Банк Фридом Финанс

ЭКВАЙРИНГОВАЯ СИСТЕМА

Описание API для клиентов.

Версия 2.5 от 06.05.2021

На 18 страницах

Оглавление

1. Общие сведения.

Документ содержит информацию для выполнения интеграции системы приёма платежей ФФИН Банка (далее ЭС) и сайта или приложения компании клиента. Интеграция включает в себя реализацию взаимодействия через предлагаемый API и встраивание платежной страницы на сайт или в приложение клиента.

Интеграционный интерфейс ЭС представляет из себя набор REST сервисов.

- Формат сообщений JSON.
- Кодировка UTF-8.

При обращении к REST API ЭС - взаимодействие осуществляется по протоколу **HTTPS** с шифрованием трафика и аутентификацией обеих сторон по сертификатам.

Веб страница ввода данных карт плательщиков размещается на стороне ЭС и может быть встроена на сайт или в приложение клиента с помощью iframe. В ЭС доступен ряд стандартных страниц, так же возможна кастомизация платежных страниц по согласованию с партнёром. Получение платежных страниц осуществляется также по протоколу HTTPS.

2. Описание процесса приема платежей.

Плательщик проводит, предусмотренные сайтом или приложением клиента, подготовительные действия для формирования корзины/заказа и т.п. и переходит на страницу оплаты.

Сайт или приложение выполняет запрос в ЭС банка и создаётся заказ. Сайт или приложение получает уникальный идентификатор заказа, сохраняет идентификатор для дальнейшего использования. Далее сайт или приложение формирует ссылку на платежную форму ЭС банка для ввода данных карты и перенаправляет плательщика на эту ссылку или открывает эту ссылку в iframe на своем сайте.

Плательщик вводит данные карты на форме и нажимает кнопку «Оплатить».

Введенные данные, в привязке к зарегистрированному ранее заказу, отправляются на сервер ЭС. ЭС, в случае успеха, отображает страницу “Оплата принята”.

Процесс обработки заказа выполняется в асинхронном режиме.

Платеж выполняется в соответствии с маршрутизацией продукта. В продукте настроены основные параметры сотрудничества согласно договору и приложениям к нему, включая комиссии. Если клиенту доступно несколько продуктов, то ему предоставляется productCode на дополнительный продукт.

3. Аутентификация.

После подключения - клиенту выдается уникальный идентификатор и пароль для доступа к API.

- Идентификатор используется в качестве логина.
- Пароль используется для формирования аутентификационного токена.
- Токен передается в каждом запросе к API и проверяется ЭС.

Токен представляет из себя строку, полученную при помощи sha512 алгоритма, примененного к конкатенации строк подписываемых параметров и пароля API клиента.

Набор и порядок конкатенируемых параметров индивидуален для каждого метода API и описан в соответствующих разделах документации.

Пример:

Необходимо получить подпись для вызова сервиса получения остатков по счету.

Подписываемый параметр: 1

Пароль: test

Сконкатенированная строка: 1test

Получаемый токен:

7791033ef951f8b51a2c21b7a99a0b5260888d5c8301e71926275a65294811024239257665085042a139789a464955326df3c1213c18d88e07d8817e71c073d4

4. Ввод данных платежа.

Страница ввода данных карты размещается на стороне ЭС. Клиент перенаправляет плательщика на форму либо встраивает данную страницу на сайт или в приложение в виде iframe фрагмента. Фрагмент запрашивается HTTP запросом.

Процесс ввода данных карты не контролируется сайтом или приложением клиента и может состоять из нескольких шагов, например, в случае необходимости проведения 3DSecure авторизации операции в банке эмитенте карты.

Для получения статуса заказа - сайт или приложение клиента должно дождаться уведомления от ЭС(если настроен URL для уведомлений), либо запросить статус заказа самостоятельно с помощью соответствующего сервиса API.

Создание заказа в ЭС:

Для создания заказа в ЭС, сайт или приложение должен вызвать сервис создания заказа путем выполнения HTTP запроса с передачей JSON объекта содержащего данные заказа:

POST /acq-company-rest/v2/acq/orders

X-Signature: <sign>

4.1 Параметры запроса на создание заказа

1	partnerId	Идентификатор клиента, передающего запрос в ЭС	Обязательное
2	forMerchantId	Идентификатор продавца (id), в пользу которого принимается платеж клиентом (клиент выступает в роли агента и\или агрегатора)	Для обычных клиентов - Не обязательное. Для агентов - обязательное
3	type	Тип операции. Допустимые значения: PAY_BASKET (оплата корзины) и INCOME_ACCOUNT (пополнение счета)	Обязательное

4	reference	Идентификатор заказа в системе клиента (на стороне ЭС также формируются номера заказов)	Не обязательное
5	clientFio	ФИО плательщика	Не обязательное
6	clientEmail	Email плательщика	Не обязательное
7	amount	Сумма платежа	Обязательное
8	currency	Валюта (RUR)	Обязательное
9	additionalIdentifier	Дополнительный идентификатор 1	Не обязательное
10	additionalIdentifier2	Дополнительный идентификатор 2	Не обязательное
11	additionalIdentifier3	Дополнительный идентификатор 3	Не обязательное
12	additionalIdentifier4	Дополнительный идентификатор 4	Не обязательное
13	additionalIdentifier5	Дополнительный идентификатор 5	Не обязательное
14	backBtnVisible	Показывать или нет кнопку возврата на сайт продавца(true – показывать, false – не показывать)	Не обязательное
15	backBtnSuccessUrl	URL возврата на сайт продавца в случае успешной оплаты	Не обязательное
16	backBtnFailureUrl	URL возврата на сайт продавца в случае неуспешной оплаты	Не обязательное
17	accountNumber	Текущий счет физического лица в ФФИН Банке на который зачисляется платеж (только если для клиента установлена соответствующая настройка)	Не обязательное
18	holdStatus	Резервирование ДС (1-резервирование, любое значение или пустое значение- без резервирования)	Не обязательное
19	productCode	Опциональный код продукта, передаваемый клиентом, если необходима иная от стандартной форма платежной страницы или продукт	Не обязательное
20	positions	Массив товарных позиций (см. таблицу 4.2)	Не обязательное

4.2 Объект товарной позиции

№	Параметр	Описание	Тип	Обязательность
1	quantity	Количество предмета расчета	Дробное	Обязательное

2	price	Цена за единицу предмета расчета с учетом скидок и наценок	Дробное	Обязательное
3	name	Наименование предмета расчета	Строка (1-128)	Обязательное
4	paymentMethodType	Признак способа расчета: 1 – Предоплата 100% 2 – Частичная предоплата 3 – Аванс 4 – Полный расчет 5 – Частичный расчет и кредит 6 – Передача в кредит 7 – оплата кредита	Целое	Обязательное
5	paymentSubjectType	Признак предмета расчета: 1 – Товар 2 – Подакцизный товар 3 – Работа 4 – Услуга 5 – Ставка азартной игры 6 – Выигрыш азартной игры 7 – Лотерейный билет 8 – Выигрыш лотереи 9 – Предоставление РИД 10 – Платеж 11 – Агентское вознаграждение 12 – Составной предмет расчета 13 – Иной предмет расчета 14 – Имущественное право 15 – Внереализационный доход 16 – Страховые взносы 17 - Торговый сбор 18 - Курортный сбор 19 – Залог	Целое	Обязательное
6	nomenclatureCode	Код товарной номенклатуры закодированный в формате base64	Строка	Не обязательное
7	unit	Единица измерения	Строка	Не обязательное

В данном сервисе подпись передается в заголовке запроса (X-Signature). Подпись вычисляется как sha512(<json объект заказа> + <password>).

Если передаются товарные позиции, то поле amount из параметров заказа должно быть равно сумме quantity*price по всем позициям.

Пример:

POST /acq-company-rest/v2/acq/orders

X-Signature:
a08e20e27d94ac8b7e761819aa0d1e54f1fca346a009ce2dad3cb68fa0ac5252aab8e2bb613ec0d4400b
c24836be05e83d2c3f876adbbb4b1b6d5910363cc99b


```
{
  "partnerId":1,
  "reference":"760563327447",
  "type":"INCOME_ACCOUNT",
  "clientFio": "Иванов Иван Иванович",
  "clientEmail": "i.ivanov@bankffin.ru",
  "amount": 5100,
  "currency": "RUR",
  "positions": [
            {
            "quantity": 1,
            "price": 5000,
            "name": "Чайник 2л",
            "paymentMethodType": "4",
            "paymentSubjectType": "1"
            },
            {
            "quantity": 1,
            "price": 100,
            "name": "Чай черный",
            "paymentMethodType": "4",
            "paymentSubjectType": "1"
            }
  ]
}
```


Ответ


```
{
  "id": 1009,
  "partnerId": 1,
  "forMerchantId": null,
  "reference": "760563327447",
  "additionalIdentifier": null,
  "additionalIdentifier2": null,
  "additionalIdentifier3": null,
  "additionalIdentifier4": null,
  "additionalIdentifier5": null,
  "type": "INCOME_ACCOUNT",
  "clientFio": "Иванов Иван Иванович",
  "clientEmail": "i.ivanov@bankffin.ru",
  "amount": "5100.00",
  "currency": "RUR",
  "backBtnVisible": null,
  "backBtnSuccessUrl": null,
  "backBtnFailureUrl": null,
  "accountNumber": null,
  "holdStatus": null,
  "productCode": null,
```

```
  "state": "REGISTERED",
  "date": "06.05.2021 15:07:17",
  "clientIp": null,
  "sign": null,
  "amountUnits": null,
  "positions": [
        {
        "quantity": 1,
        "price": 5000,
        "name": "Чайник 2л",
        "paymentMethodType": 4,
        "paymentSubjectType": 1,
        "nomenclatureCode": null,
        "unit": null
        },
        {
        "quantity": 1,
        "price": 100,
        "name": "Чай черный",
        "paymentMethodType": 4,
        "paymentSubjectType": 1,
        "nomenclatureCode": null,
        "unit": null
        }
  ]
}
```


Показ плательщику формы ввода реквизитов карты:

Далее необходимо перенаправить плательщика или отобразить в iframe форму ввода реквизитов карты по следующей ссылке:

/acq-company-web/payment/?partnerId=<partnerId>&orderId=<orderId>&sign=<sign>

partnerId - идентификатор клиента, передающего запрос в ЭС

orderId - идентификатор заказа в ЭС - поле id из ответа сервиса создания заказов.

sign - подпись. Формируется с использованием в строгом порядке параметров: partnerId, orderId.

Пример:

GET
/acq-company-web/payment/?partnerId=1&orderId=964&sign=38d3a39f8c142bdb6b9643a90563ad095b14565e1516409c7b6ddc624c25d161b3079f75eac93d804e94d76ebcc857bc1d46dadde804e1453a95bd71dd6da6f2

5. Двухэтапная авторизация.

Доступна клиентам, у которых в системных настройках эквайринговой системы включена функция HOLD.

5.1. Первый этап - преавторизация денежных средств.

При проведении данного этапа, денежные средства на счете плательщика блокируются, но не списываются.

В JSON объект при создании заказа добавляется параметр holdStatus.

5.2. Второй этап - списание средств.

Этап списания ранее заблокированных средств со счета плательщика и перечисление их на счет клиента.

POST /acq-company-web/payment/confirmHold

Входящие параметры:

partnerId - идентификатор клиента, передающего запрос в ЭС
orderId - номер заказа в эквайринговой системе
sign – подпись.

Все параметры обязательны. Подпись формируется с использованием в строгом порядке параметров: partnerId, orderId.

Успешный ответ:

```
{
"result": "success"
}
```

В случае ошибки :

```
{
"result": "error"
"message": "Текст ошибки"
}
```

Пример:

POST /acq-company
-web/payment/confirmHold?partnerId=1&orderId=964&sign=38d3a39f8c142bdb6b9643a90563ad095b14565e1516409c7b6ddc624c25d161b3079f75eac93d804e94d76ebcc857bc1d46dadde804e1453a95bd71dd6da6f2

Ответ:

```
{
"result": "success"
}
```

5.3. Отмена преавторизации - возврат средств плательщику.

Отмена преавторизации, посредством которой происходит возврат ранее заблокированной суммы на счете у плательщика.

POST /acq-company-web/payment/releaseHold

Входящие параметры:

partnerId - идентификатор клиента, передающего запрос в ЭС
orderId - номер заказа в эквайринговой системе
sign – подпись.

Все параметры обязательны. Подпись формируется с использованием в строгом порядке параметров: partnerId, orderId.

На выходе получим:

Успешный ответ:

```
{
"result": "success"
}
```

В случае ошибки :

```
{
"result": "error"
"message": "Текст ошибки"
}
```

Пример:

POST /acq-company-web/payment/releaseHold?partnerId=1&orderId=964&sign=38d3a39f8c142bdb6b9643a90563ad095b14565e1516409c7b6ddc624c25d161b3079f75eac93d804e94d76ebcc857bc1d46dadde804e1453a95bd71dd6da6f2

Ответ:

```
{
"result": "success"
}
```

6. Другие сервисы для клиентов.

6.1. Получение списка заказов.

GET /acq-company-rest/acq/orders/?partnerId={1}&orderId={2}&shopOrderId={3}&from={4}&to={5}&status={6}&sign={7}

где

2 - идентификатор заказа

3 - идентификатор заказа клиента

4 - дата "от" (формат dd.MM.yyyy)

5 - дата "до" (формат dd.MM.yyyy)

6 - статус (REGISTERED, NOT_PAID, PAID, REVERSED, CANCELED)

7 - подпись

Если дата "от" и дата "до" не заполнены, то поиск будет произведен за текущий день; иначе - поиск за период дат от и до.

Если статус указан, то будут найдены заказы в определенном статусе; иначе - по всем статусам

Обязательные параметры: 1, 7.

Подписываемые параметры в строгом порядке: 1, 2, 3, 4, 5, 6.

В результате вернется JSON вида:

```
[{
        "id": <id>,
        "partnerId": <partnerId>,
        "forMerchantId": <forMerchantId>,
        "reference": <reference>,
        "additionalIdentifier": <additionalIdentifier>,
        "additionalIdentifier2": <additionalIdentifier2>,
```

```
        "additionalIdentifier3": <additionalIdentifier3>,
        "additionalIdentifier4": <additionalIdentifier4>,
        "additionalIdentifier5": <additionalIdentifier5>,
        "type": <type>,
        "clientFio": <clientFio>,
        "clientEmail": <clientEmail>,
        "amount": <amount>,
        "currency": <currency>,
        "backBtnVisible": <backBtnVisible>,
        "backBtnSuccessUrl": <backBtnSuccessUrl>,
        "backBtnFailureUrl": <backBtnFailureUrl>,
        "accountNumber": <accountNumber>,
        "holdStatus": <holdStatus>,
        "productCode": <productCode>,
        "state": <state>,
        "date": <date>,
        "name": <name>,
        "pan": <pan>,
        "reasonPayment": <reasonPayment>,
        "paymentNumber": <paymentNumber>,
        "bankIssuer": <bankIssuer>,
        "amountClient": <amountClient>,
        "reasonReject": <reasonReject>,
        "bikbankIssuer": <bikbankIssuer>
},...]
```


где:

id - идентификатор заказа
partnerId - идентификатор клиента
forMerchantId - идентификатор клиента в пользу которого принимается платеж
reference - идентификатор заказа в системе клиента
additionalIdentifier - дополнительный идентификатор
additionalIdentifier2 - дополнительный идентификатор 2
additionalIdentifier3 - дополнительный идентификатор 3
additionalIdentifier4 - дополнительный идентификатор 4
additionalIdentifier5 - дополнительный идентификатор 5
type - тип операции
clientFio - ФИО плательщика
clientEmail - email плательщика
amount - сумма
currency - валюта (RUR)
backBtnVisible - показывать или нет кнопку возврата на сайт продавца
backBtnSuccessUrl - URL возврата на сайт продавца в случае успешной оплаты
backBtnFailureUrl - URL возврата на сайт продавца в случае неуспешной оплаты
accountNumber - номер счета на который зачислен платеж
holdStatus - признак двухэтапного списания денег с плательщика
productCode - код продукта, передаваемый клиентом
state - статус заказа
date - дата создания заказа
name - имя на банковской карте плательщика
pan - маскировочный номер на банковской карте плательщика
reasonPayment - основание платежа

paymentNumber - номер платежного поручения
bankIssuer - банк эмитент карты
amountClient - сумма для зачисления плательщику в случае пополнения счета
reasonReject - причина отклонения платежа
bikbankIssuer - БИК банка эмитента

Пример:

GET
/acq-company-rest/acq/orders?partnerId=1&sign=7791033ef951f8b51a2c21b7a99a0b5260888d5c8301e71926275a65294811024239257665085042a139789a464955326df3c1213c18d88e07d8817e71c073d4

Ответ:


```
[{
        "id": 968,
        "partnerId": 1,
        "forMerchantId": null,
        "reference": "329865027436",
        "additionalIdentifier": "",
        "additionalIdentifier2": null,
        "additionalIdentifier3": null,
        "additionalIdentifier4": null,
        "additionalIdentifier5": null,
        "type": "INCOME_ACCOUNT",
        "clientFio": "Иванов Иван Иванович",
        "clientEmail": "ivanov@gmail.com",
        "amount": "300.00",
        "currency": "RUR",
        "backBtnVisible": null,
        "backBtnSuccessUrl": null,
        "backBtnFailureUrl": null,
        "accountNumber": "40701810100015001178",
        "holdStatus": null,
        "productCode": null,
        "state": "PAID",
        "date": "05.04.2021 15:26:38",
        "clientIp": null,
        "sign": null,
        "amountUnits": null,
        "name": "IVAN IVANOV",
        "pan": "520985******1866",
        "reasonPayment": null,
        "paymentNumber": "5638597",
        "bankIssuer": "",
        "amountClient": null,
        "reasonReject": null,
        "bikbankIssuer": ""
},...]
```

7. Получение уведомлений.

По завершению процесса обработки платежа система отправляет уведомление о результате на адрес, указанный клиентом при заключении договора на эквайринг. Данное уведомление содержит подпись и является подтверждением статуса обработки платежа. Если клиент при подключении эквайринга не указал адрес для отправки оповещений, оповещения не отправляются.

Ответ клиента на сообщение должен начинаться с символов "ОК" (латинскими), если уведомление принято удачно, иначе уведомления будут отправляться каждые три минуты. Максимальное количество повторных отправок – 3.

Оповещения отправляются HTTP методом POST в формате JSON, структура сообщения аналогична приведенной в пункте 6.1.

Подпись уведомления передается в заголовке X-Signature HTTP POST запроса. Формирование подписи: sha512(<json уведомления> + <password>).

8. Токены.

Токены эквайринговой системы разработаны для того, чтобы сохранять данные банковских карт (без CVV\CVC) в защищенном хранилище и в дальнейшем при оплатах использовать привязанный к данным карты токен (уникальная в рамках системы последовательность символов) без повторного ввода или передачи данных карт.

8.1. Сохранение токенов.

Страница ввода данных карты размещается на стороне ЭС. Клиент встраивает данную страницу на сайт или в мобильное приложение в виде iframe фрагмента или фрейма (экрана) мобильного приложения. Ссылка на iframe запрашивается HTTP запросом и является уникальной для данного запроса и клиента. Запросить ссылку можно только с заранее установленных IP-адресов клиента.

Процесс ввода данных карты не контролируется сайтом или приложением клиента и может состоять из нескольких шагов.

8.1.1. Запрос страницы ввода данных карты.

Страница ввода данных карты размещается на стороне ЭС. Клиент встраивает данную страницу на сайт в виде iframe фрагмента. Фрагмент запрашивается HTTP запросом с передачей следующих параметров:

POST /acq-company-rest/card_token

Входящие параметры:

partnerID – id Партнёра

referenceNum – id запроса Клиента (чтобы клиент мог связать токен с пользователем и изначальным запросом)

productCode - код доступного продукта клиента (для выбора формы SaveToken)

sign – подпись

Все параметры обязательные.

Порядок параметров для формирования подписи (sign): partnerId + referenceNum + productCode

Ответ в JSON в случае успеха:

```
{
   state: "ok",
   data: {
      url: уникальная для запроса ссылка на форму ввода данных карты SaveToken
```

}
}

Ответ в JSON в случае ошибки:

{
state: "Error",
errorCode: "1",
errorMessage: "<суть ошибки>"
}

8.1.2. Ввод данных карты.

Клиент показывает в своём приложении или на сайте фрейм, где пользователь выбирает сохранять все данные карты, но без CVC/CVV (FULL-токен) или только номер карты (PARTIAL). Вместе с нажатием кнопки “Сохранить” на фрейме в ЭС отправляются данные карты. Пользователю показывается фрейм успеха или ошибки, а в серверную часть приложения или сайта отправляется оповещение.

Оповещения отправляются HTTP методом POST в формате JSON.

Подпись оповещения передается в заголовке X-Signature HTTP POST запроса. Формирование подписи: sha512(<json оповещения> + <password>).

Оповещение:

{
token - токен,
referenceNum - id запроса Клиента (чтобы клиент мог связать токен с пользователем и изначальным запросом),
partnerId - id клиента,
type - тип токена, FULL или PARTIAL,
isVerified - boolean, был ли успешный платёж с использованием токена,
pan - * и последние 4 цифры номера карты,
issuerBank - банк-эмитент,
paySystem - платёжная система, VISA, MC, MIR,
bankLogo - ссылка на логотип банка-эмитента карты,
state - статус токена,
validTo - срок действия токена,
maskedCardNumber - маскированный номер карты (6 первых и 4 последних)
}

8.2. Удаление токенов.

DELETE /acq-company-rest/card_token/<token>?partnerId=<partnerId>&sign=<sign>

Входящие параметры:

token – уникальный код карты (токен)
partnerId – id клиента
sign – подпись

Все параметры обязательные.
Подписываемые параметры для формирования sign в строгом порядке: partnerId, token

Ответ:

{

```
  state: "ok",
  data: {
  данные токена как в п. 8.1.2
  },
  message: "Токен успешно удален"
}
```

В случае ошибки:

```
{
  state: "error",
  errorCode: <код ошибки>,
  errorMessage: <текст ошибки>
}
```

8.3. Расчет комиссии при операциях по токенам.

Эквайринговая система рассчитывает комиссию по операции исходя из комиссии, установленной в продукте для конкретного клиента, для расчета комиссии по переводам карта-карта используется механизм расчета, учитывающий принадлежность карты, **однако банк-эмитент может начислять дополнительную комиссию к таким операциям.**

GET /acq-company-rest/card_token/get_commission_amount?...

Входящие параметры:

partnerId - id клиента
token1 - уникальный код карты (токен) – откуда перевод
token2 - уникальный код карты (токен) – куда перевод
accountNumber - счет в Фридом Финанс банке - куда перевод
productCode - код продукта клиента (необязательный)
amount - сумма
currency - валюта (RUR)
sign – подпись

Все параметры обязательные, кроме token2 и accountNumber – которые обязательны только для операций карта->карта и карта->счёт соответственно.

Порядок полей для расчёта sign:
partnerId + token1 + token2 + productCode + amount + currency
ИЛИ
partnerId + token1 + accountNumber + productCode + amount + currency

Ответ:

```
{
  state: "ok",
  data: {
    commission - комиссия,
    fullAmount - amount+comission,
    currency - валюта (RUR),
    partnerId - id клиента,
    sign - подпись ЭС
  }
}
```

В случае ошибки:

```
{
  state: "error",
  errorCode: <код ошибки>,
  errorMessage: <текст ошибки>
}
```

8.4. Операции с использованием токенов.

API-запрос на осуществления платежа или перевода с использованием токенов (сохранённых данных карт).

POST /acq-company-rest/payment_token/create_operation

Входящие параметры:

partnerId - идентификатор клиента, передающего запрос в ЭС. Обязательное поле.
amount - сумма.
currency - валюта (RUR).
reference - идентификатор заказа в системе клиента (на стороне ЭС так же ведутся № заказов), необязательный**, но необходимый, иначе клиент не сможет связать асинхронный ответ с этим запросом.**

clientFio - ФИО плательщика.
clientEmail - email плательщика.
additionalIdentifier - дополнительный идентификатор 1.
additionalIdentifier2 - дополнительный идентификатор 2.
additionalIdentifier3 - дополнительный идентификатор 3.
additionalIdentifier4 - дополнительный идентификатор 4.
additionalIdentifier5 - дополнительный идентификатор 5.
accountNumber - текущий счет физического лица в ФФИН Банке на который зачисляется платеж (только если для клиента установлена соответствующая настройка).
forMerchantId - идентификатор продавца (id), в пользу которого принимается платеж клиентом (клиент выступает в роли агента и\или агрегатора).
token1 - уникальный код карты (токен) – откуда перевод.
token2 - уникальный код карты (токен) – куда перевод.
productCode - код продукта, передаваемый клиентом для выбора продукта.
sign - подпись.

Обязательные параметры:
А) Для обычных клиентов - partnerId, amount, currency, token1, sign
Б) Для агентов, принимающих платеж в пользу других клиентов - forMerchantId, partnerId, amount, currency, token1, sign

Подписываемые параметры в строгом порядке: partnerId + forMerchantId + reference + additionalIdentifier + additionalIdentifier2 + additionalIdentifier3 + additionalIdentifier4 + additionalIdentifier5 + clientFio + clientEmail + amount + currency + accountNumber + productCode + token1 + token2

В результате вернется JSON с ссылкой на фрейм запроса CVV или ошибка.

```
{
  state: "ok",
```

```
  data: {
    url - уникальная для запроса ссылка на форму запроса CVV/CVC
  }
}
```

После получения CVV и проверки уникальной ссылки отправляется запрос в платежную систему с пришедшими ранее параметрами.

8.5. Запрос статуса токена.

Запрос статуса токена позволяет узнать статус токена и когда истечет срок жизни токена.
GET …/acq-company-rest/card_token/check_status

Входящие параметры:

partnerId – id клиента.
token - уникальный код карты (токен).
referenceNum - id запроса клиента.
sign – подпись.

Ответ:

```
{
  state: "ok",
  data: {
    данные токена как в п. 8.1.2
  }
}
```