aQsi

Программное обеспечение «Эквайринговое ядро ru.aqsi.cardcore»

Руководство эксплуатации (работа с API)

Версия документации: 1.0
Дата сборки: 16.01.2025

Содержание

Введение

Настоящий документ описывает способы взаимодействия программы (application programming interface, API) «Эквайринговое ядро ru.aqsi.cardcore» (далее – JPAY) с другими программами, обеспечивающими работу платежного терминала.

Протокол обмена данными с банковским приложением

Для обмена данными с банковским приложением используется протокол TCP/IP. Клиент подключается по умолчанию к порту 4433.
Последовательность обработки одной команды:

1. Клиент устанавливает TCP соединение с банковским приложением.
2. Передает команду.
3. Получает сообщения от приложения.
4. Отвечает на сообщения.
5. Получает ответ на команду.
6. Закрывает соединение.

Пункты 2 и 3 используются для обеспечения обратной связи во время выполнения некоторых команд. Например, клиент может получить сообщение **keepalive** во время исполнения длительный операции. Клиент может отличить сообщение от ответа на команду по имени тэга.

Формат данных

Клиент и банковское приложение обмениваются пакетами. Пакет может содержать команду, сообщение, ответ на команду или ответ на сообщение. Каждый пакет начинается с 4 двоичных байтов в которых указана длина пакета без учета этих 4х байтов. Первый из байтов длины старший. Далее следует текст в формате XML в кодировке ASCII. Имя корневого тэга команды (или сообщения) это имя команды (или сообщения). Имя корневого тэга ответа должно совпадать с именем указанным в запросе.

Ограничение доступа к функциям программного интерфейса

Функции настройки терминала, отчеты и транзакции с вводом данных карты вручную доступны только администратору терминала.

Настройки

Для перехода в режим настройки приложению передается пароль состоящий из 8 цифр. В ответ клиент получает токен — случайное 16 байтовое число в формате HEX ASCII, которое далее включается в список параметров вызова "защищенных" функций. Пароль по умолчанию 12345678. Клиент должен сохранить полученный при входе в режим администратора токен и использовать его в командах до выхода из режима настройки. Текущий пароль администратора, дополнительно к токену, указывается при смене пароля администратора.

Отчеты и транзакции с вводом данных карты вручную

Сверка итогов, печать отчетов и исполнение транзакций, в которых данные карты вводятся вручную оператором, доступны только по паролю администратора. Схема работы с токеном на них не распространяется.

Команды

В описании команд приводятся примеры значений параметров. В ответе на команду или сообщение всегда содержится тэг с кодом ошибки **status**. Список кодов ошибок приведен в П.1. Тэги в могут включатся в ответ или исключаться из него в зависимости от значения тега **status**. Например в ответе на команду **login** в случае ошибки авторизации отсутствует тэг **token**:

```
<login>
   <status>notauthorized</status>
</login>
```

В ответе на команду может содержаться лог ошибок. Он помещается в тег **error-stack**. Каждое сообщение об ошибке помещается в тег **error**. Для преобразования ошибки в текст следует преобразовать значение тега **error** из формата HEX ASCII в байты и затем полученную двоичную последовательность в строку UTF8:

```
 <loadmasterkeys>
   <status>connectionerror</status>
   <error-stack>
      <error>486F7374206E616D65206973206E6F7420666F756E642E</error>
   </error-stack>
</loadmasterkeys>
```

Сообщение в теге **error** = "Host name is not found".
Клиент, вместо ответа на команду, может получить от приложения сообщение. Имя тэга сообщения отличается от ожидаемого в ответе имени команды. Клиент должен обязательно ответить на сообщение. Если клиент не знает, как обработать сообщение, он должен в ответе на сообщение указать код статуса **notimplemented**. На сообщение **keepalive** следует ответить, указав **status**=ok.

```
<keepalive/>
```

Ответ:

```
<keepalive>
   <status>ok</status>
</keepalive>
```

и снова перейти к ожиданию ответа на команду от приложения. Сообщения может отправлять только приложение. Клиент может только отвечать на поступающие сообщения. Клиент после отправки команды приложению должен дождаться ответа, прежде чем отправлять следующую команду.

Вход в режим администратора

Пример команды:

```
<login>
   <password>12345678</password>
</login>
```

Пример ответа:

```
<login>
   <status>ok</status>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</login>
```

Выход из режима администратора

Пример команды:

```
<logout>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</logout>
```

Пример ответа:

```
<logout>
   <status>ok</status>
</logout>
```

Смена пароля администратора

Пример команды:

```
<changepassword>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
   <password>12345678</password>
   <newpassword>12345678</newpassword>
</changepassword>
```

Пример ответа:

```
<changepassword>
   <status>ok</status>
</changepassword>
```

Проверка и загрузка обновлений конфигурации с сервера конфигураций

Пример команды:

```
<updateconfiguration>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</updateconfiguration>
```

Пример ответа:

```
<updateconfiguration>
   <status>ok</status>
</updateconfiguration>
```

Загрузка рабочих ключей

Пример команды:

```
<loadworkkeys>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</loadworkkeys>
```

Пример ответа:

```
<loadworkkeys>
   <status>ok</status>
   <mac-change-receipt>
   <rrn>123456789123</rrn>
   </mac-change-receipt>
   <net-change-receipt>
   <rrn>023456789120</rrn>
   </net-change-receipt>
</loadworkkeys>
```

В ответе передаются два чека содержащие **rrn** операций загрузки ключа MAC и NET (PEK), если это предусмотрено протоколом сервера авторизации.

Загрузка мастер ключей

Пример команды:

```
<loadmasterkeys>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</loadmasterkeys>
```

Пример ответа:

```
<loadmasterkeys>
   <status>ok</status>
   <receipt>
      <pkcv>123456</pkcv>
      <mkcv>789ABC</mkcv>
      <keys>
         <key>
            <type>10</type>
            <algorithm>01</algorithm>
            <result>success</result>
            <kcv>B5A146</kcv>
         </key>
      </keys>
   </receipt>
</loadmasterkeys>
```

- **pkcv** — KCV мастер ключа PIN
- **mkcv** — KCV мастер ключа MAC
- **keys** — список ключей, в тч те что в полях **pkcv** и **mkcv**
    - **type**
      03 - Мастер ключ для шифрования MAC в Smart Vista
      05 - Мастер ключ для шифрования PIN в Smart Vista
      09 - Мастер ключ для шифрования траффика в Compass+
      0A - Зарезервировано
      0B - Мастерключ для шифрования MAC и PIN в случае если для этих целей используется один ключ
      0C - Ключ KLK1 для первой версии Tieto БРС (обновляемый ключ)
      10 - Ключ KLKM для второй версии Tieto БРС (по видимому это текущая рабочая версия)
      0D - Ключ TMK для протокола ЦФТ (без внешней библиотеки)
      0E - Ключ TAK для протокола ЦФТ (без внешней библиотеки)
      FF - зарезервировано для расширения протокола

Проверка связи с сервером авторизации

Пример команды:

```
<testconnection>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</testconnection>
```

Пример ответа:

```
<testconnection>
```

```
   <status>ok</status>
</testconnection>
```

Сведения о приложении и терминале

Пример команды:

```
<getparameters>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</getparameters>
```

Пример ответа:

```
<getparameters>
   <status>ok</status>
   <parameters>
       <sn>000000000009</sn>
       <app>1.0.67.6</app>
       <firmware-mcu>1.5.3</firmware-mcu>
       <firmware-boot>2.0.1</firmware-boot>
       <os>CS10_V1.07_181127PK</os>
       <sdk>1.0.4</sdk>
       <tid>1000000001</tid>
       <mid>243423434122313</mid>
       <tconf>19-04-01.01</tconf>
       <ntconf>cname</ntconf>
       <cconf>18-10-25.06</cconf>
       <ncconf>cname_test</ncconf>
       <econf>19-04-13.02</econf>
       <neconf>2can_jibe_emv</neconf>
       <kconf>18-08-30.04</kconf>
       <nkconf>combined_ca_database</nkconf>
       <acqid>twocan</acqid>
       <cccert>24.03.2027</cccert>
       <csca>20.11.2037</csca>
       <cshost>192.168.0.185</cshost>
       <accert>24.03.2027</accert>
       <asca>12.10.2020</asca>
       <ashost>192.168.0.2</ashost>
       <kccert>24.03.2027</kccert>
       <ksca>none</ksca>
       <devid>M2100-0000005164</devid>
        </parameters>
</getparameters>
```

- **sn** — серийный номер терминала
- **app** — версия приложения
- **firmware-mcu** — версия защищенного кернела терминала
- **firmware-boot** — версия загрузчика
- **os** — версия операционной системы
- **sdk** — версия SDK (aar)
- **tid** — идентификатор (номер) терминала
- **mid** — идентификатор мерчанта
- **tconf** — версия конфигурации терминала
- **ntconf** — имя конфигурации терминала
- **cconf** — версия общей конфигурации
- **ncconf** — имя общей конфигурации
- **econf** — версия EMV конфигурации
- **neconf** — имя EMV конфигурации
- **kconf** — версия списка ключей платежных систем конфигурации
- **nkconf** — имя списка ключей платежных систем конфигурации

- **acqid** — идентификатор эквайера
- **ccert** — дата окончания действия клиентского сертификата для подключения к серверу конфигурации
- **csca** — дата окончания действия CA сервера конфигурации
- **cshost** — имя хоста или IP сервера конфигурации
- **accert** — дата окончания действия клиентского сертификата для подключения к серверу эквайера
- **asca** — дата окончания действия CA сервера эквайера
- **ashost** — имя текущего хоста или IP сервера эквайера
- **as-main-host** - имя основного хоста эквайера
- **as-mirror-host** - имя зеркала хоста эквайера
- **kccert** — дата окончания действия клиентского сертификата для подключения к серверу загрузки ключей
- **ksca** — дата окончания действия CA сервера загрузки ключей
- **kshost** - имя текущего хоста KLD
- **ks-main-host** - имя основного хоста KLD
- **ks-mirror-host** - имя зеркала хоста KLD
- **devid** — Идентификатор устройства для эквайринга

Транзакция

Поддерживаются следующие типы транзакций:

- **purchase** (оплата)
- **refund** (возврат)
- **void** (отмена)
- **purchase-with-cashback** (оплата с кэшбеком)

Транзакция может быть выполнена без прикладывания карты. Транзакция без прикладывания карты, исполняется в случае, если она поддерживается эквайером и в параметрах транзакции указан PAN карты, или для транзакции **refund** указан параметр **forced**=true, или для транзакции **void** указан параметр **number**=last.

Purchase (оплата):

```
<transaction>
   <type>purchase</type>
   <currency>643</currency>
   <amount>000000012300</amount>
</transaction>
```

Указывается тип транзакции, сумма и код валюты. Сумма должна быть указана 12 цифрами с лидирующими нулями в копейках.

Refund (возврат):

```
<transaction>
   <type>refund</type>
   <amount>000000012300</amount>
   <rrn>120157645346</rrn>
</transaction>
```

Указывается тип операции, сумма и RRN. Сумма должна быть указана 12 цифрами с лидирующими нулями в копейках.

Purchase-with-cashback (Оплата с кешбеком):

```
<transaction>
   <type>purchase-with-cashback</type>
```

```
   <amount>000000012300</amount>
   <amount-other>000000000100</amount-other>
</transaction>
```

Указывается тип операции, сумма и сумма кешбека. Обе суммы должны быть указаны 12 цифрами с лидирующими нулями в копейках.

Void (отмена):

```
<transaction>
   <type>void/type>
   <number>8</number>
   <amount>000000000100</amount>
</transaction>
```

Указывается тип операции, номер чека отменяемой транзакции и сумма, если отмена частичная. Сумма должна быть указана 12 цифрами с лидирующими нулями в копейках. Если сумма не указана, то отмена производится на всю сумму транзакции. Для отмены последней транзакции без прикладывания карты в теге **number** надо указать значение last. Примечание: номер чека отменяемой транзакции **number** возвращается в ответе на эту транзакцию в поле **seq** (см. ниже).

```
<transaction>
   <type>void/type>
   <number>last</number>
</transaction>
```

Варианты команды transaction без прикладывания карты.

А. Если указан PAN карты

```
<transaction>
   <type>purchase</type>
   <amount>000000012300</amount>
   <pan>4000000010000001</pan>
   <expired>1805</expired>
   <password>12345678</password>
   <cardholder>JOHN SMITH</cardholder>
</transaction>
```

Указывается тип транзакции, сумма, PAN, срок действия карты, пароль администратора и имя владельца карты. Этот вариант используется в случае, если все данные карты вводятся вручную. Эквайер может поддерживать такой способ ввода данных карты для всех или некоторых типов транзакций: **purchase**, **refund**, **void**, **purchase-with-cashback**, или вовсе не поддерживать такой способ ввода.

Б. Отмена без прикладывания карты

```
<transaction>
   <type>void/type>
   <number>last</number>
   <amount>000000012300</amount>
</transaction>
```

Если номер отменяемой транзакции имеет значение last отменяется последняя транзакции. Поддерживается всеми эквайерами. Указывать сумму следует в случае частичной отмены. Частичная отмена возможна в случае если она поддерживается эквайером.

В. Возврат и отмена без прикладывания карты

```
<transaction>
   <type>refund</type>
   <amount>000000012300</amount>
   <rrn>120157645346</rrn>
   <forced>true</forced>
</transaction>

<transaction>
   <type>void</type>
   <amount>000000012300</amount>
   <number>100</number>
   <forced>true</forced>
</transaction>
```

Поддерживается некоторыми эквайерами для возврата без прикладывания карты. Следует добавить атрибут **forced** = true. Указывать сумму следует в случае частичного возврата. Частичный возврат возможен в случае если он поддерживается эквайером.

Список параметров команды transaction:

- **type** — тип операции (Приложение 2)
- **amount** — сумма (12 цифр копеек). Для транзакции void сумма может отсутствовать. В этом случае отмена производится на всю сумму отменяемой транзакции
- **amount-other** — сумма кэшбэка (12 цифр копеек) для операции **purchasewithcashback**
- **currency** — код валюты, если он отличается от кода валюты по умолчанию, указанного в конфигурации
- **rrn** — retrieval refernce number (12 символов; необязательный параметр операции refund)
- **number** — номер чека (6 цифр)
- **pan** — номер карты. Указывается при вводе данных карты вручную. Значение last отменяет последнюю транзакцию без прикладывания карты
- **expired** — окончание срока действия карты (4 цифры, ггмм). Указывается при вводе данных карты вручную
- **password** — пароль администратора (8 цифр). Указывается при вводе данных карты вручную
- **cardholder** — имя владельца карты
- **forced** – true/false возврат без прикладывания карты

Пример ответа:

```
<transaction>
   <status>ok</status>
    <receipt>
      <header>
         <line>Merchant Name</line>
         <line>Merchant Address</line>
      </header>
      <tid>1000000001</tid>
      <mid>012345678912345</mid>
      <seq>000009</seq>
      <type>purchase</type>
      <state>active</state>
      <tstatus>approved</tstatus>
      <amount-authorized>000000012300</amount-authorized>
      <amount-other>000000012300</amount-other>
      <currency>643</currency>
      <aid>A00000000031010</aid>
      <ps-code>VISA Classic</ps-code>
      <appname>VISA Classic</appname>
      <pan>************1234</pan>
      <aed>200331</aed>
      <rrn> 120157645346</rrn>
      <resp-code>000</resp-code>
```

```
      <auth-number>AFK045</auth-number>
      <datetime>180328120133</datetime>
      <tsi>EF00</tsi>
      <tvr>2040300000</tvr>
      <decline-reason/>
      <acquirer-tag>abc</acquired-tag>
      <bank-name>demo</bank-name>
      <footer>
         <line>Byte</line>
      </footer>
      <record>A1234DF3...</record>
   </receipt>
   <error-stack/>
</transaction>
```

- **status** — код ошибки (см. Приложение 1)
- **header** — строки заголовка чека
- **tid** — идентификатор (номер) терминала
- **mid** — идентификатор владельца терминала (Merchant ID)
- **seq** — номер чека
- **type** — тип транзакции. Допускаются значения: purchase (оплата), refund (возврат), purchasereversal (отмена оплаты), refundreversal(отмена возврата), reversal (отмена без указания типа отменяемой операции)
- **state** — значение этого поля в чеке всегда active
- **tstatus** — статус транзакции approved или declined
- **amount-authorized** — сумма транзакции в копейках
- **amount-other** — сумма кэшбека в копейках (для транзакции purchasewithcashback)
- **currency** — код валюты
- **aid** — идентификатор приложение EMV карты (AID)
- **ps-code** — уникальный идентификатор платежной системы, через которую проводилась операция, допустимые значения:
  4 — Visa;
  5 — MasterCard;
  6 — UnionPay;
  22 — «Мир».
- **appname** — имя приложения EMV карты рассчитанное по AID
- **apname** – Application Preferred Name (9F12) в формате HEX-ASCII
- **alabel** – Application Label (50)
- **ict-id** – Issuer Code Table Index (9F11)
- **pin-entered** – признак ввода ПИН-кода (online/offline/no)
- **cdcvm** – признак авторизации с помощью устройства (yes/ no)
- **signature** – признак наличия подписи в чеке (yes/no)
- **pan** — последние 4 цифры номера карты
- **aed** — дата окончания срока действия карты (ггммдд)
- **rrn** — retrieval reference number
- **resp-code** — код ответа сервера авторизации
- **resp-code-description** — описание кода ответа сервера авторизации [JCONF 6.11]
- **auth-number** — код авторизации
- **datetime** — дата и время проведения транзакции (yyMMddhhmmss) в часовом поясе кассы
- **tsi** — значение EMV тэга Transaction Status Information
- **tvr** — значение EMV тэга Terminal Verification Result
- **footer** — строки внизу чека
- **decline-reason** — значение этого тега используется только в случае если транзакция отклонена. Возможные значения:
  - **unabletogoonline**
  - **offlinedeclined**

    - **systemerror**
    - **onlinedeclined**
    - **cardprocessingerror**
- **acquirer-tag** — значение, использующееся для привязки к платежной системе. Копируется в чек из конфигурации [JCONF]
- **bank-name** – имя банка. Копируется в чек из конфигурации [JCONF]

Для получения имени приложения следует использовать теги **apname**, **alabel** и **appname** в указанном порядке. Для декодирования значения тега **apname** используется индекс кодовой таблицы **ict-id**. Если теги **apname** или **ict-id** отсутствуют или указанная кодировка не поддерживается терминалом, имя приложения извлекается из **alabel**. Если **alabel** тоже отсутствует, используется **appname**.
Транзакция успешно выполнена и одобрена, если в ответе имеется чек и тэг чека **tstatus** = approved. Чек в ответе может отсутствовать если значение тэга **status** отличается от ok.
Во время исполнения транзакции терминал может передавать сообщения **keepalive**, **display**, **dex** и **smartcard**.

Сообщение keepalive

Если в ответ на сообщение **keepalive** терминалу возвращается **status**=cancelled терминал прерывает обработку транзакции.

Сообщение display

Сообщение **display** содержит информацию для отображения на экране. Код сообщения передается в теге **code**. Значение тега **code** указано в таблицах [EMVA, Table 9-5] и [EMV4, Table 8]. Дополнительно к указанным в этих таблицах используются следующие коды:

- **D1** — подключение к хосту банка
- **D2** — Повторное подключение
- **D3** — Нет ответа от хоста банка
- **D4** — Ответ получен
- **D5** — Превышен счетчик попыток ввода ПИН-кода

```
<display>
   <code>03</code>
   <language>ruen</language>
   <msg>ОДОБРЕНО</msg>
   <status>02</status>
   <hold-time>000000</hold-time>
   <value-qualifier>00</value-qualifier>
   <value>000000000000</value>
   <currency>643</currency>
</display>
```

В теге **lang** передается список предпочтительных языков, для отображения сообщения. Каждый язык представлен парой символов в формате ISO-639. Описание параметров сообщения приведено в [EMVC2, A.1.194]. Если v**alue-qualifier** = 10 (AMOUNT), то **value** содержит сумму транзакции для отображения на экране, а **currency** — код валюты транзакции.

Сообщение dex

Сообщение **dex** приходит от терминала во время исполнения транзакции, если это разрешено в общих настройках терминала [JCONF] **data-exchange** = enabled. Кернел отправляет терминалу теги, указанные в EMV настройках [L2EMV]. Для указания списка тегов в конфигурации MCL используется тег DF8112 (Tags to Read), в остальных платежных системах тег DF811E (Data Exchange Tag List).

```
<dex>
        <kernel-tags>
                <tag name="5A" value="01234567890ABCDEF" />
```

```
   </kernel-tags>
</dex>
```

Терминал должен вернуть в ответ **status** = ok и, если требуется, теги, которые кернел изменит/добавит в свои данные перед тем как продолжить исполнение транзакции. Формат значений этих тегов должен соответствовать спецификации EMV.

```
<dex>
   <status>ok</status>
   <terminal-tags>
               <tag name="9F06" value="000000001000" />
   </terminal-tags>
</dex>
```

Сообщение smartcard для интеграции обработки транспортных карт

Сообщение **smartcard** предназначено для интеграции приложений, использующих карты MIFARE, с JPAY. Сообщение **smartcard** приходит от терминала во время исполнения транзакции при обнаружении карты, в случае если работа с картами MIFARE разрешена в настройках приложения [JCONF] .

```
<interfaces><mifare enable=”true”/>...</interfaces>
```

Сообщение приходит при обнаружении любой карты, в том числе банковской. Сообщение содержит код команды в теге **command**. В настоящее время со стороны терминала поддерживается только одна команда **newcard**, в которой клиенту передается дескриптор карты. Варианты ответа на эту команду представлены ниже. В ответ на неизвестные команды терминала, клиент должен отвечать командой **next**. Пример команды приведен ниже.

```
<smartcard>
        <command>newcard</command>
        <descriptor>0705804E4B097000440028</descriptor>
</smartcard>
```

Дескриптор карты имеет следующий формат. Данные представлены в формате HEX ASCII.

1 байт	N байтов	2 байта	1 байт
Длина UID (N)	UID	ATQA	SAK

Клиент анализируя параметры сообщения принимает решение о дальнейшей обработке карты. Клиент отвечает командой **next**, если принято решение продолжить обрабатывать карту, как банковскую.

```
<smartcard>
   <command>next</command>
</smartcard>
```

Клиент отвечает командой **end**, если принято решение завершить обработку карты. Этот вариант предполагает самостоятельную обработку карты MIFARE клиентом до возврата ответа. Команда содержит параметр **indicator** значение которого позволяет управлять звуковым сигналом и светодиодами терминала. Если значение параметра **ok**, индикаторы терминала отображают успешный статус транзакции, если **error** - ошибку. Если параметр отсутствует или имеет другое значение индикаторы не используются.

```
<smartcard>
   <command>end</command>
   <indicator>ok</indicator>
</smartcard>
```

Клиент отвечает командой **retry**, без параметров, если принято решение детектировать карту повторно.

```
<smartcard>
   <command>retry</command>
</smartcard>
```

Система быстрых платежей

Оплата и возврат через СБП производится с помощью команды **transaction** (8.9). Оплата СБП настраивается и разрешается в конфигурации. Для типа операции **payment**, в случае если оплата СБП разрешена, запрос QR кода делается одновременно с началом детектирования карты. Если QR код получен от агента СБП, он передается клиенту в сообщении:

```
<displayqrcode>
   <status>show</status>
   <url>https://example.com/qrcode/jAhh76JhjK</url>
</displayqrcode>
```

Клиент должен сформировать QR код и вывести его на экран. В случае, если получен ответ об оплате или отклонении оплаты от агента СБП, клиенту отправляется сообщение:

```
<displayqrcode>
   <status>hide</status>
</displayqrcode>
```

Получив такое сообщение, клиент должен прекратить отображение QR кода на экране. На сообщение **displayqrcode** следует отвечать, как и на другие сообщения возвращая в поле статуса ok. Оплата через СБП ожидается до истечения таймаута детектирования карты, или до обнаружения карты в одном из разрешенных интерфейсов. Если подтверждение оплаты СБП не получено до истечения времени таймаута, транзакция завершается. Если подтверждение оплаты СБП не получено до обнаружения карты, производится оплата по карте. Если во время детектирования карты получено подтверждение оплаты или отклонения оплаты через СБП, детектирование карты прекращается и возвращается результат транзакции. Если, при чтении статуса СБП транзакции возвращается ошибка, клиенту отправляется сообщение **displayqrcode** со статусом hide и детектирование карты продолжается до истечения времени таймаута.
При оплате через СБП можно отключить другие интерфейсы детектирования карты. В этом случае будет разрешена только оплата через СБП. Для этого в команде transaction следует передать атрибут **force-sbp** с значением true:

```
<transaction type="payment" force-sbp="true" ...>
```

Если указать значение false, то оплата через СБП не будет предлагаться. Если атрибут не указан, то будут предлагаться все способы оплаты разрешенные в конфигурации терминала.
Для того, чтобы выполнить возврат оплаты через СБП следует обязательно указать атрибут **force-sbp**. Иначе возврат будет производиться по карте.

```
<transaction type="refund" force-sbp="true" ...>
```

Пример ответа на оплату/возврат через СБП показан ниже. Обычно агент СБП имеет собственные значения **tid** и **mid**. Номер транзакции в поле **seq** назначается независимо от номеров карточных транзакций. Форматы тегов **resp-code**, **auth-number** и **rrn** определяются агентом СБП и могу отличаться от аналогичных параметров карточных транзакций. Чек оплаты/возврата СБП содержит тег **iface** с значением sbp.

```
<transaction>
        <status>ok</status>
    <receipt>
```

```
        <header>
                        <line>Merchant Name</line>
                        <line>Merchant Address</line>
                </header>
                <tid>1000000001</tid>
                <mid>012345678912345</mid>
                <seq>000009</seq>
                <type>purchase</type>
                <status>approved</status>
                <amount-authorized>000000012300</amount-authorized>
                <currency>643</currency>
                <rrn>120157645346</rrn>
                <resp-code>000</resp-code>
                <auth-number>AFK045</auth-number>
                <datetime>180328120133</datetime>
                <iface>sbp</iface>
                <tid-sbp>123456789A2</tid-sbp>
                <bank-name>БАНК</bank-name>
        </receipt>
        <error-stack/>
</transaction>
```

В теге <tid> передается номер терминала, указанный в параметре конфигурации <terminal-id>. В теге <tid-sbp> передается номер терминала, указанный в параметре конфигурации <sbp><terminal-id> В теге <bank-name> передается значение параметра конфигурации <sbp><bank-name>.
Чек СБП ЦФТ содержит дополнительную информацию. При завершении транзакции через СБП-агента ЦФТ в чек (тег **receipt**) внутри тега **cft** выводится дополнительная информация, полученная от сервиса MPI. Подробное описание и назначение полей описано в документации к MPI-сервису ЦФТ.
Пример чека при выполнении транзакции через СБП-агента ЦФТ:

```
<receipt>
    <header />
    <mid>MA0000086770</mid>
    <seq>53</seq>
    <type>purchase</type>
    <status>approved</status>
    <tstatus>approved</tstatus>
    <amount-authorized>000000000110</amount-authorized>
    <currency>643</currency>
    <rrn>232796</rrn>
    <auth-number></auth-number>
    <resp-code></resp-code>
    <datetime>220822153936</datetime>
    <iface>sbp</iface>
    <tid-sbp>123456789A2</tid-sbp>
    <bank-name>БАНК</bank-name>
    <cft>
                <localQrcId>232796</localQrcId>
                <pointId>7f7f65af-7de3-44ec-8c89-8e97c52c6f0e</pointId>
        <mpiOperationStatusDescriptionEng>Payment accepted
        </mpiOperationStatusDescriptionEng>
        <mpiOperationStatusDescriptionRus>Платеж завершен успешно
        </mpiOperationStatusDescriptionRus>
        <qrcId>AD100031O24RJFCM9NKREDRU8IUPC1N0</qrcId>
        <extEntityId>1F8BOHBWEMC</extEntityId>
        <paymentPurpose>Оплата товара или услуги</paymentPurpose>
    </cft>
    <error-code>ok</error-code>
    <footer />
  </receipt>
```

Для выполнения возврата через СБП-агента ЦФТ, используется поле **localQrcId** (Номер ссылки оплаты) сервиса MPI, которое дублируется внутри поля **rrn** jpay.

Отчет

Пример команды:

```
<report>
   <password>12345678</password>
   <report-type>brief</report-type>
</report>
```

- **password** — пароль (8 цифр)
- **report-type** — вид отчета (**brief** — краткий, **full** — полный)

Пример ответа:

```
<report>
   <status>ok</status>
    <xreport>
      <header>
         <mid>M123456789012345</mid>
         <tid>1000000001</tid>
         <title>
             <line1>Header line 1</line>
         </title>
         <cur>643</cur>
         <time>189329120002</time>
      </header>
      <transactions>
         <t>
            <status>approved</status>
            <state>active</state>
            <type>refund</type>
            <seq>000009</seq>
            <aa>000000001000</aa>
            <ao>000000000100</ao>
            <rrn>647394847384</rrn>
            <time>180322121408</time>
            <apn>AFJ879</apn>
            <arc>000</arc>
            <cur>643</cur>
            <pan>************1234</pan>
            <aed>211231</aed>
            <aid>A0000000031010</aid>
            <tvr>0000000000</tvr>
            <tsi>000000</tsi>
         </t>
         ...
      </transactions>
      <totals>
         <card-type>
            <rid>A000000003</rid>
            <name>VISA</name>
            <purchase-count>100</purchase-count>
            <purchase-sum>000000010000</purchase-sum>
            <refund-count>100</refund-count>
            <refund-sum>000000010000</refund-sum>
            <purchase-with-cashback-count>
            100
            </purchase-with-cashback-count>
            <purchase-with-cashback-sum>
            000000010000
            </purchase-with-cashback-sum>
            <purchase-reversal-count>
            100
            </purchase-reversal-count>
```

```
            <purchase-reversal-sum>
            000000010000
            </purchase-reversal-sum>
            <refund-reversal-count>2</refund-reversal-count>
            <refund-reversal-sum>
            000000010000
            </refund-reversal-sum>
            <total-card-sum>C000000010000</total-card-sum>
            <total-card-count>400</total-card-count>
         </card-type>
      ...
      </totals>
      <grand-totals>
         <purchase-total-count>100</purchase-total-count>
         <purchase-total-sum>000000010000</purchase-total-sum>
         <refund-total-count>100</refund-total-count>
         <refund-total-sum>000000010000</refund-total-sum>
         <purchase-with-cashback-total-count>
         100
         </purchase-with-cashback-total-count>
         <purchase-with-cashback-stotal-sum>
         000000010000
         </purchase-with-cashback-total-sum>
         <purchase-reversal-total-count>
         100
         </purchase-reversal-total-count>
         <purchase-reversal-total-sum>
         000000010000
         </purchase-reversal-total-sum>
         <refund-reversal-total-count>
         2
         </refund-reversal-total-count>
         <refund-reversal-total-sum>
         000000010000
         </refund-total-reversal-sum>
         <total-sum>C000000010000</total-sum>
         <total-count>400</total-count>
      </grand-totals>
   </xreport>
</report>
```

- **status** — код ошибки (см. Приложение 1)
- **tid** — идентификатор (номер) терминала
- **mid** — идентификатор владельца терминала (Merchant ID)
- **title** — строки заголовка чека
- **cur** — код валюты
- **time** — дата и время составления отчета (yymmddHHMMSS)
- **transactions** — список транзакций в полном (full) отчете. Для каждой транзакции указывается:
    - **status** — код ошибки (см. Приложение 1)
    - **state** — значение этого поля в чеке всегда **active**
    - **type** — тип транзакции
    - **seq** — номер чека
    - **aa** — сумма транзакции в копейках
    - **ao** — сумма кэшбека в копейках (для транзакции **purchasewithcashback**)
    - **rrn** — retrieval reference number
    - **time** — дата и время проведения транзакции (yymmddHHMMSS)
    - **apn** — код авторизации
    - **arc** — код ответа сервера авторизации
    - **cur** — код валюты

    - **pan** — последние 4 цифры номера карты
    - **aed** — дата окончания срока действия карты (ггммдд)
    - **aid** — идентификатор приложение EMV карты (AID)
    - **tvr** — значение тэга terminal verification result
    - **tsi** — значение тэга transaction status information
- **totals** — итоговые данные отчета сгруппированные по типу карты. Для каждого типа карты указывается:
    - **rid** — registered application provider identifier (RID) карты
    - **name** — название карты
    - **purchase-count** — количество операций оплата
    - **purchase-sum** — общая сумма всех операций оплата
    - **refund-count** — количество операций возврат
    - **refund-sum** — общая сумма всех операций возврат
    - **purchase-with-cashback-count** — количество операций оплата с кэшбэком
    - **purchase-with-cashback-sum** — общая сумма всех операций оплата с кэшбэком
    - **purchase-reversal-count** — количество операций отмены оплаты
    - **purchase-reversal-sum** — общая сумма всех операций отмены оплаты
    - **refund-reversal-count** — количество операций отмены возврата
    - **refund-reversal-sum** — общая сумма всех операций отмены возврата
    - **total-card-sum** — баланс всех операций по карте. Баланс это 12 цифр со знаком. Знак плюс обозначается символом 'D'. Знак минус символом 'C'
    - **total-card-count** — общее количество операций по карте
- **grand-totals** — итоговые данные отчета:
    - **purchase-total-count** — количество операций оплата
    - **purchase-total-sum** — общая сумма всех операций оплата
    - **refund-total-count** — количество операций возврат
    - **refund-total-sum** — общая сумма всех операций возврат
    - **purchase-with-cashback-total-count** — количество операций оплата с кэшбэком
    - **purchase-with-cashback-total-sum** — общая сумма всех операций оплата с кэшбэком
    - **purchase-reversal-total-count** — количество операций отмены оплаты
    - **purchase-reversal-total-sum** — общая сумма всех операций отмены оплаты
    - **refund-reversal-total-count** — количество операций отмены возврата
    - **refund-reversal-total-sum** — общая сумма всех операций отмены возврата
    - **total-sum** — баланс всех операций. Баланс это 12 цифр со знаком. Знак плюс обозначается символом 'D'. Знак минус символом 'C'
    - **total-count** — общее количество операций

Сверка итогов

Пример команды:

```
<settlement>
   <password>12345678</password>
   <force>on</force>
</settlement>
```

- **force** (опционально) — вызывает принудительную сверку итогов. На данный момент используется только с эквайером "**cft**" и только в случае, если основная сверка закончилась неуспешно с ошибкой 13.

Пример ответа:

```
<settlement>
   <status>ok</status>
   <bank-name>BANK</bank-name>
```

```
   <sreport>
      <resp-code>000</resp-code>
      <approval-number>ASD002</approval-number>
      <rrn>837495759322</rrn>
      <orig-amount>D003030001000</orig-amount>
      <amount>D003030001000</amount>
      <datetime>180322102354</datetime>
      <tid>1000000001</tid>
      <mid>123456789012345</mid>
      <cur>643</cur>
      <from>190101100000</from>
      <to>190102110000</to>
      <tnum>100</tnum>
      <batch-upload>passed</batch-upload>
      <art-resp-code>00</art-resp-code>
      <cov-resp-code>00</cov-resp-code>
      <cov-rrn>123456789012</cov-rrn>
      <settlement-result>passed</settlement-result>
   </sreport>
</settlement>
```

- **status** — код ошибки
- **bank-name** – имя банка. Копируется из конфигурации [JCONF]
- **resp-code** — код ответа сервера авторизации
- **pproval-number** — код авторизации
- **rrn** — retrieval reference number
- **orig-amount** — итоговая сумма подсчитанная на терминале
- **amount** — итоговая сумма переданная сервером авторизации
- **datetime** — дата и время проведения транзакции (yymmddHHMMSS)
- **tid** — идентификатор (номер) терминала
- **mid** — идентификатор владельца терминала (Merchant ID)
- **cur** — код валюты
- **from** — дата и время первой транзакции в сверке
- **to** — дата и время последней транзакции в сверке
- **tnum** — количество транзакций в сверке
- **batch-upload** — в случае если суммы при сверке не совпали требуется загрузка транзакций. В этом теге указывается результат такой загрузки. **passed** — загрузка выполнена успешно, **failed** — загрузка транзакций не удалась. Этот тег может отсутствовать
- **art-resp-code** — Код ответа на запрос завершающий загрузку транзакций. Этот тег может отсутствовать
- **cov-resp-code** — код возврата операции очистки журнала. Тег включается в ответ, если после сверки итогов выполняется операция очистки журнала (**cutover**). Для этого в конфигурации надо указать настройку **settlement cutover**="true" [4]
- **cov-rrn** — RRN операции очистки журнала. Присутствует в случае если операция очистки журнала успешно выполнена
- **settlement-result** — результат операции сверки итогов. **passed** — сверка завершена успешно. **failed** — операция не выполнена

После успешной сверки итогов из лога транзакций удаляются все записи.

Очистка журнала

Эта операция удаляет все записи из локального журнала транзакций и выполняет операцию **Cutover**. Пример команды:

```
<clear>
   <password>12345678</password>
</clear>
```

Пример ответа:

```
<clear>
   <status>ok</status>
   <sreport>
      <resp-code>000</resp-code>
      <approval-number>ASD002</approval-number>
      <rrn>837495759322</rrn>
      <orig-amount>D003030001000</orig-amount>
      <amount>D003030001000</amount>
      <datetime>180322102354</datetime>
      <tid>1000000001</tid>
      <mid>123456789012345</mid>
   </sreport>
</clear>
```

- **status** — код ошибки
- **resp-code** — код ответа сервера авторизации
- **approval-number** — код авторизации
- **rrn** — retrieval reference number
- **orig-amount** — итоговая сумма подсчитанная на терминале
- **amount** — итоговая сумма переданная сервером авторизации
- **datetime** — дата и время проведения транзакции (yymmddHHMMSS)
- **tid** — идентификатор (номер) терминала
- **mid** — идентификатор владельца терминала (Merchant ID)

Сброс пароля

Пример команды:

```
<resetpassword/>
```

Пример ответа:

```
<resetpassword>
   <status>ok</status>
</resetpassword>
```

Установка ключа хоста для SAM AV2

Команда устанавливает хост ключ для авторизации карт MIFARE через SAM. Команда требует предварительной авторизации. Параметры ключа должны быть указаны в теге **host-key** в конфигурации. Значение тега **name** используется для поиска ключа в конфигурации. Команда не проверяет наличие ключа с указанным именем в конфигурации. В теге **material** содержится значение ключа в открытом виде в формате Base64. Следует использовать шифрацию команд для передачи ключа. Значение ключа хоста также может быть непосредственно указано в конфигурации или получено с сервера загрузки ключей по команде **loadmasterkeys**. Рекомендуется выбрать один из способов загрузки этого ключа, так как каждый из перечисленных способов при исполнении перезаписывает ключ.

Пример команды:

```
<sethostkey>
        <token>DE9773A8CB888560AB0F89C07623FE03</token>
   <name>hkey1</name>
   <material>Abc8dEfg...</material>
</sethostkey>
Пример ответа:
```

```
<sethostkey>
   <status>ok</status>
</sethostkey>
```

Шифрование команд

Данные, передаваемые в командах, сообщениях и ответах на команды можно защитить шифрованием. Для установки защищенного соединения используется протокол ECDHE. Клиент устанавливает соединение используя расширение команды **login**. Клиент передает в команде атрибут **client-handshake**, который содержит публичный ключ клиента в формате base64. Поддерживается эллиптическая кривая SECP256R1. Размер ключа клиента для этого алгоритма равен 69 байтов. JPAY в ответ возвращает публичный ключ сервера в атрибуте **server-handshake** в корневом теге ответа на команду **login**, так же в формате base64. Клиент использует полученный ключ для вычисления общего секрета длиной 32 байта. Этот секрет используется как ключ AES256 для шифрования данных, передаваемых между сервером (JPAY) и клиентом. Ключ следует безопасно сохранить для шифрования и дешифрования данных. Зашифрованные команды, сообщения и ответы сохраняют свои корневые теги, описание которых приводится в разделе 8, но все вложенные теги заменяются на тег **encrypted**, в который помещается зашифрованный блок тегов в формате base64. Блог тегов формируется следующим образом:

1. Создается xml тег **encrypted-data**
2. В тег **encrypted-data** переносятся все вложенные теги команды, сообщения или ответа на команду
3. Тег полностью преобразуется в строку в формате utf-8 и далее в последовательность байтов
4. Последовательность байтов шифруется общим ключом AES256 полученным ранее с помощью протокола ECDHE в режиме GCM

Режим GCM позволяет аутентифицировать передаваемые данные. Зашифрованная последовательность должна включать, в указанном порядке: тег аутентификации длиной 16 байтов, начальное значение (iv) длиной 16 байтов, которое следует назначать согласно рекомендациям, и далее зашифрованные данные.
Зашифрованная последовательность преобразуется в формат base64 и подставляется в тег **encrypted**.
Расшифровка ответов и сообщений полученных от JPAY производится в обратном порядке.
Если JPAY получает зашифрованную команду с тегом **encrypted**, то в случае, если была выполнена инициализация защищенного соединения, ответ также шифруется. Если инициализация не была выполнена, клиенту возвращается ошибка. Если команда использует сообщения, для зашифрованной команды они так же передаются зашифрованными. Если клиент передает незашифрованную команду, то ответ и сообщения возвращаются ему не зашифрованными, даже если инициализация защищенного соединения была выполнена. Таким образом защитой соединения всегда управляет клиент.
Пример команды **login**:

```
<login client-handshake="AwAXQQRzAM5eZjNMEPceu8dkZo4FoVZH+...">
        <password>12345678</password>
</login>
```

Пример ответа на команду **login**:

```
<login server-handshake="QQRi6nNZ+O9c1KWxjxNbQ7RYr/... ">
        <encrypted>T2W4uRXp26CL/aRIb...</encrypted>
</login>
```

В ответе передается зашифрованный общим ключом и преобразованный в строку base64 XML:

```
<encrypted-data>
        <status>ok</status>
        <token>392480923040932923</token>
```

```
<encrypted-data>
```

Обработка ответа на команду **login** отличается от обработки других команд тем, что сначала надо вычислить и сохранить общий ключ, а затем расшифровать им значение тега **encrypted**.
PAN карты возвращается в ответе на команду **transaction** в теге **plain-text-pan**, только если используется защищенное соединение. Клиент должен обеспечить безопасную обработку этого параметра.
См. также [RFC4492].

Инициализация терминала

Для запуска процедуры инициализации терминала используется команда **runreset**:

```
<runreset>
   <token>DE9773A8CB888560AB0F89C07623FE03</token>
</runreset>
```

Приложение ожидает подключение программы активатора на порт 4434 и команду reset от активатора. Приложение периодически отправляет по каналу команды **runreset** сообщение:

```
<keepalive>
   <reset-connected-indicator>0</reset-connected-indicator>
</keepalive>
```

Значение тега **reset-connected-indicator** устанавливается в 1 когда к терминалу подключается активатор. Если в ответ на сообщение **keepalive** получен код ошибки, отличный от ok процедура инициализации прерывается.
Команда reset, в отличие от других команд, обрабатывается только в случае, если она передается через порт 4434.

```
<reset/>
```

В ответ на команду reset приложение:
1 . Сбрасывает свои настройки
2 . Генерирует новый мастер ключ
3 . Генерирует RSA ключ клиентского сертификата для сервера конфигураций
4 . Оправляет сообщение

```
<signconfigcientcertificate>
  <key>1234ABCD...</key>
  <device-identifier>IMS0000000009</device-identifier>
</signconfigcientcertificate>
```

- **key** — Публичный RSA ключ для создания сертификата устройства
- **device-identifier** — Символьный идентификатор устройства, включающий его серийный номер

5 . В ответ на это сообщение активатор возвращает:

```
<signconfigclientcertificate>
  <signed-cert>
     <cert>AB12C4ED ...</cert>
     <ca>56AE54C6 ...</ca>
  </signed-cert>
</signconfigclientcertificate>
```

- **cert** — подписанный сертификат терминала для доступа к серверу конфигураций. Сертификаты передаются в формате PEM. Двоичные образы PEM преобразуются в HEX ACSII и помещаются в тэги cert и ca

- **ca** — корневой сертификат, которым подписан cert

6 . Терминал генерирует RSA ключ клиентского сертификата для загрузчика ключей и запрашивает у активатора этот сертификат, отправляя ему сообщение:

```
<signkeyloaderclientcertificate>
      ...
</signkeyloaderclientcertificate>
```

Формат этого сообщения и формат ответа такие же как для сообщения **signconfigclientcertificate**. Ответ содержит сертификат клиента загрузчика ключей и корневой сертификат, которым он подписан.

7 . Терминал генерирует RSA ключ клиентского сертификата для сервера авторизации и запрашивает у активатора этот сертификат, отправляя ему сообщение:

```
<signacquirerclientcertificate>
     ...
</signacquirerclientcertificate>
```

Формат этого сообщения и формат ответа такие же как для сообщения **signconfigclientcertificate**. Ответ содержит сертификат клиента для сервера авторизации и корневой сертификат, которым он подписан. Таким образом, для каждого из трех серверов создается отдельный клиентский сертификат.

8 . Терминал отправляет активатору запрос параметров сервера конфигурации:

```
<queryconfigcredentials/>
```

9 . Активатор возвращает:

```
<queryconfigcredentials>
  <confdata>
     <cert>AB1234...</cert>
     <sign>CDEF5678...<sign>
     <url>https://myconfig.server</url>
     </confdata>
</queryconfigcredentials>
```

- **cert** — корневой сертификат сервера конфигураций
- **sign** — цифровая подпись сертификата cert. Эта подпись поверяется публичным RSA ключом, встроенным в код приложения
- **url** — URL сервера конфигураций

10 . На этом инициализация заканчивается. Приложение закрывает соединение с активатором и возвращает код ошибки в ответ на команду **runreset**:

```
<runreset>
  <status>ok</status>
</runreset>
```

Приложение А. Коды ошибок

- **ok** — операция завершена успешно
- **failed** — операция завершена с ошибкой
- **communicationerror** — ошибка связи в т.ч. ошибка установки соединения по сети
- **notimplemented** — запрашиваемая функция не реализована
- **ormaterror** — ошибка представления данных
- **notauthorized** — неопустимый логин или пароль
- **fileioerror** — ошибка доступа к файлу
- **verificationfailed** — ошибка проверки контрольной суммы или сертификата
- **missingdata** — отсутствуют данные необходимые для выполнения операции
- **systemerror** — системная ошибка
- **timeout** — превышено время ожидания завершения операции
- **invalidarguments** — указаны недопустимые значения параметров операции
- **transactionnotfound** — отменяемая транзакция отсутствует в логе
- **notreversible** — транзакция не может быть отменена
- **alreadyreversed** — транзакция уже отменена
- **notapproved** — отменяемая транзакция не одобрена
- **emverror** — ошибка ядра EMV. В т.ч. ошибка инициализации
- **notdetected** — ошибка обнаружения карты
- **notallowedinterface** — использование интерфейса запрещено для данной операции
- **cancelled** — операция отменена
- **tryagain** — ошибка чтения карты; повторить
- **usechip** — требуется провести контактную emv транзакцию
- **batchuploadfailed** — ошибка загрузки лога транзакций
- **readerdisabled** — устройство чтения карт отключено
- **preprocessingfailed** — ошибка предварительной обработки транзакции в ядре EMV
- **readernotavailable** — устройство чтения карт отсутствует
- **readererror** — ошибка устройства чтения карт
- **tryanotherinterface** — ошибка чтения карты; используйте другой интерфейс
- **pinpadmalfunctionornotprsent** — ошибка ввода ПИН-кода
- **pinentrybypassed** — ввод ПИН-кода отменен пользователем

Приложение Б. Типы транзакций

- **purchase** — оплата
- **refund** — возврат
- **purchase-with-cashback** — оплата с кэшбеком
- **void** – отмена

Приложение B. Лог

JPAY сохраняет сообщения в системном логе или выводит их на терминал. Второй способ используется для отладки и не работает, если JPAY запущен как сервис. Для того чтобы сообщения выводились на консоль JPAY надо запустить с ключом `-acons`. Если ключ не указан сообщения будут выводится в системный лог. Для вывода на консоль лога ядра L2 используется ключ `-econs`.

АО " Пи Джи групп "

https://aqsi.ru/

info@aqsi.ru

127254, город Москва, Огородный пр-д, д. 8 стр. 2, эт. 1 пом. 8, АО "Пи Джи групп"

+7 (495) 445-96-10

Служба поддержки и технических консультаций:

Техническая поддержка пользователей торгового оборудования и программных продуктов АО "ПИ Джи Групп". Решение проблем, возникающих во время эксплуатации торгового оборудования (ККМ, терминалов и т.п.) и программного обеспечения (от тестовых программ и драйверов до программно-аппаратных комплексов).

Телефон: +7(495) 445-96-10.

E-mail: info@aqsi.ru