORbVlORsyGZEUJSl-
RynQ9zLMyHYzWP3z2sQYGAvzOqufoVPe2AozhW3pZV+dN5s7oG-
cpXd39NDCOMa/Zw6oa3dJROZh8QYjv/sGOzUllMjXI5aHgTpxk37q6OxUakxuG7euhvSN71JqxHsNEu-
oJELAqLq7U+3tuh9AjTuiBpmEH99maK9e7gnVXgZd1Nk8vachs97xj9cL/O23qYMk7CMjldBfG4VOsYVq cH-
sKfbbJJ8CZXIJgmXhCYns1hmRD/kf3OhEZrO38LghC7IioOyxHYMhZyJoQ==",
"RequestKey": "3206b55f-83a2-486f-9da0-693dfd2af9b3",
"DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
"SignatureValue": "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERB-
SlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
"X509SerialNumber": "2613832945"
}
```


Пример запроса для 3DS v2:


```
{
    "RequestKey": "416a5d8e-54da-49bc-b5ae-22e4755c0609",
    "TerminalKey": "TerminalKeyE2C",
    "CardData":
"ku8wgdsKR+34u9dlW7JMdES17zL+U4EzBO5XW/ar8ZJhtfxlj3tbzYBFXlT77OojF93Ye7vYBOalUkBTAGiPQcpyQWQ5w
p4l3vdcCMXobBbO4Vprb8TpvPXG3BTeBv/CoxZxMh5JO8oBXGxeGySS4ARc4QDbEJxX5QzdybxyaZ3lxLFuz/jEqpQ24Rx
uFsOiwcQYP7TfwoBFnHGj8DP/+hWyLdPVPLyUrHvPUz+ot1QrcEFvnu/xG/l/M8rrn/Rqcrr7dVI2GPASlSzn+FJootn/25Lkr
JF8guAM33nZe8SIsapZKi31rA8KdykmF2KSON1zt4VqWkqdizXh1FpyOw==",
    "Token": "{{token}}",
    "DATA": {
        "language": "ru-RU",
        "screen_height": "1024",
        "screen_width": "1024",
        "timezone": "180",
        "cresCallbackUrl": "http://test.url"
    }
}
```

Формат ответа: **JSON**

Ответ

Таблица 4.7.3. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Площадке при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
RequestKey	String	Да	Идентификатор запроса на привязку карты
RebillId	String	Нет	Идентификатор рекуррентного платежа
CardId	String	Да	Идентификатор карты в системе Банка
Status	String	Да	Статус привязки карты
Success	Bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:


```
{
  "Success": true,
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "Status": "3DS_CHECKING",
  "CustomerKey": "testRegress5",
  "RequestKey": "8de92934-26c9-474c-a4ce-424f2021d24d",
  "CardId": "5555"
}
```


Для 3DS v1

! Если в ответе метода AttachCard возвращается статус 3DS_CHECKING, мерчанту необходимо сформировать запрос на URL ACS банка, выпустившего карту (в ответе параметр ACSUrl). URL: ACSUrl (возвращается в ответе метода AttachCard)

Метод: POST

Формат запроса: x-www-form-urlencoded

Таблица 4.7.4 Параметры запроса на ACSUrl

Наименование	Тип	Обязательность	Описание
MD	String	Да	Уникальный идентификатор транзакции в системе Банка (возвращается в ответе на AttachCard)
PaReq	String	Да	Результат аутентификации 3-D Secure (возвращается в ответе на AttachCard)
TermUrl	String	Да	Адрес перенаправления после аутентификации 3-D Secure (URL обработчик на стороне мерчанта, принимающий результаты прохождения 3-D Secure)

Пример запроса на ACS:

```
<body onload="document.form.submit()" >
<form name="form" action="{ACSUrl}" method="post" >
<input type="hidden" name="TermUrl" value="{TermUrl}" >
<input type="hidden" name="MD" value="{MD}" >
<input type="hidden" name="PaReq" value="{PaReq}" >
</form>
</body>
```

Формат ответа: JSON

Таблица 4.7.5 Параметры ответа на запрос на ACSUrl

Наименование	Тип	Обязательность	Описание
MD	String	Да	Уникальный идентификатор транзакции в системе Банка (возвращается в ответе на AttachCard)
PaRes	String	Да	Шифрованная строка, содержащая результаты 3D Secure аутентификации (возвращается в ответе от ACS)
FallbackOnTdsV1	Boolean	Нет	В случае невозможности прохождения аутентификации по 3DS v2, делается принудительный Fallback на 3DS v1 и данный атрибут выставляется в true, в противном случае не передается в ответе

Пример ответа для статуса 3DS_CHECKING (3DS v2):

```
{
  "Success": true,
  "ErrorCode": "0",
  "TerminalKey": "TerminalKeyE2C",
  "RequestKey":          "57d7cce7-0d55-4081-9c51-68654f273284",
"Status": "3DS_CHECKING",
  "ACSUrl": "https://acs.vendorcert.mirconnect.ru/mdpayacs/creq",
  "TdsServerTransId": "d7171a06-7159-4bdd-891a-a560fe9938d2",
  "AcsTransId": "e176d5d3-2f19-40f5-8234-46d3464e0b08"
}
```

Для 3DS v2

! Если в ответе метода AttachCard возвращается статус 3DS_CHECKING, мерчанту необходимо сформировать запрос на URL ACS банка, выпустившего карту (в ответе параметр ACSUrl). URL: ACSUrl (возвращается в ответе метода AttachCard)

Метод: POST

Формат запроса: x-www-form-urlencoded

Таблица 4.7.6. Параметры запроса creq

Наименование	Тип	Обязательность	Описание
creq	String	Да	JSON с параметрами threeDSServerTransID, acsTransID, challengeWindowSize, messageType, messageVersion закодированный в формат base-64

JSON creq

Наименование	Тип	Обязательность	Описание
threeDSServerTransID	String	Да	Идентификатор транзакции из ответа метода AttachCard
acsTransID	String	Да	Идентификатор транзакции, присвоенный ACS, полученный в ответе на AttachCard
challengeWindowSize	String	Да	Размер экрана, на котором открыта страница ACS. Допустимые значения: = 250 x 400 = 390 x 400 = 500 x 600 = 600 x 400 = Full screen
messageType	String	Да	Передается фиксированное значение «CReq»
messageVersion	String	Да	Версия 3DS, полученная из ответа метода Check3dsVersion

Пример запроса на ACS:

```
<body onload="document.form.submit()" >
<form name="form" action="{ACSUrl}" method="post" >
<input type="hidden" name="creq" value="
ewogICJhY3NUcmFucOlElJogImUxNzZkNWQzLTJmMTktNDBmNSO4MjMOLTQ2ZDMONjRlMGIwOCIsCiAgImNoYWxsZW5n
ZVdpbmR
vd1NpemUiOiAiMDMiLAogICJtZXNzYWdlVHlwZSI6ICJDUmVxIiwKICAibWVzc2FnZVZlcnNpb24iOiAiMi4xLjAiLAogICJOa
H JlZURTU 2VydmVyVHJhbnNJRCI6ICJkNzE3MWEwNiO3MTU5LTRiZGQtODkxYS1hNTYwZmU5OTM4ZDIiCnOK
" >
</form>
</body>
```

Creq, декодированный из base64:

```
{
   "acsTransID": "e176d5d3-2f19-40f5-8234-46d3464e0b08",
   "challengeWindowSize": "03",
   "messageType": "CReq",
   "messageVersion": "2.1.0",
```

"threeDSServerTransID": "d7171a06-7159-4bdd-891a-a560fe9938d2"
}

Формат ответа: Cres, полученный по NotificationUrl из запроса AttachCard

Таблица 4.7.7. Параметры ответа cres

Наименование	Тип	Обязательность	Описание
cres	String	Да	JSON с параметрами threeDSServerTransID, acsTransID, messageType, messageVersion, transStatus закодированный в формат base-64

JSON cres

Наименование	Тип	Обязательность	Описание
threeDSServerTransID	String	Да	Идентификатор транзакции из ответа метода Check3dsVersion
acsTransID	String	Да	Идентификатор транзакции, присвоенный ACS
messageType	String	Да	Передается фиксированное значение «CRes»
messageVersion	String	Да	Версия 3DS
transStatus	String	Да	Результат выполнения Challenge flow, возможны 2 значения: ‘Y’/’N’. ‘Y’ – аутентификация выполнена успешна ‘N’ – аутентификация не пройдена, пользователь отказался или ввел неверные данные

При успешном результате прохождения 3-D Secure подтверждается инициированный платеж с помощью методов Submit3DSAuthorization или Submit3DSAuthorizationV2, в зависимости от версии 3DS.

4.8. Метод Check3dsVersion

Метод для схемы PCI DSS

Описание: Проверяет поддерживаемую версию 3DS протокола по карточным данным из входящих параметров. Данный метод должен вызываться для платежей с Route=ACQ, но не для ApplePay.

При определении второй версии, возможно в ответе получение данных для прохождения дополнительного метода “3DS Method”, который позволяет эмитенту собрать данные браузера пользователя – это может быть полезно при принятии решения в пользу Frictionless Flow (аутентификация клиента без редиректа на страницу ACS).

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/v2/Check3dsVersion/
Боевой URL: https://securepay.tinkoff.ru/v2/Check3dsVersion/
*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.8.1 Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
CardData	String	Да	Данные карты*
Token	String	Да	Подпись запроса

* Объект CardData описан в разделе 2.4 Метод Init

Пример запроса:

{
"TerminalKey": "TerminalKeyE2C",
"PaymentId": "777",
"CardData":
"b3tSlUYwsf3Erdv5ReB7WpWK3/NBWLIwDiSLjQGOcBxAOMgs7ALd7ediORbVlORsyGZEUJSlRynQ9zLMyHYzWP3z2sQYG
AvzOqufoVPe2AozhW3pZV+dN5s7oGcpXd39NDCOMa/Zw6oa3dJROZh8QYjv/sGOzUllMjXl5aHgTpxk37q6OxUakxuG7euh
vSN71JqxHsNEuoJELAqLq7U+3tuh9AjTuiBpmEH99maK9e7gnVXgZd1Nk8vachs97xj9cL/O23qYMk7CMjldBfG4VOsYVq
cHsKfbbJJ8CZXIJgmXhCY ns1hmRD/kf3OhEZrO38LghC7IioOyxHYMhZyJoQ==",
"Token": "daab60d01863965284f1db558ff37534715afd0f1e726ca9611b1f90720ad03b",
}

Ответ

Таблица 4.8.2 Параметры ответа

Наименование	Тип	Обязательность	Описание
Version	String	Да	Версия протокола 3DS. Пример: “1.0.0” – первая версия “2.1.0” – вторая версия
TdsServerTransID	String	Нет	Уникальный идентификатор транзакции, генерируемый 3DS-Server, обязательный параметр для 3DS второй версии. Пример: 17d3791b-5cfa-4318-bc233d949e8c4b7e
ThreeDSMethodURL	String	Нет	Дополнительный параметр для 3DS второй версии, который позволяет пройти этап по сбору данных браузера ACS-ом
PaymentSystem	String	Да	Платежная система карты
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:


```
{
   "Success": true,
   "ErrorCode": "0",
   "Message": "OK",
   "Version": "2.1.0",
   "TdsServerTransID": "17d3791b-5cfa-4318-bc23-3d949e8c4b7e",
   "ThreeDSMethodURL": “https://acs.vendorcert.mirconnect.ru/ds/6300”
    "PaymentSystem": "mir"
}
```

Процесс прохождения этапа “3DS Method”:

Если в ответе метода был получен параметр ThreeDSMethodURL, то необходимо отправить запрос на стороне браузера по полученному ThreeDSMethodURL. для сбора информации ACS-ом о девайсе пользователя. Отправка запроса 3DS Method в браузере должна происходить в скрытом frame.

Тайм-аут ожидания ответа выполнения запроса 3DS Method перед выполнением запроса FinishAuthorize должен быть 10 секунд.

Формат запроса: x-www-form-urlencoded

Таблица 4.8.3 Описание параметров запроса 3DS Method

Название параметра	Тип	Обязательность	Описание
*threeDSMethodData	String	Y	JSON с параметрами threeDSMethodNotificationURL, threeDSServerTransID, закодированный в формат base-64

* Запрещено использовать padding

threeDSMethodNotificationURL, threeDSServerTransID

Название параметра	Тип	Обязательность	Описание
threeDSMethodNotificationURL	String	Y	Обратный адрес, на который будет отправлен запрос после прохождения threeDSMethod
threeDSServerTransID	String	Y	Идентификатор транзакции из ответа метода

Пример запроса на 3DS Method Url:

```
<body onload="document.form.submit()">
<form name="form" action="{ThreeDSMethodURL}" method="post" >
  <input type="hidden" name=" threeDSMethodData "
value="eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjU2ZTcxMmE1LTE5MGEtNDU4OC05MWJjLWUwODYyNmU3N2M0NCIs
InRocmVlRFNNZXRob2R0b3RpZmljYXRpb25VUkwiOiJodHRwczovL3Jlc3QtYXBpLXRlc3QudGlua29mZi5ydS92Mi9Db21
wbGV0ZTNEU01ldGhvZHYyInO">
</form>
</body>
```

Пример декодированного значения threeDSMethodData

```
{
"threeDSServerTransID": "56e712a5-190a-4588-91bc-e08626e77c44",
"threeDSMethodNotificationURL":https://rest-api-test.tinkoff.ru/v2/Complete3DSMethodv2 }
```

4.9. Метод Submit3DSAuthorization

Метод для схем PCI DSS

Описание: Осуществляет проверку результатов прохождения 3-D Secure и при успешном результате прохождения 3-D Secure подтверждает инициированный платеж.

При использовании одностадийной оплаты осуществляет списание денежных средств с карты покупателя. При двухстадийной оплате осуществляет блокировку указанной суммы на карте покупателя.

Запрос

Метод: POST

Тестовый URL*: https://rest-api-test.tinkoff.ru/v2/Submit3DSAuthorization

Боевой URL: https://securepay.tinkoff.ru/v2/Submit3DSAuthorization

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Формат запроса: x-www-form-urlencoded

После получения на TermUrl мерчанта ответа ACS с результатами прохождения 3-D Secure необходимо сформировать запрос к методу Submit3DSAuthorization со следующими параметрами:

Таблица 4.10.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
MD	String	Да	Уникальный идентификатор транзакции в системе Банка (возвращается в ответе от ACS)
PaRes	String	Да	Шифрованная строка, содержащая результаты 3-D Secure аутентификации (возвращается в ответе от ACS)
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
Token	String	Да	Подпись запроса
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком

Пример запроса:

```
<body onload="document.form.submit()" >
<form name="form" action="https://rest-api-test.tinkoff.ru/v2/Submit3DSAuthorization" method="post" > <input type="hidden" name="MD" value="2561504" >
 <input type="hidden" name="PaRes"
value="eJxVUttygjAU/BWG1w4mXKXOMY5WOrVTrOOtl7cAqeJI1AAO+vVNFKrlaffkZM9mD9Crsq12ZCJPd7yrmy2sa4z
HuyTlq66+mD8bvt4jMF8LxoYzFpeCEQhZntMVO9JE3vC8Hx9j27A8LzEcN7aNCPu24VIrihKXetiPdAKT/pQdCNSDiJzTsgA
1VCqKeE15QYDGh8FoTBy73fZtQDWFjInRkFi4+Uz82JbH1zJwmjEyHcwAXRDEu5IX4kQ8R/YOBEqxJeui2HcQOlGesKolS
kCqCuhmYFIqlEuVKk3IDL8uPwI3jDaBGZ4XeLxZVeFw5I7nX11AqgMSWjDpzPSxb/ma6XRct4Pl4y51oJkar5zLx1wx7NWI/
t3BfQFkxkKuoHHfMGDVfseZugLoDwO6+X16UfHFhUyk/32OMH3vZ5+nYBu/2d4xcMTDsnO4j19VqJcmpZjKYKT3q6QigJ
QMqveF6lVL9O8X+AWMIbbt" >
  <input type="hidden" name="PaymentId " value="10063" >
```

```
 <input type="hidden" name="TerminalKey" value="TerminalKeyE2C" >
 <input type="hidden" name="Token"
value="871199b37f207f0c4f721a37cdcc71dfcea880b4a4b85e3cf852c5dc1e99a8d6" > </form>
</body>
```

Ответ

Формат ответа: JSON

Таблица 4.10.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
OrderId	String	Да	Номер заказа в системе Площадки
Success	bool	Да	Успешность операции (true/false)
Status	String	Да	Статус транзакции
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Статус платежа:

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

Пример ответа:

```
{
    "Success": true,
    "ErrorCode": "0",
    "TerminalKey": "TerminalKeyE2C",
    "Status": "CONFIRMED",
    "PaymentId": "10063",
    "OrderId": "21050",
    "Amount": 100000
}
```

4.10. Метод Submit3DSAuthorizationV2

Метод для схем PCI DSS

Описание: Осуществляет проверку результатов прохождения 3-D Secure v2 и при успешном результате прохождения 3-D Secure v2 подтверждает инициированный платеж.

При использовании одностадийной оплаты осуществляет списание денежных средств с карты покупателя. При двухстадийной оплате осуществляет блокировку указанной суммы на карте покупателя.

Запрос

Метод: POST

Тестовый URL*: https://rest-api-test.tinkoff.ru/v2/Submit3DSAuthorizationV2

Боевой URL: https://securepay.tinkoff.ru/v2/Submit3DSAuthorizationV2

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Формат запроса: x-www-form-urlencoded

После получения на NotificationUrl мерчанта ответа ACS(Cres) с результатами прохождения 3-D Secure v2 необходимо сформировать запрос к методу Submit3DSAuthorizationV2 со следующими параметрами:

Таблица 4.11.1. Параметры запроса

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

Ответ

Формат ответа: JSON

Таблица 4.11.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
OrderId	String	Да	Номер заказа в системе Площадки
Success	bool	Да	Успешность операции (true/false)
Status	String	Да	Статус транзакции
PaymentId	String	Да	Уникальный идентификатор транзакции в системе Банка

ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Статус платежа:

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

Пример ответа:

{

"Success": true,

"ErrorCode": "0",

"TerminalKey": "TerminalKeyE2C",

"Status": "CONFIRMED",

"PaymentId":”10063”,

"OrderId": "21050",

"Amount":100000

}

4.11. Метод RemoveCard

Метод для схем PCI DSS \ Без PCI DSS

Описание: Удаляет привязанную карту у покупателя.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/RemoveCard/

Боевой URL: https://securepay.tinkoff.ru/e2c/v2/RemoveCard/

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.12.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
CardId	Number	Да	Идентификатор карты в системе Банка
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
IP	String	Нет	IP-адрес запроса
DigestValue	String	Да*	Значение хеша в Base64
SignatureValue	String	Да*	Значение подписи в Base64
X509SerialNumber	String	Да*	Серийный номер сертификата
Token	String	Нет*	Подпись запроса

***** передается или **Token** (если отключена валидация подписи), или набор параметров **DigestValue**, **SignatureValue**, **X509SerialNumber** (при включенной валидации подписи)

Пример запроса:

```
{
  "TerminalKey": "TerminalKeyE2C",
  "CustomerKey": "TestCustomer10",
  "CardId": 1234,
  "IP": "192.168.40.74",
  "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
```

"SignatureValue":
"rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
"X509SerialNumber": "2613832945"
}

Ответ

Формат ответа: **JSON**

Таблица 4.12.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Площадке при заведении терминала
CardId	Number	Да	Идентификатор карты в системе Банка
CustomerKey	String	Да	Идентификатор покупателя в системе Площадки
Status	String	Да	Статус карты: D – удалена.
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» в случае успеха
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:

{
"CardId": "4750",
"Status": "D",
"Success": true,
"ErrorCode": "0",
"TerminalKey": "TerminalKeyE2C",
"CustomerKey": "TestCustomer10"
}

5. Нотификации о привязке карты

Нотификации о привязке – это уведомления магазину о статусе выполнения метода привязки карты AddCard.

Описание: Т-Банк может уведомлять магазин об успешных/ошибочных привязках карты. Для этого необходимо написать своему менеджеру, либо на eacq_accounts@tbank.ru с указанием E2C- терминала и URL, на который будут отправляться POST-запросы со статусами привязки.

После успешного выполнения метода AddCard на адрес Notification URL высылается уведомление POST-запросом с информацией о привязке карты. При использовании формы привязки карты на стороне банка при обращении к методу AttachCard 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 AddCard URL или Fail AddCard URL в зависимости от результата привязки карты.

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 дамп.

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

Актуальный список внешних сетей*, используемых Т-Банком, 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 5.1 Параметры нотификации

Наименование	Тип	Описание
TerminalKey	String	Идентификатор терминала, выдается Площадке Банком
CustomerKey	String	Идентификатор покупателя в системе Площадки
RequestKey	String	Идентификатор запроса на привязку карты
Success	bool	Успешность запроса
Status	String	Статус привязки
PaymentId	String	Уникальный идентификатор транзакции в системе Банка

ErrorCode	String	Код ошибки, «0» - если успешно
CardId	Number	Идентификатор привязанной карты
Pan	String	Маскированный номер карты
ExpDate	String	Срок действия карты
NotificationType	String	Тип нотификации, всегда константа «LINKCARD»
RebillId	String	Идентификатор рекуррентного платежа
Token	String	Подпись запроса. Формируется по такому же принципу, как и в случае запросов в банк

Таблица 5.2 Статусы привязок, по которым приходят http(s)-нотификации

Status	Описание
COMPLETED	Карта успешно привязана
REJECTED	Привязка карты неуспешна

Пример http(s)-нотификации:

{"TerminalKey": "TerminalKeyE2C",
"CustomerKey": "5b718a19-2abe-1147-a7d9-b43b198ceee3",
"RequestKey": "acdad9d1-1847-4bdc-b743-db86d75253f8",
"Success": true,
"Status": "COMPLETED",
"PaymentId": "700000198023",
"ErrorCode": "0",
"CardId": 70000000707,
"Pan": "532130******1359",
"ExpDate": "1122",
"NotificationType": "LINKCARD",
"RebillId": "700000004090",
"Token": "f2fdd7fec8225872590e1558b7ea258c75df8f300d808006c41ab540dd7514d9"
}

Ответ на HTTP(s)-нотификацию:

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

PHP. Пример ответа на http(s)-нотификацию

```
<?php echo «OK»;?>
```

Java. Пример ответа на http(s)-нотификацию

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

Если ответ «OK» 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.

6. Коды ошибок, передаваемые на FailURL

Таблица 6.1. Ошибки при выплате на карту

CODE	MESSAGE	DETAILS (опционально)	Методы выплат, в которых возможно получение ошибки
0	None		Init; Payment
1	Параметры не сопоставлены		Init; Payment
2	Отсутствуют обязательные параметры		Init; Payment
3	Внутренняя ошибка системы интернет эквайринга		Init; Payment
4	Не получится изменить статус платежа		Init; Payment
5	Обратитесь в поддержку, чтобы уточнить детали		Init; Payment
9	Переадресовываемый url пуст		Init; Payment
11	Невозможно выполнить платеж		Init; Payment
17	Неверные введенные данные		Init; Payment
21	Внутренняя ошибка вызова сервиса ACQAPI		Init; Payment
53	Обратитесь к продавцу		Init; Payment
76	Операция по иностранной карте недоступна	Операция по иностранной карте недоступна. Воспользуйтесь картой российского банка	Init
78	Выплата на иностранную карту недоступна	Выплата на иностранную карту недоступна. Воспользуйтесь картой российского банка	Init
100	Попробуйте еще раз. Если ошибка повторится — обратитесь в поддержку;		Init
102	Операция отклонена, пожалуйста обратитесь в интернет-магазин или воспользуйтесь другой картой.	Заказ не может быть оплачен, обратитесь службу поддержки	Init; Payment
102	Превышен лимит на сумму выплат в месяц	Для решения вопроса обратитесь к персональному менеджеру	Payment

107	Неверно введен CardId. Проверьте, что такая карта была ранее привязана		Init
201	Поле PaymentId не должно быть пустым		Payment
202	Терминал заблокирован		Init; Payment
203	Параметры запроса не должны быть пустыми		Init; Payment
204	Неверный токен. Проверьте пару TerminalKey/SecretKey		Init; Payment
205	Неверный токен. Проверьте пару TerminalKey/SecretKey	Указанный терминал не найден	Init; Payment
207	Параметр DATA превышает максимально допустимый размер		Init
208	Наименование ключа из параметра DATA превышает максимально допустимый размер		Init
209	Значение ключа из параметра DATA превышает максимально допустимый размер		Init
210	Размер поля TerminalKey должен быть от {min} до {max}		Init; Payment
212	Размер поля OrderId должен быть от {min} до {max}		Init
216	Размер поля CustomerKey должен быть от {min} до {max}		Init
217	Поле PaymentId числовое значение должно укладываться в формат (<{integer} цифр>.<{fraction} цифр>)		Payment
218	Значение PAN не является числовым		Init
219	Неверный срок действия карты		Init
221	Значение CVV не является числовым		Init
222	Поле CardId числовое значение должно укладываться в формат (<{integer} цифр>.<{fraction} цифр>)		Init

233	Размер поля CardId должен быть от {min} до {max}		Init
234	Размер поля PAN должен быть от {min} до {max}		Init
236	Размер поля Token должен быть от {min} до {max}		Init; Payment
237	Размер поля PaymentId должен быть от {min} до {max}		Init
238	Размер поля ExpDate должен быть от {min} до {max}		Init
239	Размер поля CVV должен быть от {min} до {max}		Init
240	Поле Amount числовое значение должно укладываться в формат (<{integer} цифр>.<{fraction} цифр>)		Init
243	Ошибка шифрования карточных данных		Init; Payment
244	Ошибка сопоставления карточных данных		Init
248	Параметр CVV не сопоставлен		Init
250	Параметр DATA не сопоставлен		Init
251	Неверная сумма. Сумма должна быть больше или равна {value} копеек		Init
252	Срок действия карты истек		Init
255	Платеж не найден		Payment
257	Некорректное значение признака последней выплаты. Используйте значения true или false	Некорректное значение признака последней выплаты. Используйте значения true или false	Init
260	Максимальная длина номера телефона 30 символов		Init
316	Максимальная длина номера телефона 19 символов		Init
322	Передана некорректная подпись		Init; Payment
325	Транзакция не найдена		Payment
326	Неверный amount		Init

328	Должны присутствовать данные для списания и данные для пополнения		Init
403	Превышен лимит на количество пополнений в месяц		Payment
404	Превышен лимит на сумму пополнения через бесконтактные сервисы		Payment
405	Превышен лимит на сумму пополнения по виртуальной карте		Payment
406	Превышен лимит на сумму пополнения в месяц через мобильное приложение		Payment
501	Терминал не найден		Init; Payment
503	CustomerKey не найден		Init
508	Неверный номер карты		Init
515	Внутренняя ошибка		Init; Payment
600	Интернет-магазин отклонил операцию по данной карте. Обратитесь в интернет-магазин для выяснения причин отказа в платеже		Payment
619	Отсутствуют обязательные данные отправителя	Не переданы персональные данные отправителя для операции emoney2card более 15000 руб	Init
620	Проверьте сумму — она не может быть равна 0	Сумма операции не может быть равна 0	Init
623	Выплата по этому заказу уже прошла	Запрещено проводить платеж с OrderId для которого уже есть успешный платеж	Init; Payment
632	Превышен лимит на сумму операции	Лимит на сумму пополнения emoney2card. См. лимиты	Payment
633	Превышен лимит на количество переводов в день по иностранным картам	Лимит на кол-во пополнений emoney2card для карт эмитированных нерезидентами РФ за 1 отчетный день	Payment
634	Превышен лимит на сумму переводов по номеру карты в месяц	Лимит на сумму пополнения emoney2card по номеру	Payment

		карты одного получателя в отчетный месяц	
637	Не хватает данных получателя или отправителя для выплаты на иностранную карту. Проверьте заполнение	Отсутствуют персональные данные получателя/отправителя при переводе на иностранную карту	Payment
642	Проверьте номер карты	Карта не прошла проверку по алгоритму Луна	Payment
648	Магазин заблокирован или еще не активирован. Обратитесь в поддержку, чтобы уточнить детали		Init; Payment
650	Сообщите покупателю, чтобы попробовал оплатить еще раз. Если ошибка повторится — обратитесь в поддержку		Payment
651	Не получилось совершить платеж. Свяжитесь с поддержкой	Передаваемый Request_Id не найден	Payment
903	Повторите попытку позже		Payment
914	Платеж не найден		Payment
999	Попробуйте повторить попытку позже		Init; Payment
1001	Свяжитесь с банком	Свяжитесь с банком, выпустившим карту, чтобы провести платеж	Init; Payment
1003	Неверный магазин	Неверный номер магазина. Идентификатор магазина недействителен	Init; Payment
1004	Банк, который выпустил карту, считает платеж подозрительным		Payment
1005	Платеж отклонен банком, выпустившим карту	Платеж отклонен банком, выпустившим карту	Payment
1006	Платеж не прошел	Свяжитесь с банком, выпустившим карту, чтобы провести платеж	Payment
1007	Банк, который выпустил карту, считает платеж подозрительным		Payment
1008	Банк, который выпустил карту, отклонил платеж		Payment

1012	Банк, который выпустил карту, отклонил платеж		Payment
1013	Банк, который выпустил карту, отклонил платеж — сумма превышает лимит по карте	Сумма превышает лимит платежа вашего банка. Воспользуйтесь другой картой или обратитесь в банк	Payment
1014	Карта недействительна	Неправильные реквизиты — проверьте их или воспользуйтесь другой картой	Payment
1015	Неверный номер карты	Неверный номер карты	Payment
1017	Попробуйте снова или свяжитесь с банком, выпустившим карту	Попробуйте снова или свяжитесь с банком, выпустившим карту	Payment
1018	Неизвестный статус платежа		Payment
1019	Банк, который выпустил карту, отклонил платеж — сумма превышает лимит по карте		Payment
1030	Повторите попытку позже	Не получилось оплатить. Попробуйте еще раз	Payment
1033	Истек срок действия карты		Payment
1034	Попробуйте повторить попытку позже	Не получилось оплатить. Воспользуйтесь другой картой или обратитесь в банк, выпустивший карту	Payment
1038	Превышено количество попыток ввода ПИН-кода — попробуйте снова или обратитесь в банк, выпустивший карту		Payment
1039	Платеж отклонен — счет не найден		Payment
1041	Карта утеряна	Карта утеряна. Свяжитесь с банком, выпустившим карту	Payment
1043	Банк, который выпустил карту, считает платеж подозрительным		Payment
1051	Недостаточно средств на карте	Не получилось оплатить. На карте недостаточно средств	Payment
1053	Платеж отклонен — счет не найден		Payment
1054	Истек срок действия карты	Неправильные реквизиты — проверьте их или	Payment

		воспользуйтесь другой картой	
1057	Покупатель запретил такие операции для своей карты		Payment
1058	Покупатель запретил такие операции для своей карты		Payment
1059	Банк, который выпустил карту, считает платеж подозрительным		Payment
1061	Покупатель превысил лимит платежей по своей карте		Payment
1062	Банк, который выпустил карту, отклонил платеж		Payment
1063	Банк, который выпустил карту, считает платеж подозрительным		Payment
1064	Проверьте сумму		Payment
1078	Данный тип операции не поддерживается картой		Payment
1080	Плательщик ввел неверный срок действия карты		Payment
1082	Неверный CVV	Неправильные реквизиты — проверьте их или воспользуйтесь другой картой	Payment
1085	Операция успешна	Успех	Payment
1086	Платеж отклонен — не получилось подтвердить ПИН-код		Payment
1088	Банк, который выпустил карту, отклонил платеж		Payment
1089	Попробуйте повторить попытку позже	Не получилось оплатить. Попробуйте еще раз или обратитесь в банк, выпустивший карту	Payment
1091	Технические работы в банке, который выпустил карту		Payment
1092	Банк, который выпустил карту, отклонил платеж		Payment
1093	Банк, который выпустил карту, считает платеж подозрительным		Payment
1094	Банк, который выпустил карту, считает платеж подозрительным		Payment

1096	Системная ошибка	Системная ошибка	Payment
1116	Некорректная сумма выдачи	Сумма баланса меньше суммы переданной в операции выдачи	Payment
1201	Обратитесь в поддержку, чтобы уточнить детали		Payment
1202	Сумма платежа превышает лимит по разовой операции в этом магазине. Обратитесь в поддержку, чтобы уточнить детали	Для решения вопроса обратитесь к персональному менеджеру	Payment
1203	Сумма платежа превышает лимит по разовой операции или количеству операций в этом магазине. Обратитесь в поддержку, чтобы уточнить детали	Для решения вопроса обратитесь к персональному менеджеру	Payment
1204	Достигнут лимит по суточному обороту. Чтобы изменить лимит, обратитесь в поддержку	Для решения вопроса обратитесь к персональному менеджеру	Payment
1205	Магазин не принимает карты этой страны. Обратитесь в поддержку, чтобы уточнить детали	Для решения вопроса обратитесь к персональному менеджеру	Payment
1207	Сообщите покупателю, чтобы попробовал оплатить еще раз. Если ошибка повторится — обратитесь в поддержку	Для решения вопроса обратитесь к персональному менеджеру	Payment
1217	Воспользуйтесь другой картой или обратитесь к продавцу	Воспользуйтесь другой картой или обратитесь к продавцу	Payment
1218	Воспользуйтесь другой картой или обратитесь к продавцу	Воспользуйтесь другой картой или обратитесь к продавцу	Payment
1237	Получатель находится в базе ЦБ РФ, операция отклонена по 161-ФЗ: t.tb.ru/161FZ		Payment
2200	Превышено допустимое количество запросов авторизации операции		Payment
9999	Внутренняя ошибка системы		Init; Payment
9999	cert expired for terminal with id:...	У сертификата на терминале истек срок действия. Для выпуска нового сертификата воспользуйтесь инструкцией	Init; Payment

		по ссылке или обратитесь к персональному менеджеру	
926	Сделка уже закрыта		Init; Payment
927	Сделка не найдена		Init; Payment
251	Неверная сумма. Сумма должна быть больше или равна 100 копеек.		Init
935	Идентификатор получателя должен быть передан в параметре PaymentRecipientId для сделки NN		Init
936	Идентификатор сделки должен быть передан в параметре DealId для сделки NN		Init
937	Тип сделки должен быть передан в параметре CreateDealWithType для сделки NN		Init
938	Превышен размер параметра PaymentRecipientId		Init
939	Параметр получателя LevelOfConfidence должен быть передан в теле запроса, а не в объекте DATA для сделки NN		Init
940	Параметры сделки, отправителя и получателя не могут быть переданы одновременно в DATA и в теле запроса		Init
941	Параметры получателя для выплаты на иностранную карту должны быть переданы в объекте recipientAccountInfo, а не в объекте DATA		Init
942	Параметры отправителя для выплаты на иностранную карту должны быть переданы в объекте senderAccountInfo, а не в объекте DATA		Init
943	Признак финальной выплаты должен быть передан в параметре FinalPayout		Init
944	Некорректный формат параметра LevelOfConfidence		Init

-910	Некорректный идентификатор сделки		Init
-913	Нельзя провести повторное списание в безопасных сделках типа 1 к N и 1 к 1		Init
-914	Указанная сделка открыта на другом терминале		Init
-915	Некорректное значение параметра StartSpAccumulation		Init
-916	Параметры StartSpAccumulation и SpAccumulationId не могут быть переданы в одном запросе		Init
-919	Некорректное значение параметра SpFinalPayout		Init
-920	Невозможно осуществить выплату. Сделка закрыта		Init
-921	Невозможно осуществить выплату. Сумма выплаты превышает баланс сделки		Init

Таблица 6.2. Ошибки при выплате по СБП

CODE	MESSAGE	DETAILS (опционально)	Методы выплат, в которых возможно получение ошибки
99	Попробуйте повторить попытку позже		Init
102	Попробуйте повторить попытку позже		Init
1203	Воспользуйтесь другой картой или обратитесь к продавцу		Init
1202	Попробуйте повторить попытку позже		Init
1204	Попробуйте повторить попытку позже		Init
3004	Способ СБП недоступен для магазина		Init
9400	По техническим причинам пополнение невозможно		Init

9161	Пополнение данного договора невозможно		Init
9160	Отсутствуют идентификационные данные Плательщика		Init
9191	Сумма пополнения превышает разрешенный лимит		Init
9181	Отсутствует успешный авторизационный запрос с переданным идентификатором		Init
9151	Минимальная сумма 10 рублей		Init
9301	По техническим причинам пополнение невозможно		Init
9153	Запрещены пополнения на 0.00		Init
9164	Недостаточный уровень идентификации		Init
9144	Невозможно однозначно определить получателя платежа, найдено более одного контакта с данным номером телефона		Init
9143	Неверный номер телефона получателя		Init
9152	Сумма пополнения больше максимальной		Init
9501	%s are not implemented		Init
9500	Too many rows		Init
9401	Invalid authorities		Init
9403	Access is denied to the specified properties		Init
9159	Невозможно идентифицировать клиента по переданным реквизитам пополнения		Init
926	Сделка уже закрыта		Init
927	Сделка не найдена		Init

251	Неверная сумма. Сумма должна быть больше или равна 100 копеек.		Init
935	Идентификатор получателя должен быть передан в параметре PaymentRecipientId для сделки NN		Init
936	Идентификатор сделки должен быть передан в параметре DealId для сделки NN		Init
937	Тип сделки должен быть передан в параметре CreateDealWithType для сделки NN		Init
938	Превышен размер параметра PaymentRecipientId		Init
939	Параметр получателя LevelOfConfidence должен быть передан в теле запроса, а не в объекте DATA для сделки NN		Init
940	Параметры сделки, отправителя и получателя не могут быть переданы одновременно в DATA и в теле запроса		Init
941	Параметры получателя для выплаты на иностранную карту должны быть переданы в объекте recipientAccountInfo, а не в объекте DATA		Init
942	Параметры отправителя для выплаты на иностранную карту должны быть переданы в объекте senderAccountInfo, а не в объекте DATA		Init
943	Признак финальной выплаты должен быть передан в параметре FinalPayout		Init
944	Некорректный формат параметра LevelOfConfidence		Init
3004	Способ СБП недоступен для магазина.		Init

-906	Сумма возврата больше суммы покупки		Init
-907	Некорректный терминал для выплаты		Init
-910	Некорректный идентификатор сделки		Init
-913	Нельзя провести повторное списание в безопасных сделках типа 1 к N и 1 к 1		Init
-914	Указанная сделка открыта на другом терминале		Init
-915	Некорректное значение параметра StartSpAccumulation		Init
-916	Параметры StartSpAccumulation и SpAccumulationId не могут быть переданы в одном запросе		Init
-919	Некорректное значение параметра SpFinalPayout		Init
9302	При pay пополнении найден дубликат		Init
-907	Wrong terminal for Payout		Init
-920	Невозможно осуществить выплату. Сделка закрыта		Init
-921	Невозможно осуществить выплату. Сумма выплаты превышает баланс сделки		Init

7. Инструкция по получению сертификата

7.1. Сертификат КриптоПро ГОСТ:

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

Для получения УНЭП необходимо обратиться к вашем 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.

7.2. Сертификат RSA:

Воспользуйтесь инструкцией, которая доступна по ссылке.

8. Методы работы с электронными сертификатами

8.1. Метод AddCertificate

Описание: Добавляет новый сертификат для терминала для подписи запросов. Подробнее о том, как получить сертификат см. Инструкция по получению сертификата.

Для терминала может быть загружено несколько сертификатов. Проверка подписи происходит с помощью сертификата, серийный номер которого указан в запросе в поле X509SerialNumber.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/AddCertificate
Боевой URL: https://securepay.tinkoff.ru/e2c/v2/AddCertificate
*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Content-Type: multipart/form-data

Таблица 9.1.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком.
Certificate	File	Да	Сертификат в формате .cer *
DigestValue	String	Да	Значение хеша в Base64
SignatureValue	String	Да	Значение подписи в Base64
X509SerialNumber	String	Да	Серийный номер сертификата

***** Сертификат должен соответствовать требованиям:

1) Файл не поврежден

2) Формат файла ".cer".

3) Загружается именно открытая часть ключа

4) Алгоритм подписи подходит под используемые:

- |"1.2.643.7.1.1.3.2" | "Алгоритм цифровой подписи ГОСТ Р 34.10-2012 для ключей длины 256 бит"
- |"1.2.643.7.1.1.3.3" | "Алгоритм цифровой подписи ГОСТ Р 34.10-2012 для ключей длины 512 бит"

Алгоритм подписи для RSA-сертификата:

- sha 256RSA

5) Содержимое файла начинается со строки "-----BEGIN CERTIFICATE-----", заканчивается "-----END CERTIFICATE-----"

6) Сертификат должен быть закодирован с помощью Base-64 (см. 8.4. Сохранение сертификата в кодировке Base-64)

Пример запроса:

TerminalKey: TerminalKeyE2C
Certificate: <Действующий сертификат CryptoPro>
DigestValue: "s9GDxXSwaBpZr/kOs5zPHdlEV4XQVdeOOQIJwlAQ5LQ="
SignatureValue:
"ZrjXi4z3LkxS3OvDh+Df1l5YuMaTk6rTxoxBGF3hjpIDs6NCtBpVu+ChcJaa3vwXRILA9RXtqnGPbVp27XOYVYEyO/d
yimBrM/MqeWU6kdjYBARu/NssHVp8U/1D5ympsycz7Agfl13tSuOjCPZ2LSJWG4eaKPouOCWG2BsDYTs1YJ+TljXp
9dxSfx5EuVxhOC+F6HazwJhEUNkWZwb5fCVROx6d85OV93wv3Yz+H7Nt3U4anhSHDmPANbZJTnjFUw7fQE16tUl
OGrPXOAVeZD3jOpdJJiFpc/D6crGzpmvjy7frPSHIPusNtnl5BfOqvbEN9ZidwqBJPRFvnkjrOg=="
X509SerialNumber: a44c3fed7bd693226466e0b0c9a836d1855a7

Формат ответа: **JSON**

Ответ

Таблица 9.1.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком.
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки. «0» - если успешно
X509SerialNumber	String	Да	Серийный номер сертификата в десятичном формате (dec)
StartDate	String	Да	Дата и время начала срока действия сертификата в формате ДД.ММ.ГГГГ ЧЧ:ММ:СС
ExpirationDate	String	Да	Дата и время окончания срока действия сертификата в формате ДД.ММ.ГГГГ ЧЧ:ММ:СС
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа:


```
{
    "Success": true,
    "ErrorCode": "0",
    "TerminalKey": "TerminalKeyE2C",
    "X509SerialNumber": "276529428848014209313654660773503233854207008 2",
    "StartDate": "06.06.2022 15:21:31",
    "ExpirationDate": "06.09.2022 15:31:31"
}
```

8.2. Метод UpdateCertificateStatus

Описание: Меняет статус сертификата для подписи запросов.

Запрос

Тестовый URL*: https://rest-api-test.tinkoff.ru/e2c/v2/UpdateCertificateStatus
Боевой URL: https://securepay.tinkoff.ru/e2c/v2/UpdateCertificateStatus

*Для возможности отправки запросов на тестовую среду напишите на почту acq_help@tbank.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 9.2.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
X509SerialNumber	String	Да	Серийный номер сертификата в формате dec из ответа по методу AddCertificate
SetStatus	String	Да	Новый статус сертификата. Возможные значения: • Active – активен • Blocked - заблокирован
DigestValue	String	Да	Значение хеша в Base64
SignatureValue	String	Да	Значение подписи в Base64
X509SerialNumber	String	Да	Серийный номер сертификата

Пример запроса:

{
"TerminalKey": "TerminalKeyE2C",
"X509SerialNumber ": "276529428848014209313654660773503233854207008 2",
"SetStatus": "Blocked",
DigestValue: "s9GDxXSwaBpZr/kOs5zPHdlEV4XQVdeOOQIJwlAQ5LQ=",
SignatureValue:
"ZrjXi4z3LkxS3OvDh+Df1l5YuMaTk6rTxoxBGF3hjpIDs6NCtBpVu+ChcJaa3vwXRILA9RXtqnGPbVp27XOYVYEyO/dyimBr
M/MqeWU6kdjYBARu/NssHVp8U/1D5ympsycz7Agfl13tSuOjCPZ2LSJWG4eaKPouOCWG2BsDYTs1YJ+TljXp9dxSfx5EuV
xhOC+F6HazwJhEUNkWZwb5fCVROx6d85OV93wv3Yz+H7Nt3U4anhSHDmPANbZJTnjFUw7fQE16tUlOGrPXOAVeZD3jO
pdJJiFpc/D6crGzpmvjy7frPSHIPusNtnl5BfOqvbEN9ZidwqBJPRFvnkjrOg==",
X509SerialNumber: a44c3fed7bd693226466e0b0c9a836d1855a7
}

Формат ответа: JSON

Ответ

Таблица 11.2.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Площадке Банком
Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки. «0» - если успешно
Message	String	Нет	Краткое описание ошибки
Details	String	Нет	Подробное описание ошибки

Пример ответа

{
"TerminalKey": "TerminalKeyE2C",
"Success": true,
"ErrorCode": "0"
}

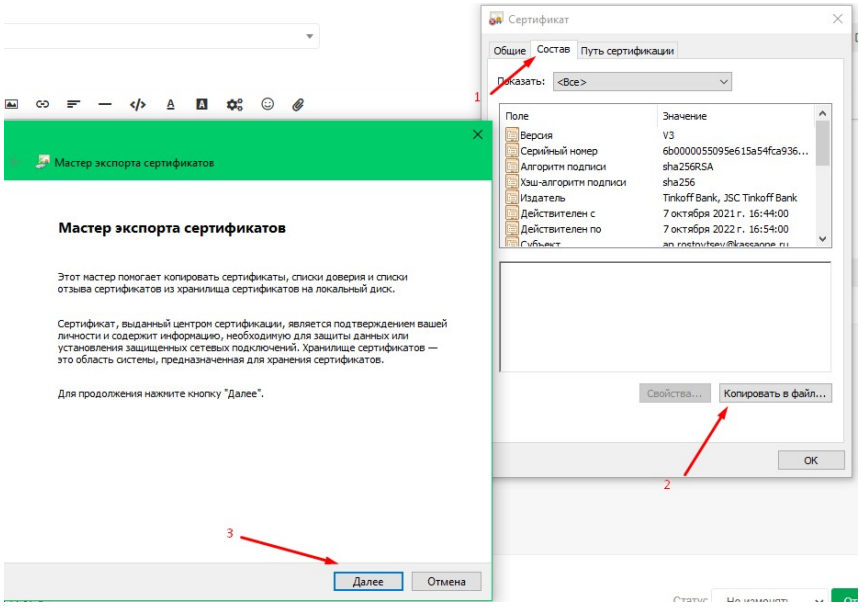
8.3. Подпись запроса с помощью токена

Для генерации подписи используется пароль (см. Параметры выплат параметр Password) из личного кабинета мерчанта.

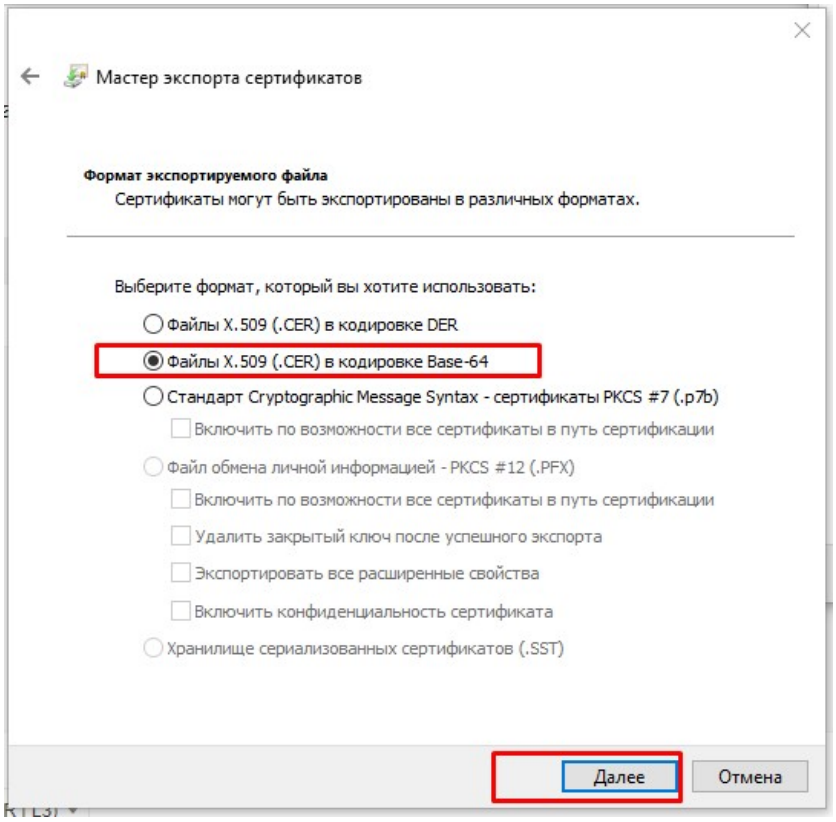
1. Собрать массив всех передаваемых параметров в виде пар Ключ-Значение:
   [{"TerminalKey","TestB"}, {"PaymentId","20150"}]
2. Добавить в массив пару (Password, значение):
   [{"TerminalKey","TestB"}, {"PaymentId","20150"}, {"Password","Dfsfh56dgKl"}]
3. Отсортировать массив по Ключам по алфавиту:
   [{"Password", "Dfsfh56dgKl"}, {"PaymentId","20150"}, {"TerminalKey","TestB"}]
4. Конкатенировать значения всех пар:
   Dfsfh56dgKl20150TestB
5. Вычислить SHA-256 от полученного в предыдущем пункте значения и записать значение в Token.
   Формирование подписи запроса Token завершено.

8.4. Сохранение сертификата в кодировке Base-64

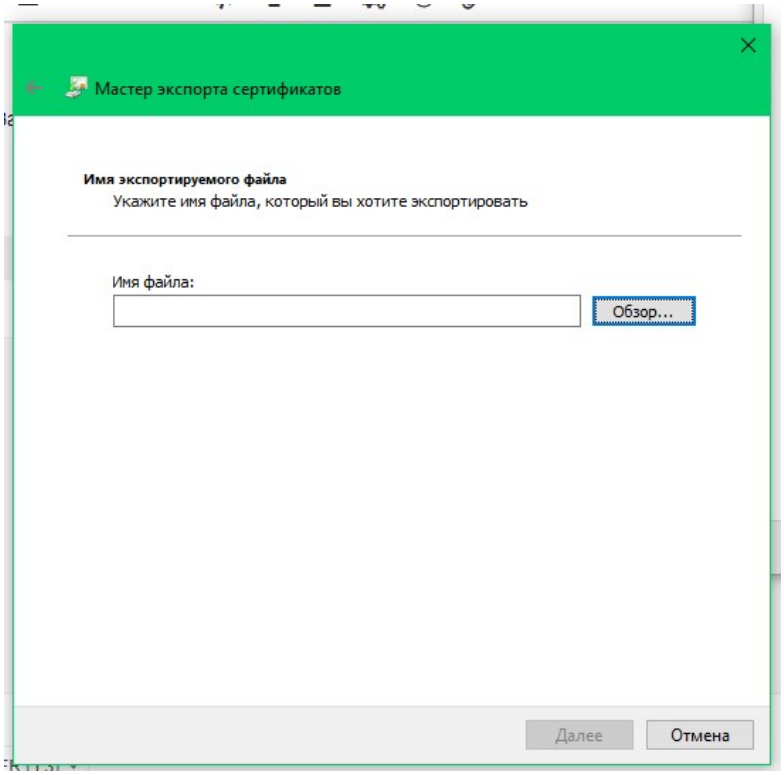
Запускаем сертификат на вкладке «Состав» нажимаем «Копировать в файл»



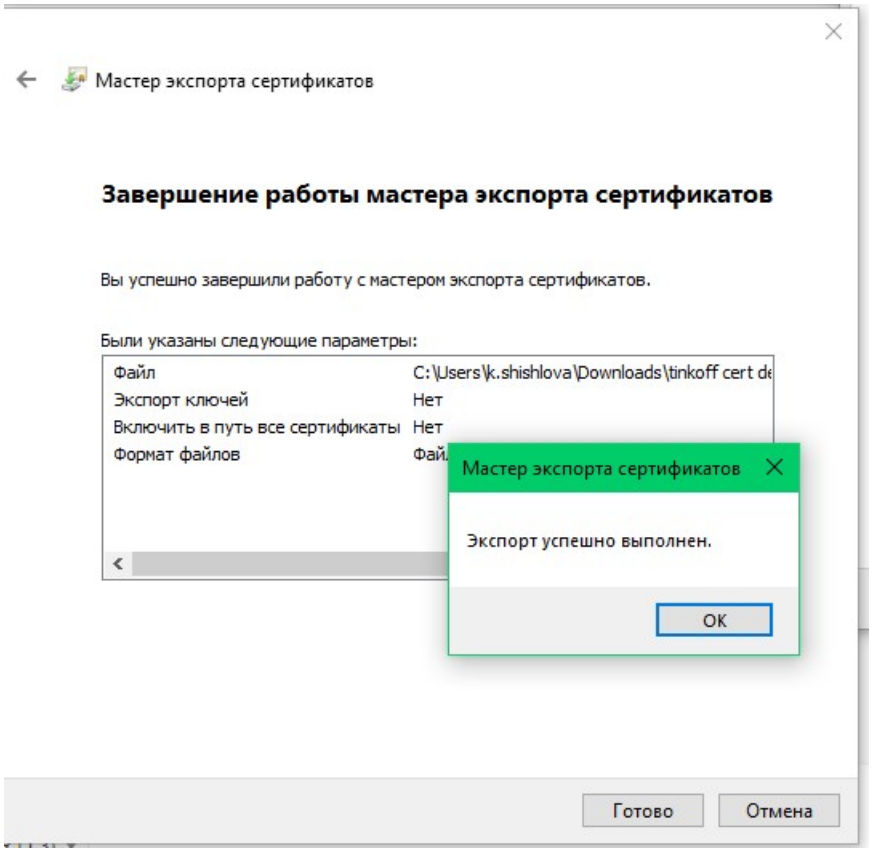
В окне выбираем формат “Файлы X.509 (.CER) в кодировке Base-64”



Выбираем путь сохранения файла и затем нажимаем на кнопку «Далее».



Затем на кнопку «Готово». Результат на скрине



9. Карты для проведения тестирования

Успешная привязка/выдача	5000000000000447 expDate: любая действующая дата в формате 11/24 cvv: любой набор из 3 цифр
Успешная привязка authRC=1057, message='Покупатель запретил такие операции для своей карты' в ответ на метод Payment	5000000000000553 expDate: любая действующая дата в формате 11/24 cvv: любой набор из 3 цифр
Тестирования Выплат СБП Успешный флоу	1. Phone: 79066589133 SbpMemberId: 100000000012 2. Phone: 79066589133 SbpMemberId: 100000000004
Тестирования Выплат СБП. Неуспешный флоу Получение ошибки: -955	Phone 79066589133 SbpMemberId 100000000013
Тестирования Выплат СБП. Неуспешный флоу Получение ошибки: 9143	Phone 79066589134 SbpMemberId 100000000004