Программный интерфейс банковского приложения JPAY

Версия 1.0.7

10 октября 2021 г.

Содержание

1 История изменений

Версия документа	Изменения
1.0.4	1. Добавлена команда directpayment (8.14) 2. Добавлена поддержка обмена данными с ядром L2 (8.9) 3. Добавлены сообщения терминала D1-D5 (8.9) 4. К сообщению display добавлены новые параметры (8.9) 5. В ответ на команду getparameters добавлен тег ashost (8.8) 6. В чек добавлены теги acquirer-tag и bank-name (8.9)
1.0.5	7. В команды transaction и directpayment добавлены сообщения selectaid, getlanguage, getofflinepin, getonlinepin (8.9, 8.14) для обработки на стороне клиента.
1.0.6	8. Изменения в описании команды transaction. Операции без прикладывания карты
1.0.7	9. Добавлено описание лога и данных
1.0.7	10. После успешной сверки итогов записи из лога транзакций удаляются

2 Ссылки

[EMVA] - EMV Contactless Specifications For Payment Systems. Book A.

[EMV4] - EMV Integrated Circuit Card Specifications for Payment Systems. Book 4

[L2EMV] – LIBL2. EMV настройки терминала. Версия 1.0.2

[JCS] - Настройки банковского приложения JPAY. Версия 1.0.4

[EMVC2] - EMV Contactless Specifications for Payment Systems, Book C-2, Kernel 2 Specification

3 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,
6. закрывает соединение.

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

4 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.

5 Ограничение доступа к функциям программного интерфейса

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

6 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.

7 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.

8 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.

8.1 Вход в режим администратора

Пример команды:

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

Пример ответа:

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

8.2 Выход из режима администратора

Пример команды:

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

Пример ответа:

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

8.3 Смена пароля администратора

Пример команды:

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

Пример ответа:

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

8.4 Проверка и загрузка обновлений конфигурации с сервера конфигураций

Пример команды:

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

Пример ответа:

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

8.5 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.

8.6 Загрузка мастер ключей

Пример команды:

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

Пример ответа:

```
<loadmasterkeys>
      <status>ok</status>
      <receipt><pkcv>123456</pkcv><mkcv>789ABC</mkcv></receipt>
</loadmasterkeys>
```

- **pkcv** - KCV мастер ключа PIN
- **mkcv** - KCV мастер ключа MAC

8.7 Проверка связи с сервером авторизации

Пример команды:

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

Пример ответа:

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

8.8 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 хоста или IP сервера эвайера
- **kccert** - дата окончания действия клиентского сертификата для подключения к серверу загрузки ключей
- **ksca** - дата окончания действия CA сервера загрузки ключей
- **devid** - Идентификатор устройства для эквайинга

8.9 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.

```
<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>
</transaction>
```

Если номер отменяемой транзакции имеет значение last. Используется для отмены последней транзакции. Поддерживается всеми эквайерами.

В. Возврат без прикладывания карты

```
<transaction>
      <type>refund</type>
```

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

Поддерживается некоторыми эквайерами для возврата без прикладывания карты. Следует добавить атрибут forced = true.

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>
            <appname>VISA Classic</appname>
            <pan>************1234</pan>
            <aed>2003</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
- **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)
- **appname** - имя приложение EMV карты
- **pan** - последние 4 цифры номера карты
- **aed** - дата окончания срока действия карты (ггмм)
- **rrn** - retrieval reference number
- **resp-code** - код ответа сервера авторизации
- **auth-number** - код авторизации.
- **datetime** - дата и время проведения транзакции (yymmddHHMMSS)
- **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 [JCS].
- **bank-name** – имя банка. Копируется в чек из конфигурации [JCS].

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**, **getonlinepin**, **getofflinepin**, **selectaid** и **getlanguage** Если в ответ на сообщение **keepalive** терминалу возвращается **status=**cancelled терминал прерывает обработку транзакции. 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** приходит от терминала во время исполнения транзакции, если это разрешено в общих настройках терминала [JCS] **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"><code>
      </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"><code>
      </terminal-tags>
</dex>
```

Сообщение **getonlinepin** содержит запрос на ввод онлайн ПИН-кода. Клиент должен обеспечить безопасный ввод ПИН-кода и вернуть ПИН-блок в ответе на сообщение.

В параметрах сообщения передается PAN необходимый для формирования ПИН-блока:

```
<getonlinepin>
      <pan>1234567890123456</pan>
</getonlinepin>
```

Ответ содержит зашифрованный рабочим PEK ключом ПИН-блок в формате ISO-9564-0:

```
<getonlinepin>
      <status>ok</status>
      <pin-block>D42082B4AE20F603</pin-block>
```

```
</getonlinepin>
```

Рабочий ключ PEK используется по умолчанию или задается в настройках приложения. Тег **status** должен содержать значение **ok** в случае успешного ввода ПИН-кода, **pinentrybypassed** в случае, если ввод ПИН-кода отменен пользователем или **pinpadmalfunctionornotpresent** в случае, если ввод ПИН-кода не возможен по техническим причинам.

Сообщение **getofflinepin** содержит запрос на ввод оффлайн ПИН-кода контактной карты. Клиент должен обеспечить безопасный ввод ПИН-кода и вернуть ПИН-блок в ответе на сообщение.

В параметрах сообщения передается счетчик оставшихся попыток ввода ПИН-кода **ptc**:

```
<getofflinepin>
      <ptc>03</ptc>
</getofflinepin>
```

Ответ содержит зашифрованный сессионным ключом PEK ключом ПИН-блок в формате ISO-9564-2:

```
<getofflinepin>
      <status>ok</status>
      <pin-block>D42082B4AE20F603</pin-block>
</getofflinepin>
```

Сессионный ключ PEK используется по умолчанию или задается в настройках приложения. Клиент должен создавать новый сессионный ключ каждый раз перед вводом ПИН-кода.

Тег **status** должен содержать значение **ok** в случае успешного ввода ПИН-кода, **pinentrybypassed** в случае, если ввод ПИН-кода отменен пользователем или **pinpadmalfunctionornotpresent** в случае, если ввод ПИН-кода не возможен по техническим причинам.

Сообщение **selectaid** содержит запрос выбора приложения на контактной карте. Сообщение содержит список имен приложений составленный из тегов Preferred Name, Application Name или Application Identifier.

```
<selectaid>
      <aids>
            <aid>Mir</aid>
            <aid>Maestro</aid>
      </aids>
</selectaid>
```

Клиент должен обеспечить вывод на экран терминала списка приложений в том порядке, в котором они перечислены в сообщении, выбор одного из сообщений и передачу порядкового номера выбранного приложения в теге **selected-aid** в ответе на сообщение. Если сообщение не выбрано, (пользователь отменил выбор) следует вернуть 0.

```
<selectaid>
      <status>ok</status>
      <selected-aid>1</selected-aid>
</selectaid>
```

Сообщение **getlanguage** содержит запрос выбора языка для обработки транзакции по контактной карте. Сообщение содержит список предпочитаемых языков для отображения сообщений в формате ISO-639.

```
<getlanguage>
      <preferred-language>ruen</preferred-language>
</getlanguage>
```

Клиент должен обеспечить выбор языка и вернуть его в ответе на сообщение.

```
<getlanguage>
      <status>ok</status>
      <language>ru</language>
</getlanguage>
```

8.10 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>
                        <aex>2112</aex>
                        <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>1
            <purchase-sum>000000010000</purchase-sum>
            <refund-count100</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>1
      <purchase-total-sum>000000010000</purchase-total-sum>
      <refund-ctotal-ount100</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
  - **aex** - дата окончания срока действия карты (ггмм)
  - **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-count** - общее количество операций по карте
  - **total-card-sum** - баланс всех операций по карте. Баланс это 12 цифр со знаком. Знак плюс обозначается символом 'D'. Знак минус символом 'C'.
- **grand-totals** - итоговые данные очета:
  - **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-count** - общее количество операций
- t**otal-sum** - баланс всех операций. Баланс это 12 цифр со знаком. Знак плюс обозначается символом 'D'. Знак минус символом 'C'.

8.11 Сверка итогов

Пример команды.

```
<settlement>
      <password>12345678</password>
</settlement>
```

Пример ответа

```
<settlement>
      <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>
            <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
- **resp-code** - код ответа сервера авторизации
- a**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

8.12 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)

8.13 Сброс пароля

Пример команды:

```
<resetpassword/>
```

Пример ответа:

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

8.14 Прямой вызов ядра Level 2

Команда используется для работы с библиотекой libL2 напрямую, в случаях, когда не требуется использовать функционал банковского приложения. В параметрах команды передается в формате HEX-ASCII двоичный запрос **trd** (Transaction Related Data).

Пример команды:

```
<directpayment>
      <trd>9C01009F0206000000000100…</trd>
</directpayment>
```

Команда обрабатывается так же как команда **transaction**. В процессе работы клиент получает сообщения **keepalive**, **display**, **dex**, **getonlinepin**, **getofflinepin**, **selectaid** и **getlanguage**. Клиент должен их обработать и вернуть ответ, как это описано в параграфе 8.9. Дополнительно к этим сообщениям, клиент может получить запрос **onlinerequest** на онлайн обработку транзакции, если ядру требуется онлайн авторизация транзакции. Внутри тега сообщения передается тег **request.** Атрибут **iface** указывает тип интерфейса, по которому производится транзакция. Возможные значения: contactless, contact, magstripe. Значение тега **request** содержит двоичные данные преобразованные в строку HEX-ASCII. Двоичные данные включают EMV тег FF8105 - Data Record который содержит набор EMV тегов для формирования финансовой транзакции и тег FF8106 - Discretionary Data. В Discretionary Data, в случае если это требуется, передается зашифрованный онлайн пин блок (EMV тег D0).

Пример сообщения:

```
<onlinerequest>
      <request iface="contactless">FF810581AD9F0206000000002300…</request>
</onlinerequest>
```

Подробное описание тега приводится в [EMVC2].

Клиент должен вернуть EMV данные полученные от сервера в том же формате в котором он получил данные от ядра. Внутри тега **onlinerequest** ответа следует поместить тег **status** с с кодом ошибки и тег **response** значение которого - HEX-ASCII строка созданная из данных сервера. Как минимум данные должны содержать тег 8A - Authorization Code. Для бесконтактного ядра значение 8A = 3030 (“00”) означает “одобрено”, остальные значения - “отклонено”. Значение 8A=5A33(“Z3”) означает что онлайн запрос не может быть выполнен из за отсутствия связи. Пример ответа клиента на сообщение **onlinerequest**:

```
<onlinerequest>
      <status>ok</status>
      <response>8A023030</response>
</onlinerequest>
```

Ответ на команду **directpayment** содержит тег **final-outcome.** Атрибут **iface** указывает тип интерфейса, по которому производится транзакция. Возможные значения: contactless, contact, magstripe. Значение **final-outcome** это HEX-ASCII строка преобразованная из двоичного блока данных, который может содержать теги:

DF8129 - Outcome Parameters Set. Содержит статус транзакции. Для **iface**=magstripe этот тег отсутствует, так как результат транзакции определяется по коду ответа хоста.

FF8105 - Data Record. Содержит EMV теги для формирования batch записи финансовой транзакции. Набор EMV тегов определяется платежной системой. Вместе с этими EMV тегами в Data Record может присутствовать признак запроса подписи (D1 подпись требуется если значение тега равно 01).

FF8106 - Discretionary Data - необязательный EMV тег, содержащий различную информацию в зависимости от типа платежной системы. Например информацию об ошибках. Может содержать флаг запроса подписи D1. Подпись требуется если значение тега D1 равно 01.

Подробное описание EMV тегов приводится в [EMVC2]. В ответ также добавляется код ошибки.

Пример ответа на команду **directpayment**:

```
<directpayment>
      <status>ok</status>
      <final-outcome>DF81290830…</final-outcome>
</directpayment>
```

9 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

```
<keep-alive>
      <reset-connected-indicator>0</reset-connected-indicator>
</keep-alive>
```

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. Сертиикаты 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>
```

10 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** - отменемая 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

11 Приложение Б. Типы транзакций

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

12 Приложение В. 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.

13 Приложение Г. Данные

Данные приложения хранятся в каталоге cardcore в файле jpay.db. Расположение каталога зависит от платформы. Данные хранятся зашифрованными.