Описание сервиса Tinkoff A2C_V2

12.05.2023

Тинькофф Банк

Версия документа: 1.15

Оглавление

История изменений

Версия	Описание	Дата
1.0	Документ создан	15.06.2021
1.1	Добавлен параметр account в метод init в параметр DATA	31.01.2022
1.2	Добавлен метод GetAccountInfo Обновлена обязательность полей для переводов на иностранные карты в методе Init	18.02.2022
1.3	Исправлено значение “TerminalKey” в примерах запросов и ответов	11.05.2022
1.4	Добавлено примечание по предоставлению доступа к тестовому URL	12.05.2022
1.5	Изменено описание полей CheckType в методе AddCard. Удалены статусы LOOP_CHECKING и LOOP_CHECKED и обновлено описание статусов AUTHORIZING и AUTHORIZED.	18.08.2022
1.6	Добавлен раздел методы работы с электронными сертификатами.	29.08.2022
1.7	Изменили обязательность параметра rebill_id в методе GetCardList	30.08.2022
1.8	Добавлен список тестовых карт.	14.09.2022
1.9	Добавлен раздел методы работы с электронными сертификатами.	16.09.2022
1.10	Удален PaymentUrl в ответе Init	03.11.2022
1.11	Добавлено описание get-параметров которые могут вернуться при использовании метода AddCard	30.01.2023
1.12	Добавлен параметр paymentPurposeDetails в параметры объекта DATA метода Init	31.01.2023
1.13	Добавлен список внешних сетей, используемых при отправке нотификаций. Обновлен список портов, которые можно использовать в Notification URL	13.02.2023
1.14	Добавлена информация о том, что запросы на тестовую среду необходимо осуществлять с боевого терминала.	23.03.2023
1.15	Добавлена информация о нотификациях о привязках счета по номеру телефона через СБП	12.05.2023

Термины и сокращения

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

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://oplata.tinkoff.ru Он совпадает с паролем от терминала интернет-эквайринга

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

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

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

Пример:

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 выплат

2.1 Общая информация

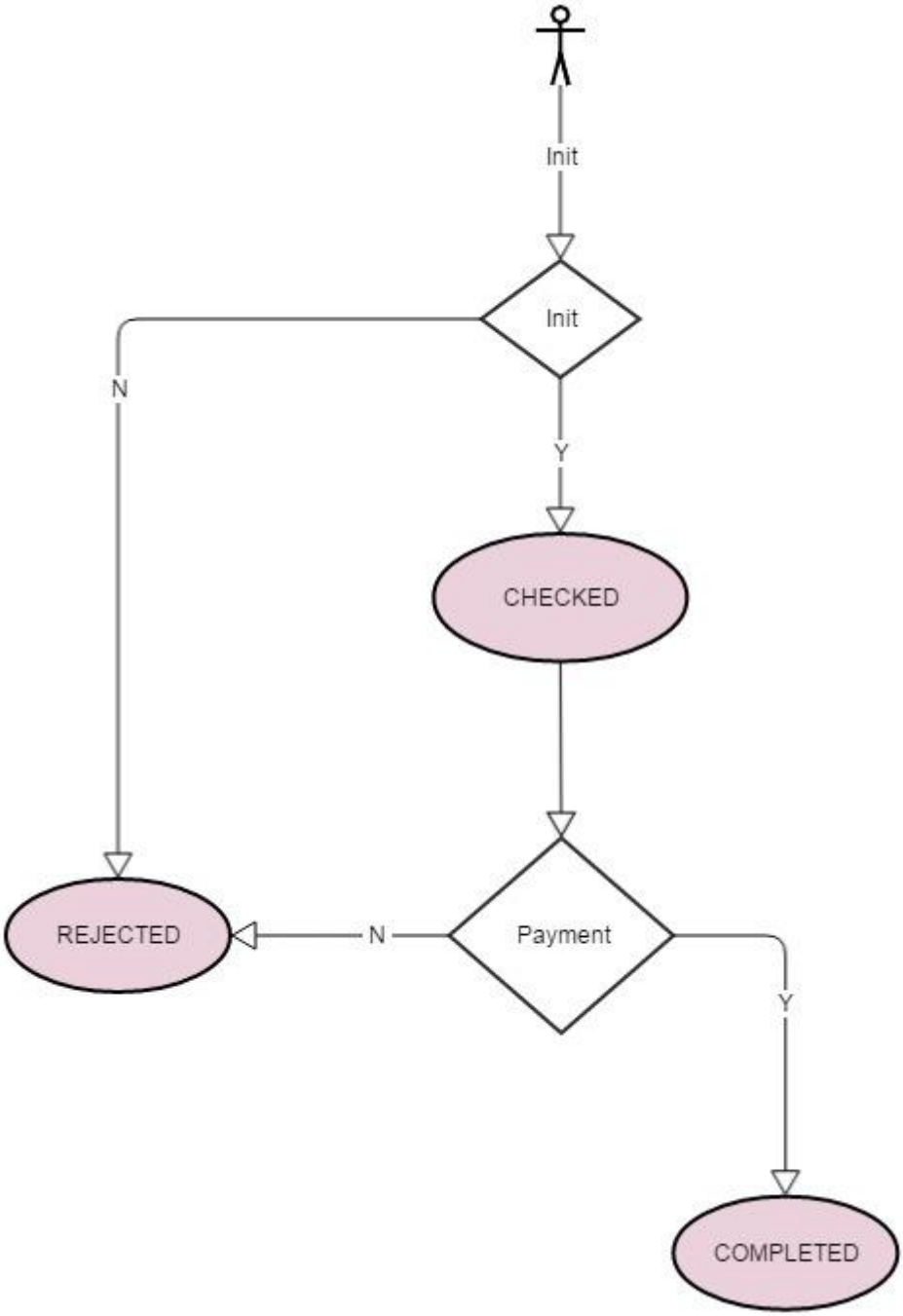
Выдача средств осуществляется вызовом методов с передачей параметров методом GET или POST в зависимости от метода. Все методы и передаваемые параметры являются чувствительными к регистру. Порядок передачи параметров в запросе значения не имеет.
Для POST запроса в заголовке должен присутствовать **Content-Type: 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@tinkoff.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

2.2 Схема проведения платежа

На схеме показаны статусы платежа и возможные методы, которые могут быть вызваны, если платеж находится в данном статусе. На стрелках Y – обозначает успешное выполнение метода, N – что при обработке метода произошла ошибка.


Init
Init
N
Y
CHECKED
REJECTED
N
Payment
Y
COMPLETED

2.3 Метод Init

Описание: Инициирует платежную сессию.

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@tinkoff.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала.

Метод: POST

Таблица 2.3.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком
OrderId	String	Да	Уникальный номер заказа в системе Продавца
IP	String	Нет	IP-адрес клиента
CardId	String	Да	Идентификатор карты пополнения привязанной с помощью метода AddCard.
Amount	Number	Да	Сумма в копейках
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
CustomerKey	String	Нет	Идентификатор покупателя в системе Продавца. Если передается и Банком разрешена автоматическая привязка карт к терминалу, то для данного покупателя будет осуществлена привязка карты. Тогда в нотификации на AUTHORIZED будет передан параметр CardId (подробнее см. метод GetGardList)

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 «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

* Если у терминала включена опция привязки покупателя после успешной выплаты и передается параметр CustomerKey, то в передаваемых параметрах DATA могут присутствовать параметры команды AddCustomer. Если они присутствуют, они автоматически привязываются к покупателю. Например, если указать:

```
"DATA": {
      "Phone": "+71234567890",
      "Email": "a@test.ru"
   }
```

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

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

Список параметров, передаваемых в объект DATA

Таблица 2.3.2. Данные отправителя

Наименование	Обязательность	Описание
s_lastname	Нет	Фамилия отправителя платежа
s_firstname	Нет	Имя отправителя платежа
s_middlename	Нет	Отчество отправителя платежа
s_dateOfBirth	Нет	Дата рождения отправителя платежа в формате ДД.ММ.ГГГГ

Наименование	Обязательность	Описание
s_citizenship	Нет	Гражданство отправителя (3-х буквенный ISO-код)
s_passportSeries	Нет	Серия паспорта
s_pasportNumber	Нет	Номер паспорта
s_passportIssueDate	Нет	Дата выдачи паспорта в формате ДД.ММ.ГГГГ
s_accountNumber	Нет	Номер карты или расчетного счета отправителя, с которого производится зачисление на карту

Таблица 2.3.3. Данные получателя

Наименование	Обязательность	Описание
r_lastname	Нет	Фамилия получателя платежа
r_firstname	Нет	Имя получателя платежа
r_middlename	Нет	Отчество получателя платежа
agreement_number	Нет	Номер договора займа

Таблица 2.3.4. Данные перевода

Наименование	Обязательность	Описание
t_domestic	Да	Направление перевода. 0 – международный, 1 – внутри страны

Обязательные параметры для переводов на иностранные карты.

Параметры передавать в объект DATA. При переводах на российские карты так же можно передавать данные об отправителе и получателе, но в этом случае параметры не являются обязательными.

Таблица 2.3.5. Данные отправителя

Наименование	Обязательность	Описание
s_lastname	Да	Фамилия отправителя платежа
s_firstname	Да	Имя отправителя платежа
s_dateOfBirth	Да	Дата рождения отправителя платежа в формате ДД.ММ.ГГГГ
s_accountNumber	Да	Номер карты или расчетного счета отправителя, с которого производится зачисление на карту

Таблица 2.3.6. Адрес отправителя

Наименование	Обязательность	Описание
s_address	Да	Полный адрес отправителя
s_addressZip	Да	Почтовый индекс
s_addressCountry	Да	Страна. Указывается в формате ISO 3166-1 Numeric
s_addressCity	Да	Город\Населенный пункт

Таблица 2.3.7. Данные получателя

Наименование	Обязательность	Описание
r_lastname	Да	Фамилия получателя платежа
r_firstname	Да	Имя получателя платежа

Таблица 2.3.8. Данные перевода

Наименование	Обязательность	Описание
t_domestic	Да	Направление перевода. 0 – международный, 1 – внутри страны

paymentPurposeDetails	Нет	Дополнительные данные по операции для отображения в назначении платежа*. Не более 75 символов.

* Для возможности передачи параметра необходимо обратиться к вашему менеджеру.

Пример запроса:


```
{
   "TerminalKey":  "TinkoffBankTest",
   "Amount":  "1751",
   "OrderId":  "autoOrd1615285401068DELb",
   "CardId":  "70000000857",
   "DATA": {
      "Phone": "+71234567890",
      "Email": "a@test.ru"
   },
  "DigestValue":  "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
   "SignatureValue":  "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wb
KimIBD7Ov8j8eP/tKn7/g==",
      "X509SerialNumber":  "2613832945"
}
```


Формат ответа: **JSON**

Ответ

Таблица 2.3.9. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком
Amount	Number	Да	Сумма в копейках
OrderId	String	Да	Уникальный номер заказа в системе Продавца
Success	bool	Да	Успешность операции
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": "TinkoffBankTest",
"Status": "CHECKED",
"PaymentId": "2353039",
"OrderId": "PaymentTestN",
"Amount": 100
}

Статус платежа:

при успешном сценарии: **CHECKED**

при неуспешном: **REJECTED**

2.4 Метод Payment

Описание: Производит пополнение карты.

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@tinkoff.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 2.4.1. Параметры запроса

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

Пример запроса:

```
{
    "TerminalKey": " TinkoffBankTest ",
    "PaymentId": "700000085140",
    "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
    "SignatureValue": "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
    "X509SerialNumber": "2613832945"
}
```

Формат ответа: **JSON**

Ответ

Таблица 2.3.7. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком
OrderId	String	Да	Уникальный номер заказа в системе Продавца
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:

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

Статус платежа:

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

2.5 Метод GetState

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@tinkoff.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 Продавцу 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

Пример запроса:


```
{
  "TerminalKey": "TinkoffBankTest ",
  "PaymentId": "700000085101",
  "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
  "SignatureValue": "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMel12ERBSlOnzy6YFqMaa5nRb9ZrK9wb
KimIBD7Ov8j8eP/tKn7/g==",
  "X509SerialNumber": "2613832945"
}
```


Формат ответа: **JSON**

Ответ

Таблица 2.5.2. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком.
OrderId	String	Да	Уникальный номер заказа в системе Продавца.
Success	bool	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:

{
"TerminalKey":"TinkoffBankTest",
"OrderId":"21057",
"Success":true,
"Status":"COMPLETED",
"PaymentId":"10063",
"ErrorCode":"0"
}

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

Статус	Промежуточный?	Значение
UNKNOWN*	Да	Статус не определен
CREDIT_CHECKING**	Да	На стадии обработки

* при получении Status=UNKNOWN необходимо вызывать метод GetState каждую минуту в течение 10 минут.

** 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.

3. Алгоритм формирования подписи запроса (Signature)

3.1 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","TinkoffBankTest"},{"PaymentId","20150"}]

2) Сортировать по Ключам:

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

3) Конкатенировать значения:

20150TestB

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://acdn.tinkoff.ru/static/documents/rsa-crypto-lib-java-mapi.zip

3.2 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","TinkoffBankTest"},{"PaymentId","20150"}]

2) Сортировать по Ключам:

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

3) Конкатенировать значения:

20150TestB

4) Вычислить хэш-сумму по ГОСТ Р 34.11-2012 256 и записать значение в DigestValue (должно получится 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.8.**

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

Язык	Ссылка	Версия Крипто Про
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.

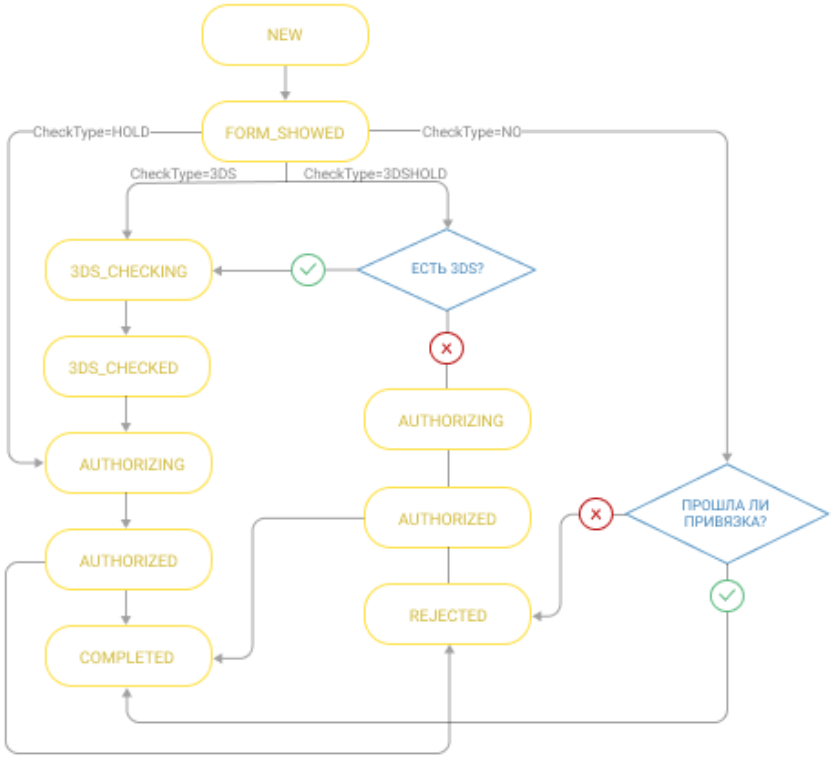


Рисунок 1. Статусная схема привязки карт

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

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@tinkoff.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 Продавцу Банком.
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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

Пример запроса:

{
"TerminalKey":" TinkoffBankTest ",
"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 Продавцу при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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":"TinkoffBankTest","CustomerKey":"Customer1"}

4.3 Метод GetCustomer

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@tinkoff.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 Продавцу Банком.
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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

Пример запроса:


```
{
    "TerminalKey":" TinkoffBankTest ",
    "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 Продавцу при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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:

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

4.4 Метод RemoveCustomer

Описание: Удаляет 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@tinkoff.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Метод: POST

Таблица 4.4.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком.
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
IP	String	Нет	IP-адрес запроса
DigestValue	String	Да	Значение хеша в Base64
SignatureValue	String	Да	Значение подписи в Base64
X509SerialNumber	String	Да	Серийный номер сертификата

Пример запроса:


```
{
    "TerminalKey":" TinkoffBankTest ",
    "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 Продавцу при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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":"TinkoffBankTest",
"CustomerKey":"Customer1"
}

4.5 Метод GetCardList

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@tinkoff.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 Продавцу Банком.
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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

Пример запроса:


```
{
    "TerminalKey":" TinkoffBankTest ",
    "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	String	Нет	Идентификатор рекуррентного платежа
CardType	Enum	Да	Тип карты: – 0 - карта списания; – 1 - карта пополнения; – 2 - карта пополнения и списания
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

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.

Примечание: Возможно настроить отображение кода ошибки в Fail и Success Add Card URL.

Пример формы банка: https://rest-api-test.tinkoff.ru/html/payForm/e2c/html/e2cError.html?CardId=870051&CustomerKey=testRegress6&ErrorCode=101

Для включения настройки необходимо сообщить об этом при передаче Success и Fail Add Card URL Банку.

Важно понимать, что это не единственные параметры, которые могут вернуться. Пример основных параметры на страницах успеха/неуспеха:

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

Коды ошибок указаны на сайте:

https://oplata.tinkoff.ru/landing/develop/documentation/code_error

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@tinkoff.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 Продавцу Банком
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца

Наименование	Тип	Обязательность	Описание
CheckType	String	Нет	Если CheckType не передается, автоматически проставляется значение NO. Возможные значения: 1. NO – сохранить карту без проверок. Rebill ID 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. Для карт МИР с 3DS v1 выполняется списание, а затем отмена на 1р. Для прочих карт 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)
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

*Для карт МИР с 3DS v1 выполняется списание, а затем отмена на 1р.

Пример запроса:

{
"TerminalKey":"TinkoffBankTest",
"CustomerKey":"TestCustomer10",
"IP":"192.168.40.74",
"CheckType": "NO",
"DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
"SignatureValue": "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",
"X509SerialNumber": "2613832945"
}

Формат ответа: **JSON**

Ответ

Таблица 4.6.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Продавцу при заведении терминала
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
RequestKey	String	Да	Идентификатор запроса на привязку карты
PaymentURL	String	Да	Ссылка на страницу привязки карты. На данную страницу необходимо переадресовать клиента для привязки 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":"TinkoffBankTest",
"CustomerKey":"Customer1",
"RebillId":"123456",
"PaymentURL":"https://securepay.tinkoff.ru/e2c/f36d8e7f-4bc6-4250-9f64-7fe986d3dc62",

"RequestKey": "8de92934-26c9-474c-a4ce-424f2021d24d"
}

4.7 Метод RemoveCard

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@tinkoff.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	Идентификатор терминала, выдается Продавцу Банком
CardId	Number	Да	Идентификатор карты в системе Банка
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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

Пример запроса:


```
{
    "TerminalKey":"TinkoffBankTest",
    "CustomerKey":"TestCustomer10",
    "CardId": 1234,
    "IP":"192.168.40.74",
    "DigestValue": "qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",
    "SignatureValue": "rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7
Ov8j8eP/tKn7/g==",
    "X509SerialNumber": "2613832945"
}
```


Формат ответа: **JSON**

Ответ

Таблица 4.7.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Платежный ключ, выдается Продавцу при заведении терминала
CardId	Number	Да	Идентификатор карты в системе Банка
CustomerKey	String	Да	Идентификатор покупателя в системе Продавца
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":"TinkoffBankTest",
"CustomerKey":"Customer1"
}

5. Нотификации о привязке карты

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

Нотификации по http(s)

Описание: Тинькофф Оплата может уведомлять магазин об успешных/ошибочных привязках карты. Для этого необходимо написать своему менеджеру, либо на eacq_accounts@tinkoff.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 Тинькофф Оплата, для отправки нотификаций:
91.194.226.0/23
91.218.132.0/22
212.233.80.0/22
*Для корректной работы нотификаций необходимо добавить данные сети в исключения сетевых фильтров или других видов защиты в случае их использования.

URL: Notification URL
Метод: POST

Таблица 5.1 Параметры нотификации

Наименование	Тип	Описание
TerminalKey	String	Идентификатор терминала, выдается Продавцу Банком
CustomerKey	String	Идентификатор покупателя в системе Продавца
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":"TinkoffBankTest",
"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. Нотификация о привязке счета по номеру телефона

Уведомляет об успешных/ошибочных выполнениях привязок счета по номеру телефона получателя.
Нотификация отправляется по http(s)*.
*Отправка нотификаций о привязке счета по номеру телефона выполняется аналогично п.5

Запрос
URL: Notification URL
Метод: POST

Таблица 6.1. Параметры запроса на Notification URL

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком
CustomerKey	String	Да	Идентификатор получателя в системе Продавца
RequestKey	String	Да	Идентификатор запроса на привязку карты
Success	bool	Да	Успешность операции
Status	String	Да	Статус операции
PaymentId	Number	Да	Уникальный идентификатор транзакции в системе Банка
Pan	String	Нет	Маскированный номер телефона. Пример: +7(012)***-**-89. Pattern: ^\+\d{1}\(\d{3}\)*{3}-*{2}-\d{2}$
BankName	String	Нет	Наименование банка получателя
ErrorCode	String	Да	Код ошибки, «0» - если успешно
AccountToken	String	Нет	Токен привязки
NotificationType	String	Да	Константа. «LINKPHONE»
Token	String	Да	Подпись запроса. Формируется по такому же принципу, как и в случае запросов в банк

7. Метод GetAccountInfo

Описание: Метод предназначен для получения остатка по счету для A2C.

Запрос

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

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

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

Метод: POST

Таблица 6.1 Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала
DigestValue	String	Да	Значение хеша в Base64
SignatureValue	String	Да	Значение подписи в Base64
X509SerialNumber	String	Да	Серийный номер сертификата

Пример запроса:

{

"TerminalKey": "TinkoffBankTest",
"DigestValue":"qfeohMmrsEvr4QPB8CeZETb+W6VDEGnMrf+oVjvSaMU=",

SignatureValue":"rNTloWBbTsid1n9B1ANZ9/VasWJyg6jfiMeI12ERBSlOnzy6YFqMaa5nRb9ZrK9wbKimIBD7Ov8j8eP/tKn7/g==",

"X509SerialNumber":"2613832945">,

}

Формат ответа: **JSON**

Ответ

Таблица 6.2 Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала

Success	bool	Да	Успешность операции
ErrorCode	String	Да	Код ошибки, «0» - если успешно
Message	String	Нет	Описание ошибки, если ErrorCode<>0
Details	String	Нет	Детальное описание ошибки, если ErrorCode<>0
Total	String	Нет	Всего
Hold	String	Нет	Заблокировано
Available *	String	Нет	Доступно
Token	String	Да	Подпись запроса.

(*) Основным параметром является Available, который считается по формуле:

Available(доступно) = Total(всего) - Hold(заблокировано) + Over(овердрафт)

Пример ответа:

{
"TerminalKey": "TinkoffBankTest",
"Success": true,
"ErrorCode": "0",
"Total": "1477014,66",
"Hold": "0",
"Available": "1477014,66"
"Token":"f2fdd7fec8225872590e1558b7ea258c75df8f300d808006c41ab540dd7514d9"
}

8. Коды ошибок, передаваемые на FailURL

Подробный список ошибок с описанием представлен в отдельном документе по следующей ссылке.

9. Этап тестирования

Для тестирования методов используйте терминал с приставкой E2CDEMO. Запросы по данному терминалу необходимо отправлять на боевой URL: https://securepay.tinkoff.ru/e2c/v2/

Сценарии для тестирования на E2CDEMO терминале:

1. Добавить клиента (методом AddCustomer)
2. Удалить клиента (методом RemoveCustomer)
3. Привязать карту к клиенту (AddCard)
4. Удалить карту (RemoveCard)
5. Показать списков всех карт клиента (GetCardList)
6. Проверить нотификации после привязки
7. Провести успешную выдачу на привязанную карту (Payment)
8. Провести выдачу с ошибкой

Тестовые карты*:
5000000000000447 - Успешная привязка/выдача
5000000000000553 - Успешная привязка, authRC=1057,message=Transaction not permitted to card в ответ на метод Payment

Срок действия: 11/22

CVV: 123

*Данные карты необходимо использовать на боевой среде на DEMO-терминале. Список карт для кейсов на тестовой среде см в 9. Список тестовых карт.

10. Список тестовых карт

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

Таблица 9.1 Список тестовых карт для оплаты через протокол 3ds2.0

Поведение карты	TransStatus*	Описание	PAN
Ошибка оплаты Недостаточно средств	Нет	Нет	2201382000000831 expDate: 12/25 cvv: 123
Успешная оплата 3ds2.0 Frictionless Flow	Нет	AUTHENTICATION_SUCCESSFUL Успешное прохождение аутентификации без ввода пароля	2201382000000013 expDate: 12/25 cvv: 123
Успешная оплата 3ds2.0 Challenge Flow	C	CHALLENGE_REQUIRED Требуется полное прохождение 3DS с редиректом на acsURL. Открытие формы для ввода одноразового пароля (OTP)	2201382000000047 expDate: 12/25 cvv: 123 Метод аутентификации на ACS: Static Passcode. Ввести верный пароль 1qwezxc
Ошибка оплаты 3ds2.0 Restricted	R	ACCOUNT_VERIFICATION_REJECTED Эмитент отклонил аутентификацию	2201382000000005 expDate: 12/25 cvv: 123
Ошибка оплаты Frictionless Flow Not Authenticated	N	NOT_AUTHENTICATED Карта не поддерживает 3DS	2201382000000021 expDate: 12/25 cvv: 123

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

Таблица 9.2 Список тестовых карт для оплаты без 3ds

Поведение карты	TransStatus*	Описание	PAN
Успешная оплата	Нет	-	2200770239097761 expDate: 12/25 cvv: 123
Ошибка оплаты Недостаточно средств	Нет	-	4249170392197566 expDate: 12/25 cvv: 123
Ошибка оплаты Ошибка при списании	Нет	-	5586200071492075 expDate: 12/25 cvv: 123

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

11. Инструкция по получению сертификата

11.1 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 АО «Тинькофф 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.

11.2 Сертификат RSA:

Воспользуйтесь инструкцией, которая доступна по ссылке

12. Методы работы с электронными сертификатами

12.1 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@tinkoff.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Запрос AddCertificate должен быть подписан с помощью токена.

Метод: POST

Content-Type: multipart/form-data

Таблица 11.1.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу Банком.
Certificate	File	Да	Сертификат в формате .cer*
Token	String	Да	Подпись запроса

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. 11.4 Сохранение сертификата в кодировке Base-64)

Пример запроса:
TerminalKey : TinkoffBankTest
Certificate : <Действующий сертификат CtyptoPro>
Token : a44c3fed7bd693226466e0b0c9a836d1855a777543f8f7fe7152b15e31fb0d7b

Формат ответа: **JSON**

Ответ

Таблица 11.1.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу 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": " TinkoffBankTest",
```

"X509SerialNumber": "276529428848014209313654660773503233854207008 2",
"StartDate": "06.06.2022 15:21:31",
"ExpirationDate": "06.09.2022 15:31:31"
}

12.2 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@tinkoff.ru с просьбой добавить ваши IP в WL. Запрос необходимо отправлять с боевого терминала

Запрос UpdateCertificateStatus должен быть подписан с помощью токена.

Метод: POST

Таблица 11.2.1. Параметры запроса

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу 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
Token	String	Да	Подпись запроса

Пример запроса:

{
"TerminalKey": " TinkoffBankTest ",
" X509SerialNumber ": "276529428848014209313654660773503233854207008 2",
" SetStatus": "Blocked",
"Token":"2c31b9c49d503b72eer50d301ba6510345195b33df0cbf79f39f6a8766d84a59"
}

Формат ответа: JSON

Ответ

Таблица 11.2.2. Параметры ответа

Наименование	Тип	Обязательность	Описание
TerminalKey	String	Да	Идентификатор терминала, выдается Продавцу 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": "TinkoffBankTest",
"Success": true,
"ErrorCode": "0"
}

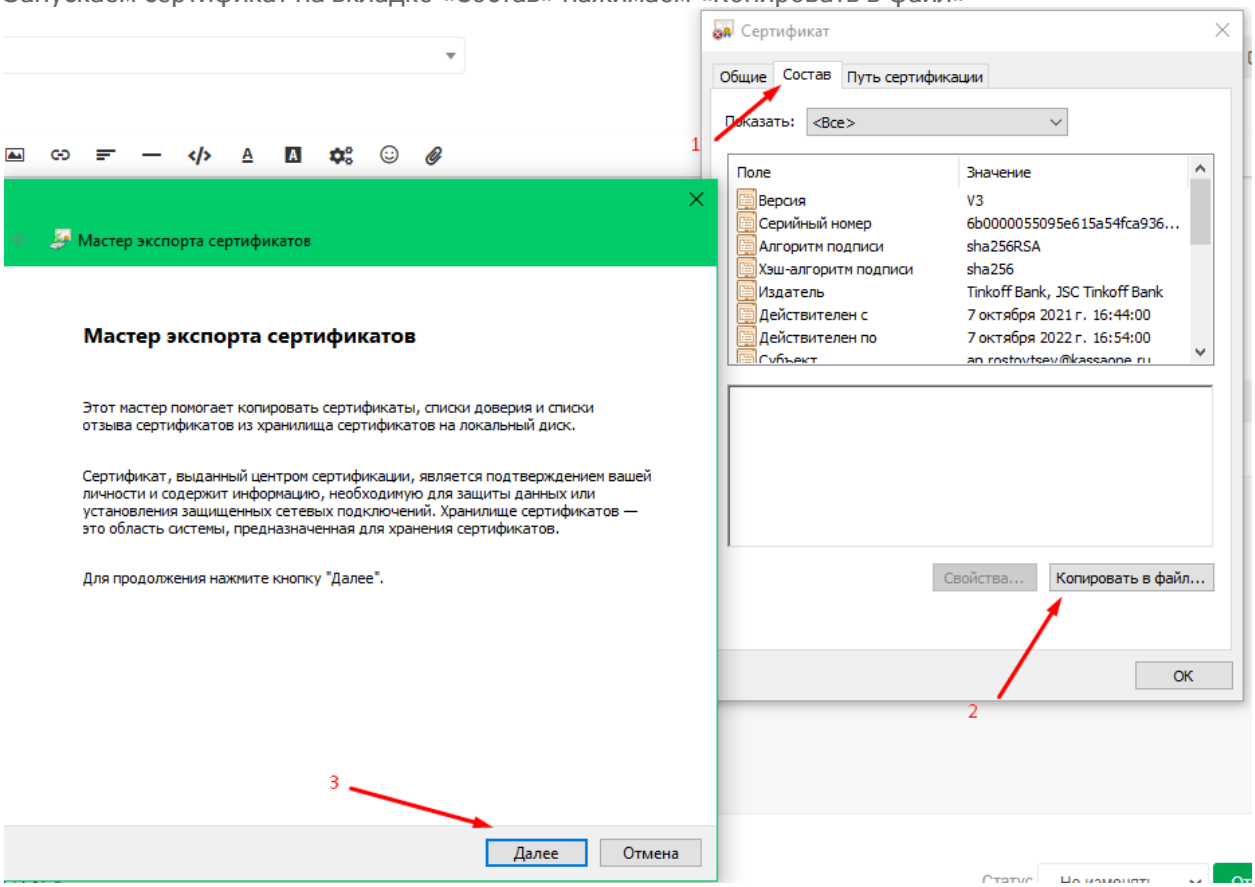
12.3 Подпись запроса с помощью токена

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

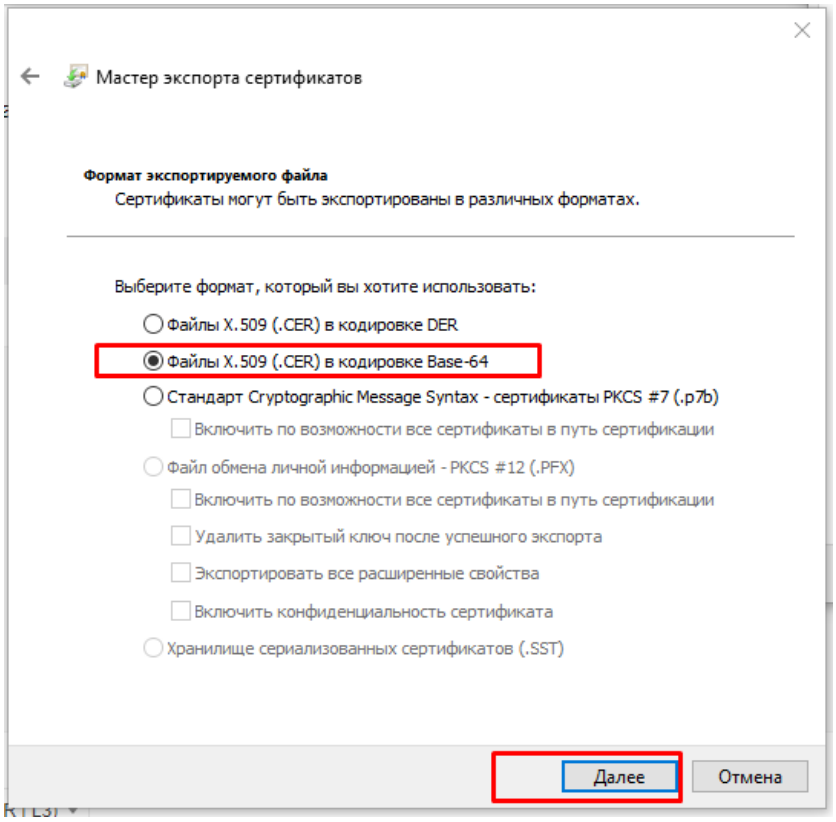
Собрать массив всех передаваемых параметров в виде пар Ключ-Значение:
[{"TerminalKey","TestB"}, {"PaymentId","20150"}]
Добавить в массив пару (Password, значение):
[{"TerminalKey","TestB"}, {"PaymentId","20150"}, {"Password","Dfsfh56dgKl"}]
Отсортировать массив по Ключам по алфавиту:
[{"Password", "Dfsfh56dgKl"},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

12.4 Сохранение сертификата в кодировке Base-64

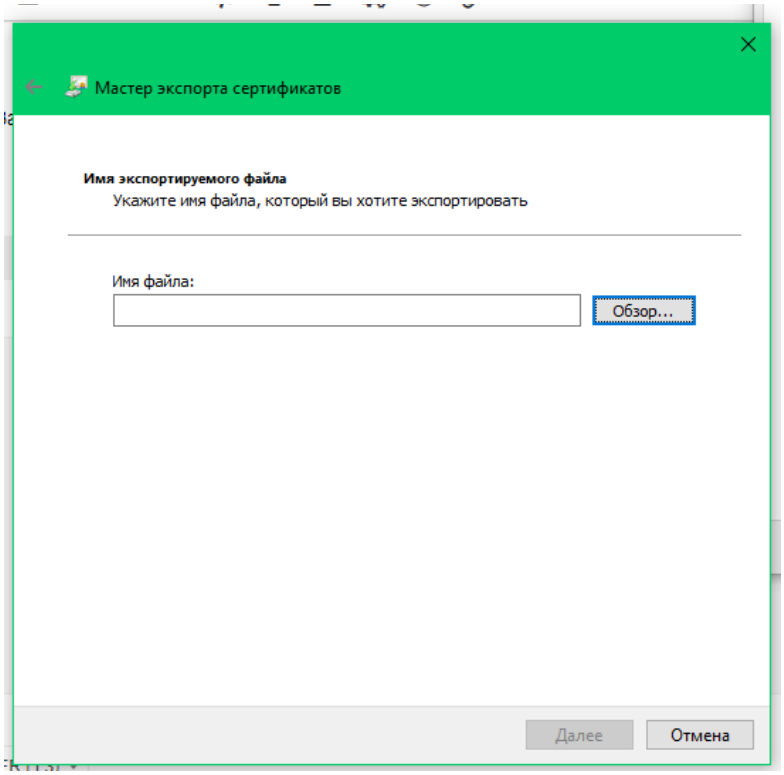
Запускаем сертификат на вкладке «Состав» нажимаем «Копировать в файл»



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



Выбираем путь куда сохранить файл и «далее»



Затем кнопка «Готово».

Результат на скрине