

ECR (electronic cash registrar) протокол ПриватБанк (JSON based)

Зміст:

1. Опис фізичного рівня протоколу.
2. Опис каналного протоколу передавання даних.
3. Архітектура роботи. Розробка клієнтського застосунку.
4. Прикладний рівень.
5. Методи
 - 5.1 Операція «Оплата»
 - 5.2 Операція «Повернення»
 - 5.3 Операція «Скасування»
 - 5.4 Операція «Часткове скасування»
 - 5.5 Операція «Перевірка зв'язку»
 - 5.7 Операція «Читання дисконтної картки»
 - 5.8 Операція «Версія ПЗ»
 - 5.9 Операція «Ping»
 - 5.10 Операція «Баланс»
 - 5.11 Операція «Сервіс повернення»
 - 5.12 Операція «Сервіс оплата частинами» (Pay by parts)
 - 5.13 Операція «Сервіс повернення оплати частинами»
 - 5.14 Інші сервіси
 - 5.15 Універсальний сервіс
 - 5.16 Cashback (продаж із видачею готівки)
 - 5.17 Audit (X – balance)
 - 5.18 Verify (Загальний звіт)копія чек
 - 5.19 Копія чека «Звірка»
 - 5.20 Отримання чека

- 5.21 Друк інформації про всі транзакції в пакеті
- 5.22 ReadBonusCard
- 5.23 GetPinBonusCard
- 5.24 PinChangeBonusCard
- 5.25 GetPhoneNumber (Введення номеру телефону)
- 5.26 Preauthorization (Преавторизація)
- 5.27 SaleCompletion (Завершення преавторизація)
- 5.28 Операція Partial approval
- 5.29 GetOTPPassword (Введення OTP паролю)
- 5.30 GetReceiptInfo (Отримання даних по чеку)

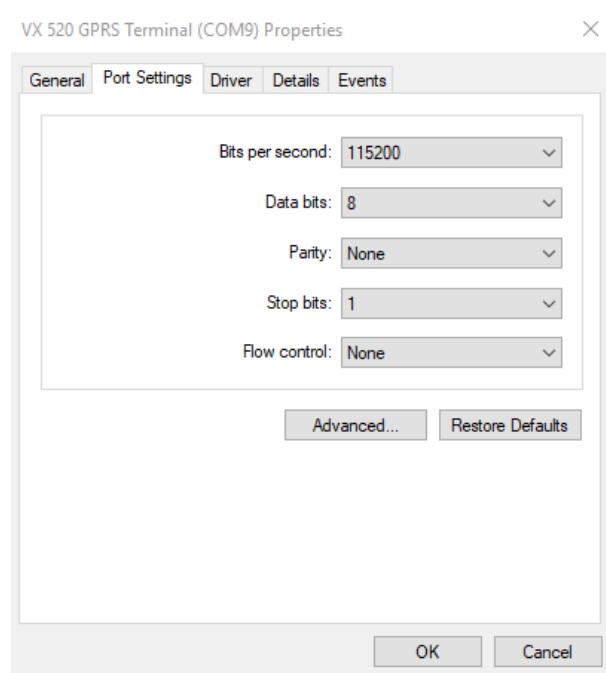
6. Службові повідомлення

- 6.1 Повідомлення deviceBusy
- 6.2 Переривання повідомлення
 - 6.2.1 Відповідь на повідомлення interrupt - підтвердження interruptTransmitted.
- 6.3 Повідомлення methodNotImplemented
- 6.4 Повідомлення getMerchantList
- 6.5 Повідомлення getMaskList
- 6.6 Сервісне повідомлення debug
- 6.7 Service message getLastResult
- 6.8 Service message getLastStatMsgCode
- 6.9 Service message getLastStatMsgDescription
- 6.10 Service message Identify
- 6.11 Service message getDiscountName
- 6.12 Service message correctTransaction
 - 6.12.1 Відповідь на повідомлення correctTransaction - підтвердження correctionTransmitted

1. Опис фізичного рівня протоколу

Термінал, залежно від апаратних можливостей конкретної моделі, під'єднується до касового апарата (або програмного забезпечення, яке емулює) через USB порт, апаратний COM порт або за допомогою Ethernet/WIFI (IEEE 802.x).

За наявності різних інтерфейсів одночасно, на стороні терміналу передбачено можливість перемикання між ними шляхом налаштування суміщення на стороні банку. Для OS сімейства Windows USB-підключення передбачає встановлення стандартного драйвера вендора. При цьому в системі створюється віртуальний COM-порт (usbserial bridge). Формула налаштувань COM порту така: **115200, 8N1**.



Для Linux використовується індивідуальне налаштування COM/USB. Як загальні рекомендації, під час під'єднання через USB необхідно використовувати стандартний usbserial драйвер. У разі не визначення, можна вказати id пристрою вручну:

```
modprobe usbserial vendor=0x11ca product=0x0222 (для Verifone vx520)
```

Також корисно відключити/видалити modemmanager, що істотно прискорить визначення/перевизначення пристрою UDEV. За бажання, можна прописати і власні правила UDEV.

Не виключається робота клієнтського застосунку з USB безпосередньо, за вибором інтегратора. Для цього рекомендується використання бібліотеки **libusb**. Ця бібліотека - кросплатформна.

Робота через мережеві інтерфейси передбачає використання протоколу TCP. Порт за замовчуванням - 2000. У налаштуваннях терміналу є можливість змінити номер порту для касового суміщення, а також увімкнути/вимкнути DHCP.

Описувати власне фізику роботи послідовних портів у цьому документі не вбачається за доцільне, оскільки низькорівневе програмування пристроїв із використанням сигналів DTE/DCE, прямого програмування портів, переривань тощо давно поступилося місцем програмуванню з використанням високорівневих мов, які оперують абстракціями на кшталт Open, Close, Read, Write. Конкретна реалізація залежатиме від обраної мови програмування й операційної системи. Спеціальних команд прикладного рівня для керування відкриттям і закриттям з'єднання не потрібно в тому сенсі, що програміст самостійно керує відкриттям і закриттям порту, і надсилання JSON-повідомлення для цього не потрібне.

2. Опис каналного протоколу передавання даних

Основа цього протоколу (далі - Протокол) - формат обміну даними JSON. Для спрощення обміну даними, байтові послідовності JSON завершуються символами-термінаторами (делімітерами) 0x00 (як рядки в мові C). Застосування делімітера покликане слугувати заміною керуючим ASCII кодам, що застосовуються в подібних протоколах. Протокол єдиний для всіх типів підключення. Внаслідок цієї установки, у разі роботи з послідовним портом, він виконує роль протоколу каналного рівня. У разі IEEE 802.x, оскільки дейтаграми упаковуються в TCP/IP, Протокол стає прикладним, не змінюючись за змістом.

Важливо!

Під час використання мережевого типу підключення, необхідно враховувати таке:

- Термінал не є повноцінним мережевим сервером. У Протоколі відсутнє управління доступом і запобігання колізіям.
- Для реалізації такого функціонала необхідно розробляти власне ПЗ - транслятор у websocket, інший обраний протокол/транспорт.
- Існує еталонна схема підключення клієнта до терміналу, яка використовується в емуляторі каси ПриватБанк:
 1. Коннект на IP, порт
 2. відправляємо хендшейк (див. далі), перевіряємо відповідь
 3. Дисконект
 - 4 Відправляємо Identify (див. далі), перевіряємо відповідь (можливе розгалуження коду, відповідно до отриманих у відповіді найменувань вендора і моделі терміналу)
 5. Дисконект
 6. Далі - основний режим роботи: відкриваємо з'єднання, тримаємо його відкритим (keepalive) і слухаємо вхідні дані ПОСТІЙНО, розділяючи дейтаграми за символом 0x00
 7. Отриманий на етапі 6 дескриптор з'єднання, зберігаємо, як глобальний, і використовуємо його для читання, в міру потреби. НЕ ВІДКЛЮЧАЄМОСЬ
 8. За бажання, проводиться перевірка з'єднання і перепідключення в разі виявлення обриву (див. пункт 1)
- Аналогічна за структурою схема застосовується і під час роботи через USB/COM
- Ця схема проходить тестування і гарантує роботу з терміналами будь-яких вендорів, що підтримують Протокол. В основі схеми - постійний монопольний доступ із підтримкою з'єднання і перепідключенням у разі обриву.
- Інші способи роботи можливі, але не гарантуються.

Важливо!

Як значення того чи іншого поля JSON усі не-ASCII символи, зокрема кирилиця, передаються в кодуванні utf-8.

Наприклад:

Вихідний JSON рядок:

```
{"method":"GetTerminalInfo","step":0}
```

Вихідний JSON [] byte:

```
0x7b, 0x22, 0x6d, 0x65, 0x74, 0x68, 0x6f, 0x64, 0x22, 0x3a, 0x22, 0x47, 0x65, 0x74, 0x54, 0x65,  
0x72, 0x6d, 0x69, 0x6e, 0x61, 0x6c, 0x49, 0x6e, 0x66, 0x6f, 0x22, 0x2c, 0x22, 0x73, 0x74, 0x65,  
0x70, 0x22, 0x3a, 0x30, 0x7d
```

NULL-terminated JSON:

```
0x7b, 0x22, 0x6d, 0x65, 0x74, 0x68, 0x6f, 0x64, 0x22, 0x3a, 0x22, 0x47, 0x65, 0x74, 0x54, 0x65,  
0x72, 0x6d, 0x69, 0x6e, 0x61, 0x6c, 0x49, 0x6e, 0x66, 0x6f, 0x22, 0x2c, 0x22, 0x73, 0x74, 0x65,  
0x70, 0x22, 0x3a, 0x30, 0x7d, 0x00
```

Процедура хендшейку

Процедура хендшейка здійснюється за допомогою передавання повідомлення Ping:

```
{"method":"PingDevice","step":0}
```

Однак це повідомлення відрізняється додатковим термінатором 0x00 на початку переданої послідовності:

Written data (port):

```
00 7b 22 6d 65 74 68 6f 64 22 3a 22 50 69 6e 67  
44 65 76 69 63 65 22 2c 22 73 74 65 70 22 3a 30  
7d 00
```

Відповідь на це повідомлення:

```
{"method":"PingDevice","step":0,"params":{"code":"00","responseCode":"0000"},"error":false,  
"errorDescription":""}
```

Read data (port):

```
7b 22 6d 65 74 68 6f 64 22 3a 22 50 69 6e 67 44  
65 76 69 63 65 22 2c 22 73 74 65 70 22 3a 30 2c  
22 70 61 72 61 6d 73 22 3a 7b 22 63 6f 64 65 22  
3a 22 30 30 22 2c 22 72 65 73 70 6f 6e 73 65 43  
6f 64 65 22 3a 22 30 30 30 30 22 7d 2c 22 65 72  
72 6f 72 22 3a 66 61 6c 73 65 2c 22 65 72 72 6f  
72 44 65 73 63 72 69 74 70 74 69 6f 6e 22 3a 22  
22 7d 00
```

У результаті такого обміну пристрій вважається виявленим, процедура хендшейку - пройденою.

Разом запит-відповідь:

```
1 Written data (port):
2
3 00 7b 22 6d 65 74 68 6f 64 22 3a 22 50 69 6e 67
4 44 65 76 69 63 65 22 2c 22 73 74 65 70 22 3a 30
5 7d 00
6
7 .{"method":"PingDevice","step":0}.
8
9
10 Read data (port):
11
12 7b 22 6d 65 74 68 6f 64 22 3a 22 50 69 6e 67 44
13 65 76 69 63 65 22 2c 22 73 74 65 70 22 3a 30 2c
14 22 70 61 72 61 6d 73 22 3a 7b 22 63 6f 64 65 22
15 3a 22 30 30 22 2c 22 72 65 73 70 6f 6e 73 65 43
16 6f 64 65 22 3a 22 30 30 30 30 22 7d 2c 22 65 72
17 72 6f 72 22 3a 66 61 6c 73 65 2c 22 65 72 72 6f
18 72 44 65 73 63 72 69 74 70 74 69 6f 6e 22 3a 22
19 22 7d 00
20
21 {"method":"PingDevice","step":0,"params":{"code":"00","responseCode":"0000"},"error":false,"errorDescription":""}.
22
23 Close port.
```

Усі інші дейтаграми передаються з NULL-термінатором тільки наприкінці JSON повідомлення. Самі JSON повідомлення передаються у вигляді бінарного протоколу як послідовності байтів.

Identify (рекомендовано використовувати після хендшейка)

Приклад обміну даними:

{"method":"ServiceMessage","step":0,"params":{"msgType":"identify"}}

Data to send: =====

7b 22 6d 65 74 68 6f 64 22 3a 22 53 65 72 76 69 63 65 4d 65 73 73 61 67 65 22 2c 22 73 74
65 70 22 3a 30 2c 22 70 61 72 61 6d 73 22 3a 7b 22 6d 73 67 54 79 70 65 22 3a 22 69 64 65
6e 74 69 66 79 22 7d 7d 00

=====

**{"method": "ServiceMessage", "step": 0, "params": {"msgType": "identify", "result":
"true", "vendor": "PAX", "model": "s800"}, "error": false, "errorDescription": ""}**

RCV: =====

7b 22 6d 65 74 68 6f 64 22 3a 20 22 53 65 72 76 69 63 65 4d 65 73 73 61 67 65 22 2c 20 22
73 74 65 70 22 3a 20 30 2c 20 22 70 61 72 61 6d 73 22 3a 20 7b 22 6d 73 67 54 79 70 65 22
3a 20 22 69 64 65 6e 74 69 66 79 22 2c 20 22 72 65 73 75 6c 74 22 3a 20 22 74 72 75 65 22
2c 20 22 76 65 6e 64 6f 72 22 3a 20 22 50 41 58 22 2c 20 22 6d 6f 64 65 6c 22 3a 20 22 73
38 30 30 22 7d 2c 20 22 65 72 72 6f 72 22 3a 20 66 61 6c 73 65 2c 20 22 65 72 72 6f 72 44
65 73 63 72 69 70 74 69 6f 6e 22 3a 20 22 22 7d 00

=====

3. Архітектура роботи. Розробка клієнтського застосунку.

Розроблення клієнтського застосунку необхідне для реалізації під'єднання каси безпосередньо до термінала з використанням Протоколу, а також, можливо, для імплементації додаткового функціоналу, наприклад, протоколу WebSocket, HTTP(S) тощо.

Сервер може складатися з частини, що пише і слухає, які працюють у дуплексі.

Слухальна частина (Listener) підтримує **ПОСТІЙНЕ** під'єднання до термінала і формує буфер даних після досягнення NULL - термінатора (делімітера). У разі втрати з'єднання - проводиться спроба його відновлення.

Приклад (мовою Go):

```
//Listener carries out a permanent TCP connection to get raw data from terminal
(upon receiving).
//Removes delimiter (0x00). Sends data to channel (BuffHandler() func is the
receiver).
//Carries out disconnections
//Must be used as a goroutine
func Listener() {

    var err error

    Gconn, err = net.Dial("tcp", GconnOption)
    if err != nil {
        log.Fatalf("net.Dial: %v", err)
    }

    log.Println("Listening Network...")

    answ := make([]byte, 1024)

    buf := bytes.Buffer{}

    for {
        i := 0
        i, err = Gconn.Read(answ)
        if err != nil && err != io.EOF {
            log.Printf("conn.Read: %v", err)
            Gconn.Close()
            go reconnect() // reconnect
            break
        }
        if i != 0 {
            answ = answ[:i]
            buf.Write(answ)
            if buf.Bytes()[len(buf.Bytes())-1] == 0x00 { //delimiter
                trace(buf.Bytes()) //if debug, write trace
                a := buf.Bytes()[:len(buf.Bytes())-1] //remove delimiter
                buffer <- a
                buf.Reset()
            }
        }
    }
}
```


Обробник (з передаванням даних у websocket)

```
//BuffHandler gets data via channel from the Listener func (upon receiving).
//Transmits terminal answers to websocket.
//Must be used as goroutine.
func BuffHandler() {
    for {
        buf := <-buffer
        conn := *wsconn
        err := conn.WriteMessage(websocket.TextMessage, buf)
        if err != nil {
            log.Println("Error in BuffHandler: ", err)
        }
    }
}
```

Writer

```
//prepare method ping
if getJSON.Method == "PingDevice" && getJSON.Step == 0 && !busy {

    go func() {

        var p jsonEntityNoParams
        p.Method = "PingDevice"
        p.Step = 0

        data, err := json.Marshal(p)
        if err != nil {
            log.Println("Error marshaling JSON: ", err)
            return
        }
        data = append(data, 0x00)// add terminator

        err = writeEth(data)
        if err != nil {
            log.Printf("conn.Write: %v", err)
        }
    }()
}

//writer
func writeEth(buffer []byte, timeout ...int) error {

    var n int
    var err error

    var t Stopwatch
    t.Start() // start timer
    defer t.Stop() // sure to stop timer

    //set timeout
    var tout int
    if len(timeout) == 0 {
        tout = 30
    } else {
        tout = timeout[0]
    }

    if connIsClosed() {
        if err = Gconn.Close(); err == nil {
```

```

        err = errors.New("disconnection")
    }
    return err
}

for {
    if err = Gconn.SetWriteDeadline(time.Now().Add(5 * time.Second)); err !=
nil {
        continue
    }
    n, err = Gconn.Write(buffer)
    if err != nil {
        log.Println("Error in WriteEth(): ", err)
        sleep(1)
    } else {
        break
    }
    if t.GetElapsed() >= time.Duration(tout)*time.Second {
        err = errors.New("connection timeout")
        break
    }
}

if err != nil {
    log.Printf("TCP write: %v", err)
    err = errors.New("write error")
}

fmt.Println("Wrote", n, "bytes.")

trace(buffer) // trace with delimiter NB!

return err
}

```

Методи категорії **ServiceMessage** є асинхронними і можуть бути виконані в будь-який момент часу, навіть у момент виконання іншого методу. Інші методи перехоплюють процес і до свого завершення повертають **ServiceMessage Busy**. Таким чином, під час реалізації інтегратором роботи з терміналом, необхідно розуміти, що пристрій має працювати з 3 логічними потоками даних

1. Поточна операція
2. Повідомлення служби обміну
3. У відповідь на конкуруючий запит, що не належить до категорії сервісних, відповідь: **ServiceMessage Busy**.

Сказане вище не означає, що до терміналу можна звертатися з різних кас паралельно. Описані 3 логічні потоки існують у межах 1 конекту. Для здійснення запиту від другого, третього... N клієнта потрібно повністю звільняти термінал і проводити щоразу процедуру хендшейка.

Ці особливості потрібно завжди враховувати як під час обміну даними між касою і терміналом у монопольному режимі, так і під час розроблення middleware-засобів для організації множинного доступу до терміналу.

4. Прикладний рівень

Взаємодія за допомогою JSON

Попри наявність більш продуктивних і сучасних альтернатив, наприклад, таких, як Protocol Buffers, запропонований Google, з погляду поширеності, сумісності та зручності імплементації, формат JSON є наразі фактичним стандартом формату серіалізації (передання) структурованих даних.

У загальному вигляді команди для терміналу мають такий вигляд:

```
{  
  "method": "DoSomething",  
  "step": 0,  
  "params": {  
    "par1": "xxx",  
    "parN": "yyy"  
  }  
}
```

Відповідь на цей запит:

```
{  
  "method": "DoSomething",  
  "step": 0,  
  "params": {  
    "answer": "OperationResults"  
  },  
  "error": true,  
  "errorDescription": "TIMEOUT"  
}
```

Цей JSON приведений до єдиного типу і легко десеріалізується:

C#:

=====

```
public class Params  
{  
    public string answer { get; set; }  
}
```

```
public class RootObject
{
    public string method { get; set; }
    public int step { get; set; }
    public Params @params { get; set; }
    public bool error { get; set; }
    public string errorDescription { get; set; }
}
```

=====

GO:

=====

```
type jsonEntity struct{
    Method string `json:"method"` // tag descriptors
    Step int `json:"step"`
    Params map[string]string `json:"params"`
    Error bool `json:"error"`
    ErrorDescription string `json:"errorDescription"`
}
```

=====

Емулятор касового апарату, приклад

:

Данные

OPEN

RESPONSE: {"method":"PingDevice","step":0,"params":{"code":"00"},"error":false}

Устройство подключено

RESPONSE: {"method":"GetTerminalInfo","step":0,"params":{"code":"00","version":"T5PRVa_03K TKASPRBK80321-922-618"},"error":false}

Firmware: T5PRVa_03K TKASPRBK80321-922-618

method - назва проведеної операції

step - зарезервовано для багатопрохідних операцій (в однопрохідних операціях завжди дорівнює «0»)

params - масив додаткових параметрів для термінала

error - присутність помилки (true/false)

errorDecription - опис помилки

Важливо!

У цій реалізації API помилки завжди описуються двома полями:

- поле **error** має тип **bool** і, відповідно, може набувати значень **true** і **false**.
- поле **errorDescription** - текстове, і містить опис помилки та/або її код.

У разі прямого під'єднання каси до термінала, тобто без використання ПЗ для суміщення, таймауту, обриви й інші помилки зв'язку перехоплюють на відповідному рівні порту або мережевої моделі, і їх обробляє інтегратор на програмному рівні.

Важливо!

У разі **responseCode** \geq 1000, а саме:

1000 – General error (should be used in exceptional case)
1001 – Transaction canceled by user
1002 – EMV Decline
1003 – Transaction log is full. Need close batch
1004 – No connection with host
1005 – No paper in printer
1006 – Error Crypto keys
1007 – Card reader is not connected
1008 – Transaction is already complete

Відповідь приходить у скороченому вигляді (без чека й інших атрибутів транзакції)

```
{  
  "method": "Purchase",  
  "step": 0,  
  "params": {  
    "responseCode": "1001"  
  },  
  "error": true,  
  "errorDescription": "Transaction canceled by user"  
}
```

За деяких **responseCode**, наприклад, 05, 96 тощо, на касу передають чек, але відсутня низка параметрів, таких як **RRN**, **ApprovalCode** тощо. У таких випадках відповідь термінала і чек повинні відповідати таким у **WPOS**.

При **responseCode** відмінних від 0000 термінал передає **"error":true**
Виключенням є **RC=10**, який може бути при **Partial approval** або при відміні операції **Cashback**, але успішному проведенні оплати, в даному випадку передавати **"error":false**
Також виключенням є метод **GetReceiptInfo**, при його виконанні значення **responseCode** дублюється з транзакції по якій робиться запит інформації, а значення **error** відноситься до методу **GetReceiptInfo**.

Purchase, код 5, приклад (Verifone):

```
{  
  "method": "Purchase",  
  "step": 0,  
  "params": {  
    "receipt": "ПРИВАТБАНК\\nDNIPRO\\nSALECPL\\nMUDROGO 5  
12222\\nЕ\\€\\€\\__\\J\\€J\\€J\\J\\J\\J3333\\n\\n\\nВ І Д Х И Л Е  
Н О\\n\\nАВТОРІЗАЦІЮ ВІДХИХАНЕНО!\\nЗВЕРНІТЬСЯ В  
БАНК-ЕМІТЕНТ КАРТКИ\\nКОД 05\\n\\n\\n\\n\\n\\n\\n\\n\\nMASTER  
\\n*****9732\\nMasterCardAID: A0000000041010\\nТЕРМІНАЛ #  
S11103W202 Жовтень 2019 16:38:11\\nКОД АВТОРІЗ.: DECLINEЧЕК N:  
051886\\n\\n\\n(c)SSI 2018 PRIVATBANK v.T5PRVa_.03K\\nSmartPos_EMV  
01.001",  
    "responseCode": "0005"  
  },  
  "error": true,  
  "errorDescription": "ВІДХИЛЕНО! ЗВЕРНІТЬСЯ ДО БАНКУ-ЕМІТЕНТА  
КАРТКИ"  
}
```

Важливою відмінністю цього протоколу є відсутність конфірмації й видачі чека на касу як окремих сутностей. У ситуації, де мається на увазі confirm - використовується автопідтвердження. Усі операції, що передбачають повернення чека на касу, використовують формат запит-відповідь, минаючи окрему вимогу видачі чека.

5. Методи

5.1 Операція «Оплата»

Це основна операція термінала, призначена для оплати товару.

5.1.1 Запит на оплату:

```
{  
  "method": "Purchase",  
  "step": 0,  
  "params": {  
    "amount": "0.60",  
    "discount": "",  
    "merchantId": "0",  
    "facepay": "false", // true  
    "subMerchant": "" // поле можна передавати тільки після  
    передання субмерчанта менеджеру банку і внесення його в  
    довідник  
  }  
}
```

amount - сума проведеної транзакції,

discount - додаткова сума (знижка)

merchantId - індекс мерчанта, який використовується для транзакції

facepay - параметр, що запускає з каси оплату через FacePay24 (для Android терміналів з підтримкою цього сервісу)

Важливо! У полі Amount допускається поділ суми (грн/коп) тільки через знаки "." або ",",

ПЗ термінала при отриманні від каси в полі amount будь-яких символів, відмінних від "." або ",", має переривати продаж і повертати на касу відповідь:

"responseCode": "1000"

"errorDescription": "Операція неможлива"

Сума, що надається у форматі дійсного числа, приводиться до рядка, кількість розрядів у дробовій частині - 2

0 -> "0.00"

5.1.2 Відповідь на запит

Термінал обробить транзакцію як звичайну карткову операцію і передасть касі код відповіді, отриманий від хоста. Після зв'язку з хостом термінал генерує повідомлення, наведене нижче. Якщо каса друкує чеки, термінал надішле їй усі необхідні дані, інакше буде відіслано тільки поле «**responseCode**», а решта — порожні. Увімкнення друку чеків касою або терміналом налаштовується в профілі терміналу.

```
{
  "method": "Purchase",
  "step": 0,
  "params": {
    "amount": "0.60",
    "approvalCode": "999999",
    "captureReference": "",
    "cardExpiryDate": "2020",
    "cardHolderName": "INSTANT/ISSUE",
    "date": "02.10.2019",
    "discount": "0.00",
    "hstFld63Sf89": "",
    "invoiceNumber": "999999",
    "issuerName": "VISA ПРИВАТ",
    "merchant": "TSTTTTTT",
    "pan": "4731XXXXXXXXX9838",
    "posConditionCode": "00",
    "posEntryMode": "022",
    "processingCode": "000000",
    "receipt": "text-of-receipt",
    "responseCode": "0000",
    "rrn": "99999999999999",
    "rrnExt": "88888888888888",
    "terminalId": "TSTSALE2",
    "time": "09:11:07",
    "track1": "",
    "signVerif": "0",
    "txnType": "1",
```



```

        "trnStatus": "1",
        "adv": "ПриватБанк",
        "adv2p": "Беремо і робимо!",
        "bankAcquirer": "ПриватБанк",
        "paymentSystem": "VISA",
        "subMerchant": ""
    },
    "error": false,
    "errorDescription": ""
}

```

responseCode - код відповіді термінала

approvalCode - код авторизації

amount - сума проведеної транзакції,

discount - додаткова сума (знижка),

pan - номер картки з урахуванням контрольної цифри

date - дата транзакції

time - час транзакції

cardHolderName - ім'я власника картки

invoiceNumber - номер чека підтвердженої транзакції

issuerName - ім'я емітента, за яким проводилася операція

merchant - номер мерчанта, за яким проводилася операція

posEntryMode - метод введення карти

processingCode - код обробки

captureReference - 17 поле в повідомленні сервера

cardExpiryDate - термін придатності картки

hstFld63Sf89 - дані з хоста (63 поле, 89 підполе)

posConditionCode - код стану термінала

receipt - чек

rrn - унікальний номер транзакції

rrnExt - ідентифікатор транзакції в банку емітента картки (приходить якщо картка не ПриватБанку)

terminalId - ідентифікатор термінала на хості

track1 - перший трек карти

signVerif - перевірка підпису (CVM)

txnType - тип поточної транзакції (*значення згідно з прийнятими в BPOS)

trnStatus - статус транзакції (**значення згідно з прийнятими в BPOS)

adv - перша хостова реклама (поле 63.29)

adv2p - друга хостова реклама (поле 63.29, перша та друга реклама розділені через <p>)

bankAcquirer - найменування банку-еквайєра

paymentSystem - найменування МПС, що передається з хоста

subMerchant - Субмерчант (поле 57.98)

*- *get_TxnType(BYTE* pVal), property TxnType*

[OUT] *pVal* – gets transaction type of current transaction. Intended for usage for

GetTxnDataByInv and GetTxnDataByOrder methods.

TxnType has one of following values:

0 – undefined

1 – Purchase

2 – Refund

3 – Void

*** - get_TrnStatus(BYTE * pVal), property TrnStatus*

[OUT] pVal – Get transaction status

TrnStatus has one of following values:

0 – undefined

1 – approved

2 – declined

3 – reversed

4 – canceled

5.1.3 Підтвердження оплати

У разі необхідності конфірмації, **завжди використовується автопідтвердження операцій. Це правило також вірне для всіх методів, для яких у ВРОС було потрібне підтвердження.**

5.2 Операція «Повернення»

Повернення - операція, зворотна операції «Оплата», призначена для скасування транзакції, навіть якщо вона не міститься в поточному пакеті транзакцій, а була проведена на терміналі раніше.

5.2.1 Запит на повернення:

```
{  
  "method": "Refund",  
  "step": 0,  
  "params": {  
    "amount": "0.30",  
    "discount": "",  
    "merchantId": "0",  
    "rrn": "014721258513",  
    "subMerchant": ""  
  }  
}
```

amount - сума проведеної транзакції,

discount - додаткова сума (знижка)

merchantId - ідентифікатор мерчанта, який використовується для транзакції

rrn – RRN

5.2.2 Відповідь

Результат запиту буде аналогічний відповіді на запит оплати в 5.1.2, за винятком поля **signVerif**

```
{
  "method": "Refund",
  "step": 0,
  "params": {
    "amount": "0.60",
    "approvalCode": "999999",
    "captureReference": "",
    "cardExpiryDate": "2020",
    "cardHolderName": "INSTANT/ISSUE",
    "date": "02.10.2019",
    "discount": "0.00",
    "hstFld63Sf89": "",
    "invoiceNumber": "999999",
    "issuerName": "VISA ПРИВАТ",
    "merchant": "TSTTTTTT",
    "pan": "4731XXXXXXXXX9838",
    "posConditionCode": "00",
    "posEntryMode": "022",
    "processingCode": "000000",
    "receipt": "text-of-receipt",
    "responseCode": "0000",
    "rrn": "99999999999999",
    "rrnExt": "88888888888888",
    "terminalId": "TSTSALE2",
    "time": ""09:11:07",
    "track1": "",
    "txnType": "2",
    "trnStatus": "1",
    "adv": "ПриватБанк",
```

```
        "adv2p": "Беремо і робимо!",
        "bankAcquirer": "ПриватБанк",
        "paymentSystem": "VISA",
        "subMerchant": ""
    },
    "error": false,
    "errorDescription": ""
}
```

5.3 Операція «Скасування»

Операція «Скасування» - призначена для анулювання транзакції в межах поточного пакета транзакцій (операційного дня). Доступ до транзакції, що анулюється, здійснюється за номером чека.

5.3.1 Запит на скасування:

```
{
    "method": "Withdrawal",
    "step": 0,
    "params": {
        "invoiceNumber": "131220"
    }
}
```

5.3.2 Відповідь

Результат відповіді такий самий, як і в п.5.1.2, за винятком поля **signVerif**

```
{
    "method": "Withdrawal",
    "step": 0,
    "params": {
        "amount": "0.60",
        "approvalCode": "999999",
        "captureReference": "",

```

```
        "cardExpiryDate": "2020",
        "cardHolderName": "INSTANT/ISSUE",
        "date": "02.10.2019",
        "discount": "0.00",
        "hstFld63Sf89": "",
        "invoiceNumber": "999999",
        "issuerName": "VISA ПРИВАТ",
        "merchant": "TSTTTTTT",
        "pan": "4731XXXXXXXXX983",
        "posConditionCode": "00",
        "posEntryMode": "022",
        "processingCode": "000000",
        "receipt": "text-of-receipt",
        "responseCode": "0000",
        "rrn": "99999999999999",
        "terminalId": "TSTSALE2",
        "time": "09:11:07",
        "track1": "",
        "txnType": "3",
        "trnStatus": "1",
        "adv": "ПриватБанк",
        "adv2p": "Беремо і робимо!",
        "bankAcquirer": "ПриватБанк",
        "paymentSystem": "VISA",
        "subMerchant": ""
    },
    "error": false,
    "errorDescription": ""
}
```

у методах **Withdrawal** і **WithdrawalPartly** поле **subMerchant** не передається, термінал повертає значення **subMerchant**, яке було передане раніше в методі **Purchase**

5.4 Операція «Часткове скасування»

Операція призначена для часткового скасування за номером чека з введенням суми скасування на касі.

5.4.1 Запит на скасування:

```
{
  "method": "WithdrawalPartly",
  "step": 0,
  "params": {
    "amount": "0.30",
    "invoiceNumber": "131220"
  }
}
```

5.4.2 Відповідь

Результат відповіді такий самий, як і в п.5.1.2, за винятком поля **signVerif**, **txnType**

```
{  
  "method": "WithdrawalPartly",  
  "step": 0,  
  "params": {  
    "amount": "0.30",  
    "approvalCode": "999999",  
    "captureReference": "",  
    "cardExpiryDate": "2020",  
    "cardHolderName": "INSTANT/ISSUE",  
    "date": "09:11:07",  
    "discount": "0.00",  
    "hstFld63Sf89": "",  
    "invoiceNumber": "999999",  
    "issuerName": "VISA ПРИВАТ",  
    "merchant": "TSTTTTTT",  
    "pan": "4731XXXXXXXXX9838",  
    "posConditionCode": "00",  
    "posEntryMode": "022",  
    "processingCode": "000000",  
    "receipt": "text-of-receipt",  
    "responseCode": "0000",  
    "rrn": "99999999999999",  
    "rrnExt": "88888888888888",  
    "terminalId": "TSTSALE2",  
    "time": "09:11:07",
```

```
        "track1": "",
        "trnStatus": "1",
        "adv": "ПриватБанк",
        "adv2p": "Беремо і робимо!",
        "bankAcquirer": "ПриватБанк",
        "paymentSystem": "VISA",
        "subMerchant": ""
    },
    "error": false,
    "errorDescription": ""
}
```

5.5 Операція «Перевірка зв'язку»

Каса генерує повідомлення-запит термінала. При отриманні цього запиту, термінал відобразить список мерчантів. Користувач повинен вибрати зі списку мерчант, для якого необхідно провести тест зв'язку.

5.5.1 Запит:

```
{
    "method": "CheckConnection",
    "step": 0
}
```

5.5.2 Відповідь:

```
{
    "method": "CheckConnection",
    "step": 0,
    "params": {
        "code": "00", //для сумісності
        "responseCode": "0000"
    },
    "error": false,
    "errorDescription": ""
}
```

5.7 Операція «Читання дисконтної картки»

При отриманні цього запиту, термінал запропонує користувачеві прокатати карту через магнітний зчитувач.

5.7.1 Запит:

```
{  
  
  "method": "ReadCardDiscount",  
  
  "step": 0  
  
}
```

5.7.2 Відповідь:

У разі невдалого читання карти всі поля **params** будуть порожні. Опис полів аналогічний 5.6.2

```
{  
  "method": "ReadCardDiscount",  
  "step": 0,  
  "params": {  
    "hash": "",  
    "track1": "",  
    "track2": "99999999999999939",  
    "track3": ""  
  },  
  "error": false,  
  "errorDescription": ""  
}
```

5.8 Операція «Версія ПЗ»

Операція призначена для отримання назви програмного забезпечення, встановленого на термінал

5.8.1 Запит:

```
{  
  
  "method": "GetTerminalInfo",  
  
  "step": 0
```



```
}
```

5.8.2 Відповідь:

```
{  
  "method": "GetTerminalInfo",  
  "step": 0,  
  "params": {  
    "version": "TE7E132 TESTKKM000CT22927302"  
  },  
  "error": false,  
  "errorDescription": ""  
}
```

у полі version передається версія ПЗ термінала і мерчант + с/н термінала

5.9 Операція «Ping»

Ця команда не призначена для буквального "пінгування" обладнання на рівні TCP/IP або перевірки доступності COM порту. Успішне виконання цієї операції свідчить про те, що термінал приймає JSON згідно з описуваним протоколом, а також наразі вільний і готовий до виконання будь-якого методу.

5.9.1 Запит:

```
{  
  "method": "PingDevice",  
  "step": 0  
}
```

5.9.2 Відповідь:

```
{  
  "method": "PingDevice",  
  "step": 0,  
  "params": {  
    "code": "00", // для сумісності  
    "responseCode": "0000",  
  },  
  "error": false,  
  "errorDescription": ""  
}
```

```
}
```

5.10 Операція «Баланс»

5.10.1 Запит:

```
{  
  "method": "GetBalance",  
  "step": 0,  
  "params": {  
    "merchantId": "1" // "0" або "" — виведення всього списку  
  }  
}
```

5.10.2 Відповідь:

```
{  
  "method": "GetBalance",  
  "step": 0,  
  "params": {  
    "balance": "XXX.XX",  
    "code": "00", // для сумісності  
    "responseCode": "0000"  
  },  
  "error": false,  
  "errorDescription": ""  
}
```

У разі невдачі - "balance" : "unknown", "code": "01", "responseCode": "0001"

5.11 Операція «Сервіс повернення»

Ця операція є альтернативою операції «Повернення» (Refund). *Примітка:* код сервісу й id мерчанта зашивається хардкодом (див. Табл. 1.1), зазначати його не потрібно.

5.11.1 Запит

```
{  
  "method": "ServiceRefund",  
  "step": 0,
```

```
"params": {
  "amount": "0.30",
  "rrn": "014721258513",
  "subMerchant": ""
}
}
```

Відповідь: 5.11.2

```
{
  "method": "ServiceRefund",
  "step": 0,
  "params": {
    "amount": "0.60",
    "approvalCode": "999999",
    "captureReference": "",
    "cardExpiryDate": "2020",
    "cardHolderName": "99999999",
    "date": "02.10.2019",
    "discount": "0.00",
    "hstFld63Sf89": "text-of-6389-field",
    "invoiceNumber": "999999",
    "issuerName": "INSTANT/ISSUE",
    "merchant": "TSTTTTTT",
    "pan": "4731XXXXXXXXX9838",
    "posConditionCode": "00",
    "posEntryMode": "022",
    "processingCode": "000000",
    "receipt": "text-of-receipt",
    "responseCode": "0000",
    "rrn": "99999999999999",
    "rrnExt": "88888888888888",
    "terminalId": "TSTSALE2",
    "time": "09:11:07",
    "track1": "",
    "adv": "ПриватБанк",
    "adv2p": "Беремо і робимо!",
    "bankAcquirer": "ПриватБанк",
    "paymentSystem": "VISA",
  }
}
```

```
        "subMerchant": ""
    },
    "error": false,
    "errorDescription": ""
}
```

5.12 Операція «Сервіс оплата частинами» (Pay by parts)

Примітка: код сервісу й id мерчанта зашивається хардкодом (див. Табл. 1.1), зазначати його не потрібно.

5.12.1 Запит

```
{
  "method": "ServicePbP",
  "step": 0,
  "params": {
    "amount": "0.30",
    "amountOfParts": "6",
    "subMerchant": ""
  }
}
```

5.12.2 Відповідь:

```
{
  "method": " ServicePbP ",
  "step": 0,
  "params": {
    "amount": "0.60",
    "approvalCode": "999999",
    "captureReference": "",
    "cardExpiryDate": "2020",
    "cardHolderName": "INSTANT/ISSUE",
    "date": "02.10.2019",
    "discount": "0.00",
    "amountOfParts": "6",
    "hstFld63Sf89": "text-of-6389-field",
    "invoiceNumber": "999999",

```

```
        "issuerName": "VISA ПРИВАТ",
        "merchant": "TSTTTTTT",
        "pan": "4731XXXXXXXXX9838",
        "posConditionCode": "00",
        "posEntryMode": "022",
        "processingCode": "000000",
        "receipt": "text-of-receipt",
        "responseCode": "0000",
        "rrn": "999999999999",
        "terminalId": "TSTSALE2",
        "time": "0941",
        "track1": "",
        "adv": "ПриватБанк",
        "adv2p": "Беремо і робимо!",
        "bankAcquirer": "ПриватБанк",
        "paymentSystem": "VISA",
        "subMerchant": ""
    },
    "error": false,
    "errorDescription": ""
}
```

5.13 Операція «сервіс повернення оплати частинами» - здійснює повернення для сервісу ServicePbP.

5.13.1 Запит:

```
{
  "method": "ServiceRefPbP",
  "step": 0,
  "params": {
    "amount": "350",
    "agreementNum": "12345"
  }
}
```

5.13.2 Відповідь:

```
{
```

```
"method": "ServiceRefPbP",
"step": 0,
"params": {
  "approvalCode": "999999",
  "captureReference": "",
  "cardExpiryDate": "2020",
  "cardHolderName": "INSTANT/ISSUE",
  "date": "02.10.2019",
  "discount": "0.00",
  "invoiceNumber": "999999",
  "issuerName": "VISA ПРИВАТ",
  "merchant": "TSTTTTTT",
  "pan": "4731XXXXXXXXX9838",
  "posConditionCode": "00",
  "posEntryMode": "022",
  "processingCode": "000000",
  "receipt": "text-of-receipt",
  "responseCode": "0000",
  "rrn": "99999999999999",
  "terminalId": "TSTSALE2",
  "time": "09:11:07",
  "track1": "",
  "amount": "350",
  "hstFld63Sf89": "text-of-6389-field",
  "agreementNum": "12345",
  "adv": "ПриватБанк",
  "adv2p": "Беремо і робимо!",
  "bankAcquirer": "ПриватБанк",
  "paymentSystem": "VISA",
  "subMerchant": ""
},
"error": false,
"errorDescription": ""
}
```

5.14 Інші сервіси

За аналогією з наведеними вище сервісами, викликаються інші сервісні операції.
Коди сервісів і id мерчантів також зашиті хардкодом.

- **ServicePbPperiod** (ОЧ у періоді)
- **ServiceRefPbPperiod** (повернення ОЧ у періоді)
- **ServicePartlyRefPbPperiod** (часткове повернення ОЧ у періоді)
- **ServicePartlyRefPbP** (часткове повернення ОЧ)
- **ServiceInstantPbl** (миттєва розстрочка) // Pay by installment
- **ServiceRefPbl** (повернення МР)
- **ServicePartlyRefPbl** (часткове повернення МР)
- **ServicePblAct** (миттєва розстрочка акція)
- **ServiceRefPblAct** (повернення МР акція)
- **ServicePartlyRefPblAct** (часткове повернення МР акція)

Відмінність полягає в назві методу, а також у варіюваному параметрі - **agreementNum**, або **amountOfParts**

Таблиця 1.1 - Хардкодні номери сервісів і id мерчантів:

Service ID	Service name	Опис	MerchantID
054	ServiceRefund	Сервіс повернення	5
046	ServicePbP	Сервіс оплата частинами (ОЧ)	6
062	ServiceRefPbP	Сервіс оплата частинами повернення	6
047	ServicePartlyRefPbP	Сервіс оплата частинами часткове повернення	6
044	ServicePbPperiod	Сервіс оплата частинами в періоді	3
045	ServiceRefPbPperiod	Сервіс оплата частинами в періоді повернення	3
061	ServicePartlyRefPbPperiod	Сервіс оплата частинами в періоді часткове повернення	3
046	ServiceInstantPbI	Сервіс миттєва розстрочка Pay by installment	4
062	ServiceRefPbI	Сервіс миттєва розстрочка Pay by installment повернення	4
047	ServicePartlyRefPbI	Сервіс миттєва розстрочка Pay by installment часткове повернення	4
046	ServicePbIAct	Сервіс миттєва розстрочка акція	7
062	ServiceRefPbIAct	Сервіс миттєва розстрочка акція повернення	7
047	ServicePartlyRefPbIAct	Сервіс миттєва розстрочка акція часткове повернення	7
046	ServicePbPAct	Акційна ОЧ	8
062	ServiceRefPbPAct	Повернення акційної ОЧ	8
047	ServicePartlyRefPbPAct	Сервіс часткове повернення акційної ОЧ	8

5.15 Універсальний сервіс. Оскільки теоретично кількість сервісних операцій може бути великою(**J мерчанти**), немає можливості передбачити унікальний метод для кожної з них. Для вирішення цього завдання й існує універсальний сервіс. На відміну від унікальних сервісів, представлених вище, універсальний сервіс вимагає вказати не тільки номер мерчанта, а й номер сервісу, а також може приймати більш ніж один параметр, для чого їхній перелік необхідно вказати через slash (/).

5.15.1 Запит:

```
{
  "method": "ServiceGeneric",
  "step": 0,
  "params": {
    "amount": "0.01",
    "param": "6", // якщо їх більше 1, тоді записуємо так: 1/2/3/N
    "srvNum": "046",
    "merchantId": "6"
  }
}
```

5.15.2 Відповідь:

```
{
  "method" : "ServiceGeneric",
  "params" : {
    "adv" : "",
    "adv2p" : "",
    "amount" : "20.00",
    "approvalCode" : "230719",
    "bankAcquirer" : "ПриватБанк",
    "captureReference" : "",
    "cardExpiryDate" : "0323",
    "cardHolderName" : "VISA GOLD",
    "date" : "19.07.2023",
    "discount" : "0.00",
    "hstFld63Sf89" : " Text-of-63-89-field ",
    "invoiceNumber" : "000111",
  }
}
```

```
"issuerName" : "PRIVAT",
"merchant" : "X1K21UWR",
"pan" : "XXXXXXXXXXXX6873",
"paymentSystem" : "VISA",
"posConditionCode" : "00",
"posEntryMode" : "021",
"processingCode" : "000000",
"receipt" : "Text-of-receipt",
"responseCode" : "0000",
"agreementNum": "12345", // або rrn, або amountOfParts
"rrnExt" : "",
"signVerif" : "0",
"terminalId" : "X1K21UWR",
"time" : "15:29:59",
"track1" : "",
"trnStatus" : "1",
"txnType" : "0" },
"error" : false,
"errorDescription" : "",
"step" : 0 }
```

5.16 Cashback. Операція продажу з переведенням коштів у готівку. За своєю структурою фактично повторює логіку операції Purchase (продаж). Порівняно з **Purchase**, немає параметрів **signVerif**, **txnType**, **trnStatus**, **що повертаються**. Додається параметр **result** і **resultCashback**.

5.16.1 Запит

```
{
  "method": "Cashback",
  "step": 0,
  "params": {
    "amount": "0.30",
    "amountCash": "1.00",
    "merchantId": "1",
    "subMerchant": ""
  }
}
```

5.16.2 Відповідь

```
{
  "method": "Cashback",
  "step": 0,
  "params": {
    "amount": "1.30",
    "amountCash": "1.00",
    "approvalCode": "527080",
    "invoiceNumber": "139",
    "merchant": "1",
    "receipt": "text-of-receipt",
    "result": "Операція успішна",
    "resultCashback": "true", // false
    "rrn": "014721391803",
    "rrnExt": "88888888888888",
    "captureReference": "",
    "cardExpiryDate": "2020",
    "cardHolderName": "INSTANT/ISSUE",
    "date": "02.10.2019",
    "discount": "0.00",
    "hstFld63Sf89": "",
    "issuerName": "VISA ПРИВАТ",
    "merchant": "TSTTTTTT",
    "pan": "4731XXXXXXXXX9838",
    "posConditionCode": "00",
    "posEntryMode": "022",
    "processingCode": "000000",
    "responseCode": "0000",
    "terminalId": "TSTSALE2",
    "time": "09:11:07",
    "track1": "",
    "adv": "ПриватБанк",
    "adv2p": "Беремо і робимо!",
    "bankAcquirer": "ПриватБанк",
    "paymentSystem": "VISA",
    "subMerchant": ""
  }
}
```

```

    },
    "error": false,
    "errorDescription": ""
}

```

result: "Операція успішна " - параметр, що відповідає полю <Result></Result> аналогічного сценарію ВPOS у реалізації Ingenico.

resultCashback true/false - параметр на випадок часткового апруву операції, коли оплата відбулася, а кешбек - ні.

amountCash - сума видачі готівки

5.17 Audit. Він же X - balance

5.17.1 Запит

```

{
  "method": "Audit",
  "step": 0,
  "params": {
    "merchantId": "0"
  }
}

```

5.17.2 Відповідь

```

{
  "method": "Audit",
  "step": 0,
  "params": {
    "receipt": "08:50:00 03/10/2019 ===== [ X
БАЛАНС ] ===== ТЕРМ. TESTККМ0 В:000007 for slip:
0001 0067 Оператор MASTER ПРИВАТ бонуси/знижки 0.00 ГРН без
бонусів/знижок 1.01 ГРН з бонусами/знижками 1.01 ГРН VISA
ПРИВАТ бонуси/знижки 0.00 ГРН без бонусів/знижок 0.05 ГРН з
бонусами/знижками 0.05 ГРН ===== Загальні
підсумки: бонуси/знижки 0.00 ГРН без бонусів/знижок 1.06 ГРН з
бонусами/знижками 1.06 ГРН ===== Ingenico Group
TE7E v.131.403 15/03/ P9 ",
    "responseCode": "0000"
  },
  "error": false,

```

```
"errorDescription": ""
```

```
}
```

5.18 Звірка (Загальний звіт) Призначена для ініціювання й надсилання підсумкових сум на хост для звірки з даними, наявними на хості. Термінал може виконати звірку підсумків по черзі за всіма мерчантами, або за кожним окремо - залежно від наявності поля Merchant ID.

5.18.1 Запит

```
{
```

```
  "method": "Verify",
```

```
  "step": 0,
```

```
  "params": {
```

```
    "merchantId": "0"
```

```
  }
```

```
}
```

5.18.2 Відповідь (на прикладі BPOS Ingenico)

```
{
```

```
  "method": "Verify",
```

```
  "step": 0,
```

```
  "params": {
```

```
    "receipt": "07:53:00 02/10/2019 ===== [ Z
```

```
БАЛАНС ] ===== ТЕРМ. TESTKKM0 B:000006 for slip:
```

```
0001 0061 Оператор VISA ПРИВАТ бонуси/знижки 0.00 ГРН без
```

```
бонусів/знижок 0.03 ГРН з бонусами/знижками 0.03 ГРН
```

```
===== Загальні підсумки: бонуси/знижки 0.00 ГРН
```

```
без бонусів/знижок 0.03 ГРН з бонусами/знижками 0.03 ГРН
```

```
===== ПІДСУМКИ ВИЛУЧЕНІ
```

```
===== Ingenico Group TE7E v.131.403 15/03/ P9 ",
```

```
    "responseCode": "0000"
```

```
  },
```

```
  "error": false,
```

```
  "errorDescription": ""
```

```
}
```

5.18.3 Помилки

```
{
```

```
  "method": "Verify",
```

```
  "step": 0,
```

```
  "params": {
```

```
    "receipt": "",
```

```

        "responseCode": "0001"
    },
    "error": true,
    "errorDescription": "Cannot obtain receipt: log file is empty!"
}

```

5.19 Копія чека «Звірка» (Загальний звіт). Крім передання чека на касу, також проводиться друк чека терміналом.

5.19.1 Запит

```

{
    "method": "VerifyCopy",
    "step": 0,
    "params": {
        "merchantId": "0"
    }
}

```

5.19.2 Відповідь

```

{
    "method": "VerifyCopy",
    "step": 0,
    "params": {
        "receipt": "КОПІЯ 12:21:00 27/08/2019 ===== [ Z
БАЛАНС ] ===== ТЕРМ. TESTKKM0 B:000001 for slip:
0001 0011 Оператор VISA ПРИВАТ бонуси/знижки 0.01 ГРН без
бонусів/знижок 0.03 ГРН з бонусами/знижками 0.04 ГРН
===== Загальні підсумки: бонуси/знижки 0.01 ГРН
без бонусів/знижок 0.03 ГРН з бонусами/знижками 0.04 ГРН
===== КОПІЯ 07:53:00 02/10/2019
===== [Загальний звіт ] =====
ТЕРМ. TESTKKM0 B:000006 for slip: 0001 0061 Оператор VISA ПРИВАТ
бонуси/знижки 0.00 ГРН без бонусів/знижок 0.03 ГРН з
бонусами/знижками 0.03 ГРН ===== Загальні
підсумки: бонуси/знижки 0.00 ГРН без бонусів/знижок 0.03 ГРН з
бонусами/знижками 0.03 ГРН ===== Ingenico Group
TE7E v.131.403 15/03/ P9 ",
        "responseCode": "0000"
    }
}

```

```
},  
"error": false,  
"errorDescription": ""  
}
```

5.20 Отримання чека - PrintReceiptNum . Чи роздруковується чек терміналом у цьому випадку регулюється спеціальним прапором.

- prnFlag = 0 чек терміналом не роздруковується, видається на касу
- prnFlag = 1 друк чека терміналом, паралельно з видачею на касу
- invoiceNumber = "0" або "" - за замовчуванням береться номер останнього чека в пакеті
- Параметр "invoiceNumber", що відрізняється від "0" або "", є явною вказівкою номера чека, запитуваного касою. Тобто, якщо не передається номер чека вручну, то метод працює як друк останнього чека.

5.20.1 Запит

```
{  
  "method": "PrintReceiptNum",  
  "step": 0,  
  "params": {  
    "invoiceNumber": "123",  
    "prnFlag": "0"  
  }  
}
```

5.20.2 Відповідь

```
{  
  "method": "PrintReceiptNum",  
  "step": 0,  
  "params": {  
    "receipt": "ОПЛАТА СУМА 0.01 ГРН Підпис власника картки не  
потрібен MasterCard XXXXXXXXXXXXXXX9716 AID: A0000000041010  
ТЕРМІНАЛ TESTKKM0 03/10/2019 09:04:40 КОД АВТОРІЗАЦІЇ: 982088 ЧЕК  
N: 68 RRN: 014721392474 EXTERNAL RRN: Ingenico Group TE7E v.131.403  
15/03/P9 ",  
    "responseCode": "0000"  
  },  
  "error": false,  
  "errorDescription": ""  
}
```

```
}
```

5.20.3 Помилки

У разі помилки, на касу повертається параметр `responseCode` ≥ 1000 , порожній параметр `receipt` і опис помилки в `errorDescription`.

```
{  
  "method": "PrintReceiptNum",  
  "step": 0,  
  "params": {  
    "receipt": "",  
    "responseCode": "1000"  
  },  
  "error": true,  
  "errorDescription": "Cannot get receipt: Невірний мерчант!"  
}
```

Примітка: чеки мають накопичуватися в пам'яті термінала у вигляді якогось організованого сховища, чи то перелік спеціально пронумерованих файлів, чи то СУБД, чи то текстовий лог. Поняття **останній чек** - це є результат останньої операції виконаної терміналом у пакеті. Результат друку чека за номером, Z, X балансу або їхніх копій не замінює собою поняття останнього чека.

5.21 Друк інформації про всі транзакції в пакеті. Крім передання чека на касу, також проводиться друк чека терміналом.

5.21.1 Запит

```
{  
  "method": "PrintBatchJournal",  
  "step": 0,  
  "params": {  
    "merchantId": "0",  
    "prnFlag": "0"  
  }  
}
```

`merchantId` - індекс мерчанта, який буде використаний для цієї операції

Якщо `merchantId` = "0" або "", проводиться обробка даних за всіма доступними мерчантами

`prnFlag` - флаг друку чеку

`prnFlag` = 0 чек терміналом не роздруковується, видається на касу

`prnFlag` = 1 друк чека терміналом, паралельно з видачею на касу

`prnFlag` являється не обов'язковим параметром, якщо передавати пусте значення, або

даний параметр не передається то друкувати чек.

5.21.2 Відповідь

```
{
  "method": "PrintBatchJournal",
  "step": 0,
  "params": {
    "receipt": "text-of-receipt",
    "responseCode": "0000"
  },
  "error": false,
  "errorDescription": ""
}
```

5.21.3 Помилки

У разі помилки, на касу повертається параметр `responseCode` ≥ 1000 , порожній параметр `receipt` і опис помилки в `errorDescription`.

```
{
  "method": "PrintBatchJournal",
  "step": 0,
  "params": {
    "receipt": "",
    "responseCode": "1000"
  },
  "error": true,
  "errorDescription": "Cannot get receipt: Невірний мерчант!"
}
```

5.22 ReadBonusCard - читання бонусної картки із введенням її пінкоду, шифрування пінкоду 3DES ключем (у базі закріплюється ключ, переданий інтегратором).

5.22.1 Запит:

```
{
  "method": "ReadBonusCard",
  "step": 0,
```

```

    "prompt": {
      "line": [{
        "text": "qwerty", // якщо з каси не передається текст,
        за замовчуванням виводиться "Введіть бонусну картку"
        "font": "B",
        "align": "C"
      },
      {
        "text": "asdfg", // за замовчуванням "Магніт. або ЧІП"
        "font": "B",
        "align": "C"
      },
      {
        "text": "zxcvb", // за замовчуванням "Введіть PIN"
        "font": "B",
        "align": "C"
      }
    ]
  },
  "params": {
    "enterPIN": "1",
    "timeout": "6000"
  }
}

```

timeout - це час на операцію (читання картки / введення пін-коду), у мілісекундах. Не обов'язковий параметр. Значення за замовчуванням 60 сек.

enterPIN - ознака введення пін-коду для операції читання паливної картки. Не обов'язковий параметр. Дефолтне значення 0.

0, очікується тільки бонусна карта.

1, після читання карти очікується введення піна.

5.22.2 Відповідь:

```
{  
  "method": "ReadBonusCard",  
  "step": 0,  
  "params": {  
    "pan": "6769659308590307",  
    "pinblock" : "4C07BDD306150F2AB41A076CED87B50A",  
    "track2": "6769659308590307=25082061000076400000",  
    "result": "Операція успішна",  
    "responseCode": "0000"  
  },  
  "error": false,  
  "errorDescription": ""  
}
```

5.23 GetPinBonusCard - введення пінкоду бонусної картки, шифрування пінкоду 3DES ключем.

5.23.1 Запит:

```
{  
  "method": "GetPinBonusCard",  
  "step": 0,  
  
  "prompt": {  
    "line": [{  
  
      "text": "qwerty", // за замовчуванням "Введіть PIN"  
      "font": "B",  
      "align": "C"  
    }]  
  }  
}
```

```

        },
        {
            "text": "asdfg",
            "font": "B",
            "align": "C"
        },
        {
            "text": "zxcvb",
            "font": "B",
            "align": "C"
        }
    ]
},

"params": {
    "timeout": "6000"
}
}

```

5.23.2 Відповідь:

```

{
    "method": "GetPinBonusCard",
    "step": 0,
    "params": {
        "pinblock" : "4C07BDD306150F2AB41A076CED87B50A",
        "result": "Операція успішна",
        "responseCode": "0000"
    },
}

```

```
    "error": false,  
    "errorDescription": ""  
  }  
}
```

5.24 PinChangeBonusCard - зміна пінкоду бонусної картки, шифрування пінкоду 3DES ключем.

5.24.1 Запит:

```
{  
  "method": "PinChangeBonusCard",  
  "step": 0,  
  
  "prompt": {  
    "line": [{  
      "text": "qwerty", // за замовчуванням "Введіть PIN"  
      "font": "B",  
      "align": "C"  
    },  
    {  
      "text": "asdfg", // за замовчуванням "Введіть нов.  
PIN"  
      "font": "B",  
      "align": "C"  
    },  
    {  
      "text": "zxcvb", // за замовчуванням "Підтвердіть PIN"  
      "font": "B",  
      "align": "C"  
    }  
  ]  
}
```

```

    }
  ]
},

"params": {
  "timeout": "6000",
  "attempt": "3"
}
}

```

timeout - це час на операцію (читання картки / введення пін-коду), у мілісекундах. Не обов'язковий параметр. Значення за замовчуванням 60 сек.

attempt - кількість повторень нового пін-коду. Не обов'язковий параметр. Дефолтне значення 1. Якщо нові пін-коди не збігатимуться більше attempt разів, то помилка "errorDescription": "Нові пін-коди не збігаються".

5.24.2 Відповідь

```

{
  "method": "PinChangeBonusCard",
  "step": 0,
  "params": {
    "pinblock" : "4C07BDD306150F2AB41A076CED87B50A",
    "pinblockNew": "5D7F083C42F2750B82A6AB243DAB14C2"
    "result": "Операція успішна",
    "responseCode": "0000"
  },
  "error": false,
  "errorDescription": ""
}

```

//=====

font - шрифт, яким буде відображено текст на екрані

- "N" – Normal

- "B" – Bold

- "S" - Small

align - вирівнювання по горизонталі

- "C" - центрувати

- "L" - вирівняти по лівому краю

- "R" - вирівняти по правому краю

//=====

об'єкт, що відповідає запиту з полем "prompt" (Golang)

```
type jsonEntityWithPrompt struct {
    Method string `json:"method"`
    Params struct {
        EnterPIN string `json:"enterPIN"`
        Timeout string `json:"timeout"`
    } `json:"params"`
    Prompt struct {
        Line []struct {
            Align string `json:"align"`
            Font string `json:"font"`
            Text string `json:"text"`
        } `json:"line"`
    } `json:"prompt"`
    Step int64 `json:"step"`
}
```

5.25 GetPhoneNumber (Введення номеру телефону)

5.25.1 Запит:

```
{
  "method": "GetPhoneNumber",
  "step": 0,
  "params": {
    "deviceType": "Terminal"
  }
}
```

"deviceType" - Тип пристрою, який використовується для введення номеру телефону. Можливі значення - "Terminal"/"PinPad". За замовчуванням (якщо тип пристрою не

вказано) використовується Terminal.

5.25.2 Відповідь:

```
{
  "method": "GetPhoneNumber",
  "step": 0,
  "params": {
    "PhoneNumber": "1234567890",
    "responseCode": "0000"
  },
  "error": false,
  "errorDescription": ""
}
```

5.25.3 Помилка:

При відмові вводу номеру телефону.

```
{
  "method": "GetPhoneNumber",
  "step": 0,
  "params": {
    "responseCode": "1001"
  },
  "error": true,
  "errorDescription": "Operation was canceled"
}
```

5.26 Preauthorization (Предавторизація)

5.26.1 Запит:

```
{
  "method": "Preauthorization",
  "step": 0,
  "params": {
    "amount": "500.00",
    "merchantId": "1"
  }
}
```

5.26.2 Відповідь:


```
{
  "method": "Preauthorization",
  "step": 0,
  "params": {
    "amount": "0.60",
    "approvalCode": "999999",
    "captureReference": "",
    "cardExpiryDate": "2020",
    "cardHolderName": "INSTANT/ISSUE",
    "date": "02.10.2019",
    "discount": "0.00",
    "hstFld63Sf89": "",
    "invoiceNumber": "999999",
    "issuerName": "VISA ПРИВАТ",
    "merchant": "TSTTTTTT",
    "pan": "4731XXXXXXXXX9838",
    "posConditionCode": "00",
    "posEntryMode": "022",
    "processingCode": "000000",
    "receipt": "text-of-receipt",
    "responseCode": "0000",
    "rrn": "99999999999999",
    "rrnExt": "88888888888888",
    "terminalId": "TSTSALE2",
    "time": "09:11:07",
    "track1": "",
    "signVerif": "0",
    "txnType": "1",
    "trnStatus": "1",
    "adv": "ПриватБанк",
    "adv2p": "Беремо і робимо!",
    "bankAcquirer": "ПриватБанк",
    "paymentSystem": "VISA",
    "subMerchant": ""
  },
  "error": false,
  "errorDescription": ""
}
```

```
}
```

5.27 SaleCompletion (Завершення предавторизації)

5.27.1 Запит:

```
{
```

```
"method": "SaleCompletion",
```

```
"step": 0,
```

```
"params": {
```

```
    "amount": "500.00",
```

```
    "addamount": "500.00",
```

```
    "approvalCode": "999999",
```

```
    "rrn": "99999999999999"
```

```
}
```

```
}
```

amount - Сума преавторизації.

addamount - Остаточна сума операції.

approvalCode - Код авторизації відповідної преавторизації.

rrn - Унікальний номер транзакції при оформленні преавторизації, не обов'язкове поле.

Після відправки запиту термінал запрошує картку. Після зчитування картки на терміналах Ingenico і Verifone ввести RRN на терміналі, якщо його не було передано з каси, на РАХ вибрати потрібну операцію для завершення, на екрані терміналу.

5.27.2 Відповідь:

```
{
```

```
"method": "SaleCompletion",
```

```
"step": 0,
```

```
"params": {
```

```
    "amount": "500.00",
```

```
    "approvalCode": "999999",
```

```
    "captureReference": "",
```

```
    "cardExpiryDate": "2020",
```

```
    "cardHolderName": "INSTANT/ISSUE",
```

```
    "date": "02.10.2019",
```

```
    "discount": "0.00",
```

```
    "hstFld63Sf89": "",
```

```
    "invoiceNumber": "999999",
```

```
    "issuerName": "VISA ПРИВАТ",
```

```

    "merchant": "TSTTTTTT",
    "pan": "4731XXXXXXXXX9838",
    "posConditionCode": "00",
    "posEntryMode": "022",
    "processingCode": "000000",
    "receipt": "text-of-receipt",
    "responseCode": "0000",
    "rrn": "9999999999999",
    "rrnExt": "8888888888888",
    "terminalId": "TSTSALE2",
    "time": "09:11:07",
    "track1": "",
    "signVerif": "0",
    "txnType": "1",
    "trnStatus": "1",
    "adv": "ПриватБанк",
    "adv2p": "Беремо і робимо!",
    "bankAcquirer": "ПриватБанк",
    "paymentSystem": "VISA",
    "subMerchant": ""
  },
  "error": false,
  "errorDescription": ""
}

```

amount - У відповіді на завершення, це остаточна сума преавторизації.

discount - Розмір знижки.

5.28 Операція Partial approval

Операція Часткового схвалення (Partial approval).

Якщо у відповіді приходить успішне проведення Partial approval (RC =10), то терминал передає на касу та друкує чек про частковому схваленні оплати.

Клиент приймає рішення про подальші дії.

Можливі варіанти:

5.28.1. Покупка в межах суми Partial approval, або доплата готівкою;

5.28.2. Коригування (зменшення) суми оплати;

5.28.3 Доплата іншою карткою;

5.28.4 Скасування оплати.

Відправляємо запит на продаж, метод Purchase:

```
{
  "method": "Purchase",
  "step": 0,
  "params": {
    "amount": "2000.00",
    "discount": "",
    "merchantId": "0",
    "facepay": "false", // true
    "subMerchant": ""
  }
}
```

У відповідь отримуємо часткове схвалення ("responseCode": "0010"):

```
{
  "method": "Purchase",
  "step": 0,
  "params": {
    "amount": "1500.00",
    "approvalCode": "999999",
    "captureReference": "",
    "cardExpiryDate": "2020",
    "cardHolderName": "INSTANT/ISSUE",
    "date": "02.10.2023",
    "discount": "0.00",
    "hstFld63Sf89": "",
    "invoiceNumber": "999999",
    "issuerName": "VISA ПРИВАТ",
    "merchant": "TSTTTTTT",
    "pan": "4731XXXXXXXXXX9838",
    "posConditionCode": "00",

```

```
"posEntryMode": "022",
"processingCode": "000000",
"receipt": "text-of-receipt",
"responseCode": "0010",
"rrn": "9999999999999",
"rrnExt": "8888888888888",
"terminalId": "TSTSALE2",
"time": "09:11:07",
"track1": "",
"signVerif": "0",
"txnType": "1",
"trnStatus": "1",
"adv": "ПриватБанк",
"adv2p": "Беремо і робимо!",
"bankAcquirer": "ПриватБанк",
"paymentSystem": "VISA",
"subMerchant": ""
},
"error": false,
"errorDescription": ""
}
```

Після обрання клієнтом бажаного завершення, виконуємо наступні дії:

5.28.1. Покупка в межах суми Partial approval, або доплата готівкою.

Термінал видав чек з успішним схваленням часткової оплати, при даному сценарії дії терміналу завершені.

5.28.2 Коригування (зменшення) суми оплати.

В даному випадку потрібно виконати часткове скасування **WithdrawalPartly**.

5.28.2.1 Запит на скасування:

```
{
  "method": "WithdrawalPartly",
  "step": 0,
  "params": {
```

```
        "amount": "1000.0",
        "invoiceNumber": "131220"
    }
}
```

5.28.2.2 Відповідь

Результат відповіді такий самий, як і в п.5.1.2, за винятком поля **signVerif**, **txnType**

```
{
    "method": "WithdrawalPartly",
    "step": 0,
    "params": {
        "amount": "1000.0",
        "approvalCode": "999999",
        "captureReference": "",
        "cardExpiryDate": "2020",
        "cardHolderName": "INSTANT/ISSUE",
        "date": "09:11:07",
        "discount": "0.00",
        "hstFld63Sf89": "",
        "invoiceNumber": "999999",
        "issuerName": "VISA ПРИВАТ",
        "merchant": "TSTTTTTT",
        "pan": "4731XXXXXXXXXX9838",
        "posConditionCode": "00",
        "posEntryMode": "022",
        "processingCode": "000000",
        "receipt": "text-of-receipt",
        "responseCode": "0000",
        "rrn": "99999999999999",
        "rrnExt": "88888888888888",
        "terminalId": "TSTSALE2",
        "time": "09:11:07",
        "track1": "",
        "trnStatus": "1",
        "adv": "ПриватБанк",
        "adv2p": "Беремо і робимо!",
        "bankAcquirer": "ПриватБанк",
        "paymentSystem": "VISA",
        "subMerchant": ""
    }
}
```

```
},  
"error": false,  
"errorDescription": ""  
}
```

5.28.3 Доплата іншою карткою;

Виконуємо продаж на суму якої не вистачило при першій оплаті.
Відправляємо запит:

```
{  
  "method": "Purchase",  
  "step": 0,  
  "params": {  
    "amount": "500.00",  
    "discount": "",  
    "merchantId": "0",  
    "facepay": "false", // true  
    "subMerchant": ""  
  }  
}
```

5.28.4 Скасування оплати.

Виконує скасування оплати що пройшла по частковому схваленню за допомогою методу Withdrawal.

Запит на скасування:

```
{  
  "method": "Withdrawal",  
  "step": 0,  
  "params": {  
    "invoiceNumber": "131220"  
  }  
}
```

```
}
```

5.29 GetOTPPassword (Введення OTP паролю)

5.29.1 Запит:

```
{  
    "method": "GetOTPPassword",  
    "step": 0,  
    "params": {  
        "timeout": "6000",  
        "deviceType": "Terminal"  
    }  
}
```

deviceType - Тип пристрою, який використовується для введення номеру телефону. Можливі значення - "Terminal"/"PinPad". Якщо тип пристрою не вказано то тип пристрою визначається автоматично.

timeout - це час на операцію, у мілісекундах. Не обов'язковий параметр. Значення за замовчуванням 60 сек.

На екран виводити текст "Введіть OTP".

5.29.2 Відповідь:

```
{  
    "method": "GetOTPPassword",  
    "step": 0,  
    "params": {  
        "OTPPassword": "12345678",  
        "responseCode": "0000"  
    },  
    "error": false,  
    "errorDescription": ""  
}
```

Значення OTPpassword передавати у не зашифрованому виді.

5.29.3 Помилка:

При відмові вводу ОТП паролю.

```
{
  "method": "GetOTPPassword",
  "step": 0,
  "params": {
    "responseCode": "1001"
  },
  "error": true,
  "errorDescription": "Operation was canceled"
}
```

5.30 GetReceiptInfo (Отримання даних по чеку)

Отримання інформації по номеру чеку. Друку чеку не відбувається, інформація лише передається на касу.

5.30.1 Запит

```
{
  "method": "GetReceiptInfo",
  "step": 0,
  "params": {
    "invoiceNumber": "123"
  }
}
```

Якщо invoiceNumber = "0" або "" - за замовчуванням береться номер останнього чека в пакеті.

5.30.2 Відповідь

```
{
  "method": "GetReceiptInfo",
  "step": 0,
  "params": {
    "amount": "0.60",
    "approvalCode": "999999",
    "date": "02.10.2019",
    "invoiceNumber": "999999",
  }
}
```

```

        "issuerName": "VISA ПРИВАТ",
        "merchant": "TSTTTTTT",
        "pan": "4731XXXXXXXXX9838",
        "receipt": "text-of-receipt",
        "rrn": "9999999999999",
        "time": "09:11:07",
        "bankAcquirer": "ПриватБанк",
        "paymentSystem": "VISA",
        "signVerif": "0",
        "txnType": "1",
        "responseCode": "0000"
    },
    "error": false,
    "errorDescription": ""
}

```

responseCode - код відповіді термінала, на операцію з чека

approvalCode - код авторизації

amount - сума проведеної транзакції

pan - номер картки з урахуванням контрольної цифри

date - дата транзакції

time - час транзакції

invoiceNumber - номер чека підтвердженої транзакції

issuerName - ім'я емітента, за яким проводилася операція

merchant - номер мерчанта, за яким проводилася операція

receipt - чек

rrn - унікальний номер транзакції

signVerif - перевірка підпису (CVM)

txnType - тип транзакції

bankAcquirer - найменування банку-еквайєра

paymentSystem - найменування МПС, що передається з хоста

TxnType has one of following values:

0 – undefined

1 – Purchase

2 – Refund

3 – Withdrawal

4 - операція ОЧ або МР

Значення **error** та **errorDescription** приймають значення по результатам відпрацювання саме методу **GetReceiptInfo**

Якщо операція по якій треба повернути інформацію була не успішна, то повертати всі поля що описані протоколом, якщо їх значення немає то передавати пусте

значення параметру.

5.30.3 Помилки

У разі помилки, на касу повертається параметр `responseCode` ≥ 1000 і опис помилки в `errorDescription`.

```
{  
  "method": "GetReceiptInfo",  
  "step": 0,  
  "params": {  
    "responseCode": "1000"  
  },  
  "error": true,  
  "errorDescription": "Cannot get receipt: Невірний мерчант!"  
}
```

6. Службові повідомлення

Service Messages - особливий тип повідомлень, які обробляються сервером асинхронно, тобто незалежно від поточної операції або стану. Використовуються для передавання службових команд від клієнта серверу і, навпаки, для отримання клієнтом повідомлень про стан від сервера.

6.1 Повідомлення deviceBusy - це повідомлення сервер надсилає клієнту у відповідь на будь-яке несервісне повідомлення в тому разі, якщо сервер зайнятий поточною операцією і не може прийняти наступний запит. Відповідь на це повідомлення не потрібна. Генерується тільки терміналом.

```
{
  "method": "ServiceMessage",
  "step": 0,
  "params": {
    "msgType": "deviceBusy"
  },
  "error": false,
  "errorDescription": ""
}
```

6.2 Повідомлення interrupt - вимога переривання ініційованої операції.

```
{
  "method": "ServiceMessage",
  "step": 0,
  "params": {
    "msgType": "interrupt"
  }
}
```

6.2.1 Відповідь на повідомлення interrupt - підтвердження interruptTransmitted. Генерується тільки терміналом.

```
{
  "method": "ServiceMessage",
```

```
"step": 0,  
"params": {  
    "msgType": " interruptTransmitted"  
},  
"error": false,  
"errorDescription": ""  
}
```

Також термінал повинен повернути відповідь на активну транзакцію, під час виконання методу interrupt, з responseCode 1001.

Приклад:

```
{ "error" : true, "errorDescription" : "Операція скасов.", "method" : "Purchase", "params" : {  
"responseCode" : "1001" }, "step" : 0 }
```

Якщо метод interrupt був виконаний під час авторизації, чи до відповіді по операції на касу, то термінал повинен виконати скасування операції.

6.3 Повідомлення methodNotImplemented - повідомлення генерується терміналом у відповідь на запит не підтримуваного методу.

```
{  
"method": "ServiceMessage",  
"step": 0,  
"params": {  
    "msgType": " methodNotImplemented"  
},  
"error": false,  
"errorDescription": ""  
}
```

6.4 Повідомлення getMerchantList - запит клієнтом у сервера списку назв торговців з їхніми порядковими номерами.

6.4.1 Запит:

```
{
```

```
"method": "ServiceMessage",
"step": 0,
"params": {
    "msgType": "getMerchantList"
}
}
```

6.4.2 Відповідь на це повідомлення:

```
{
"method": "ServiceMessage",
"step": 0,
"params": {
    "3": "Оплата Частинами в періоді",
    "4": "Оплата частинами",
    "5": "Миттєва Розстрочка",
    "6": "Миттєва Розстрочка Акція",
    "12": "Табелювання",
    "msgType": "getMerchantList"
},
"error": false,
"errorDescription": ""
}
```

6.5 Повідомлення getMaskList - запит клієнтом у сервера списку масок мерчантів з їхніми порядковими номерами.

6.5.1 Запит:

```
{
"method": "ServiceMessage",
"step": 0,
"params": {
    "msgType": " getMaskList"
}
}
```

6.5.2 Відповідь на це повідомлення:

```
{
```

```

"method": "ServiceMessage",
"step": 0,
"params": {
    "1": "TESTKKM0",
    "2": "TESTKKM1",
    "3": "X1110B6L",
    "4": "X11113W2",
    "5": "REFUND",
    "6": "X11193W2",
    "7": "X10D2QLB",
    "8": "X11183W2",
    "9": "TESTKKM2",
    "10": "INKASS1",
    "11": "INKASS2",
    "12": "S1KITV99",
    "msgType": "getMaskList"
},
"error": false,
"errorDescription": ""
}

```

6.6 Сервісне повідомлення debug - використовується для вимкнення/ввімкнення функції безпосереднього зняття логів портів COM, USB, ETH. Файл, у який здійснюється запис лога, іменується trace.log і створюється в каталозі, поточному стосовно ПЗ, що ініціювало логування. **Цей функціонал повністю реалізується на стороні клієнта (запит, відповідь, реалізація запису в файл).**

Формат лога:

[17/10/2019 15:27:43] Written data (COM9)

```

7b 22 6d 65 74 68 6f 64 22 3a 22 50 69 6e 67 44
65 76 69 63 65 22 2c 22 73 74 65 70 22 3a 30 7d
00

```

```

{"method":"PingDevice","step":0}.

```

[17/10/2019 15:27:43] Read data (COM9)

```

7b 22 6d 65 74 68 6f 64 22 3a 22 50 69 6e 67 44
65 76 69 63 65 22 2c 22 73 74 65 70 22 3a 30 2c
22 70 61 72 61 6d 73 22 3a 7b 22 63 6f 64 65 22
3a 22 30 30 22 2c 22 72 65 73 70 6f 6e 73 65 43
6f 64 65 22 3a 22 30 30 30 22 7d 2c 22 65 72
72 6f 72 22 3a 66 61 6c 73 65 2c 22 65 72 72 6f
72 44 65 73 63 72 69 74 70 74 69 6f 6e 22 3a 22
22 7d 00

```

```

{"method":"PingDevice",
"step":0,
"params":{"code":"00",
"responseCode":"0000"},
"error":false,
"errorDescription":""}.

```

[17/10/2019 15:27:44] - Close port COM9

6.6.1 Запит:

```
{
  "method": "ServiceMessage",
  "step": 0,
  "params": {
    "msgType": "debug"
  }
}
```

6.6.2 Відповідь на це повідомлення:

```
{
  "method": "ServiceMessage",
  "step": 0,
  "params": {
    "msgType": "debugOn" // або debugOff
  },
  "error": false,
  "errorDescription": ""
}
```

6.7 Service message getLastResult

Параметр **LastResult** використовується для отримання результату виконаної транзакції
Перелік можливих значень відповідає аналогічному property BPOS

get_LastResult(BYTE pVal), property LastResult*

Gets the result of fulfilling of the last transaction

[OUT] pVal- Returned values:

0 – successfully fulfilled

2 – in progress

```
{
  "method": "ServiceMessage",
  "step": 0,
  "params": {
    "msgType": "getLastResult"
  }
}
```


6.7.1 getLastResult відповідь

```
{
  "method": "ServiceMessage",
  "step": 0,
  "params": {
    "msgType": "getLastResult",
    "LastResult": "0"
  },
  "error": false,
  "errorDescription": ""
}
```

6.8 Service message getLastStatMsgCode

Параметр LastStatMsgCode вказує поточний хід виконання операції терміналом. Зазвичай цей параметр змінюється паралельно зміні подій в інтерфейсі пристрою під час взаємодії з касиром/користувачем. Перелік можливих значень відповідає аналогічному property BPOS.

get_LastStatMsgCode(BYTE pVal), property LastStatMsgCode*

[OUT] pVal - returns one following status codes:

0 – status code is not available.

1 – card was read

2 – used a chip card

3 – authorization in progress

4 – waiting for cashier action

5 – printing receipt

6 – pin entry is needed

7 – card was removed

8 – EMV multi aid's

9 – waiting for card

10 – in progress

11 – correct transaction

```
{
  "method": "ServiceMessage",
  "step": 0,
  "params": {
    "msgType": "getLastStatMsgCode"
  }
}
```

```
}
```

6.8.1 getLastStatMsgCode відповідь

```
{  
  "method": "ServiceMessage",  
  "step": 0,  
  "params": {  
    "msgType": "getLastStatMsgCode",  
    "LastStatMsgCode": "6"  
  },  
  "error": false,  
  "errorDescription": ""  
}
```

6.9 Service message getLastStatMsgDescription

Параметр LastStatMsgDescription вказує поточний хід виконання операції терміналом. Зазвичай цей параметр змінюється паралельно зміні подій в інтерфейсі пристрою під час взаємодії з касиром/користувачем. Перелік можливих значень відповідає аналогічному property BPOS і є текстовою розшифровкою значень параметра LastStatMsgCode.

get_LastStatMsgCode(BYTE pVal), property LastStatMsgCode*

[OUT] pVal - returns one following status codes:

0 – status code is not available.

1 – card was read

2 – used a chip card

3 – authorization in progress

4 – waiting for cashier action

5 – printing receipt

6 – pin entry is needed

7 – card was removed

8 – EMV multi aid's

9 – waiting for card

10 – in progress

11 – correct transaction

```
{  
  "method": "ServiceMessage",  
  "step": 0,
```

```
"params": {  
    "msgType": "getLastStatMsgDescription"  
}  
}
```

6.9.1 getLastStatMsgDescription відповідь

```
{  
    "method": "ServiceMessage",  
    "step": 0,  
    "params": {  
        "msgType": "getLastStatMsgDescription",  
        "LastStatMsgDescription": "pin entry is needed"  
    },  
    "error": false,  
    "errorDescription": ""  
}
```

6.10 Identify Ідентифікація приналежності даного термінала АТ КБ ПриватБанк, а також інформація про виробника й модель даного термінала.

6.10.1 Запит

```
{  
    "method": "ServiceMessage",  
    "step": 0,  
    "params": {  
        "msgType": "identify"  
    }  
}
```

6.10.2 відповідь

```
{  
    "method": "ServiceMessage",  
    "step": 0,  
    "params": {  
        "msgType": "identify",  
        "result": "true",  
        "vendor": "PAX",
```

```

        "model": "s800"
    },
    "error": false,
    "errorDescription": ""
}

```

6.11 Service message getDiscountName використовується в межах процедури коригування транзакції. Викликається в разі getLastStatMsgCode = 11.

6.11.1 Запит

```

{
    "method": "ServiceMessage",
    "step": 0,
    "params": {
        "msgType": "getDiscountName"
    }
}

```

6.11.2 Відповідь

У разі виклику методів поза передбаченим контекстом або відсутності параметрів передавати такі (default) значення:

- **getDiscountName** повертається значення "discountName": "NO_D_GROUP"
- **pan** повертається значення "pan": "PAN_UNDEF"
- **posEntryMode** повертається значення "posEntryMode": "MODE_UNDEF"

```

{
    "method": "ServiceMessage",
    "step": 0,
    "params": {
        "msgType": "getDiscountName",
        "discountName": "someDiscountGroup",
        "pan": "123456*****9876",
        "posEntryMode": "071"
    },
    "error": false,
    "errorDescription": ""
}

```

Правила маскування PAN:

Для терміналів PAX:

- передавати цифри номеру картки згідно параметру mask_pan, інші масковані зірочками (*)
- якщо параметр mask_pan = false - необхідно маскувати PAN згідно з вимогами платіжних систем (за стандартом)

Для терміналів Ingenico, Verifone:

- під час додавання мерчанта до винятків і підключення відповідних комісій - передаються цифри номера картки згідно налаштувань комісії, інші масковані зірочками (*)
- для решти терміналів мережі - необхідно маскувати PAN згідно з вимогами платіжних систем (за стандартом)

6.12 Service message correctTransaction - коригування транзакції в межах операції Purchase.

```
{  
  "method": "ServiceMessage",  
  "step": 0,  
  "params": {  
    "msgType": "correctTransaction",  
    "amount": "0.02",  
    "discount": "0.01"  
  }  
}
```

Помилки:

У разі некоректного встановлення значень (not a number, delimiter != , | | . etc.) корекція не застосовується, "error": true, "errorDescription": "Корекція неможлива"

Purchase продовжує виконання з вихідною сумою

Решта помилок приходять в основній відповіді хоста на Purchase

6.12.1 Service message correctionTransmitted - повідомлення-відповідь у разі успішного застосування коригування транзакції.

```
{  
  "method": "ServiceMessage",  
  "step": 0,  
  "params": {  
    "msgType": "correctionTransmitted"  
  },  
  "error": false,
```

```
"errorDescription": ""  
}
```

Важливо. Correct Transaction - general workflow

1. Виклик базового Purchase
2. Виклик сервісного повідомлення getLastStatMsgCode if 11 ->
3. Отримуємо ім'я групи дисконту або PAN за допомогою getDiscountName.
4. Каса аналізує групу дисконту або PAN.
5. Проводиться корекція транзакції з урахуванням обраної знижки шляхом виклику correctTransaction (карта, PIN, etc. повторно НЕ запитуються)
6. У разі успішного передавання корекції, у межах встановленого Adjust Timeout каса приймає serviceMessage correctionTransmitted
7. В іншому випадку, Purchase продовжує виконуватися в рамках вихідних (початкових) даних.

Патчноут опису протоколу

v.1.0.0.1-1.0.0.3 - виправлення дрібних неточностей

v.1.0.0.4 - уточнення, як значення того чи іншого поля JSON усі не-ASCII символи, у т. ч.

кирилиця, передаються в кодуванні utf-8

v.1.0.0.5 - додано опис властивостей провідколу BPOS наведених як приклади

(get_LastResult, get_LastErrorCode та ін.)

v.1.0.0.6 - прибрано методи getLastErrorCode і getLastErrorDescription, додано уточнення

щодо id мерчантів за сервісними методами

v.1.0.0.7 - у сервісі "ServicePbP" додано поле "amountOfParts" у відповіді термінала;

- уточнення формату даних, що приймається, у полі "amount" (п.6.1.)

v.1.0.0.8 - додано методи CashAdvance і PrintHTML (п.5.22, 5.23)

v.1.0.0.9 - уточнення щодо методів "ServiceRefund", "ServicePbP", за іменними сервісами нема потреби передавати поле "merchantId", оскільки ці дані зашиті в методах хардкодом (див. Таблиця 1.1)

v.1.0.1.0 - додано опис особливостей genericDriverJsonUSB_Linux

v.1.0.1.1 - прибрано передачу track2 під час проведення всіх операцій, крім методів **ReadCardBank** і **ReadCardDiscount**

v.1.0.1.2 - у методі "**Purchase**" додано поле "**facepay**" для запуску оплати через FacePay24 з каси

- у методі "**CashAdvance**" прибрані поля "currencyCode" і "accountNumber"

v.1.0.1.3 - уточнення щодо роботи з дійсними числами в методі "**Purchase**"

v.1.0.1.4 - для фінансових операцій додано поля "**adv**", "**adv2p**" і "**bankacquirer**"

v.1.0.1.5 - у п.2 додано покроковий опис проведення хендшейку з боку касового ПЗ, доповнено опис канального протоколу

v.1.0.1.6 - виправлена помилка в полі "**errorDescription**", що повертається терміналом

- виправлено запит методу "**GetBalance**"
- додано методи "**getDiscountName**" (п.6.10) і "**correctTransaction**" (п.6.11)

v.1.0.1.7 - розширення методу correctTransaction

- виправлено помилку "bankaquirer" на "bankAcquirer"
- додано поле у відповіді термінала "paymentSystem"

v.1.0.1.8 - додавання поля "rrnExt"

- зміна правил передачі track2 у методі ReadCardBank
- При responseCode відмінних від 0000 термінал передає "error":true

v.1.0.1.9 - додано поле у відповіді термінала за фінансовими операціями "panHash"

v.1.0.2.0

- додано поле у відповіді термінала за фінансовими операціями "subMerchant"

v.1.0.2.1

- прибрали поле "panHash"
- додано поле "subMerchant" у запиті за фін. операціями
- z-звіт перейменовано на Загальний звіт

v.1.0.2.2

- додано поле "textMess" у відповіді термінала за фін. операціями
- додано методи ServicePbPAct і ServiceRefPbPAct для Акційної ОЧ

v.1.0.2.3

- прибрали поле "textMess" у відповіді термінала за фін. операціями
- додано методи ReadBonusCard, GetPinBonusCard, PinChangeBonusCard

v.1.0.2.4

- додано поля у відповіді термінала за методом "ServiceGeneric"

v.1.0.2.5

- додано методи **GetPhoneNumber**, Preauthorization, **SaleCompleteon**
- додано поле track2 у відповіді термінала по методу ReadBonusCard

v.1.0.2.6

- додано операцію **Partial approval**
- додано поле у відповіді термінала по методам **Preauthorization**, **SaleCompleteon**

v.1.0.2.7

- розширено опис сервісного повідомлення **interruptTransmitted**
- у методі **PrintBatchJournal** додано параметр **prnFlag**
- до списку хардкодних сервісів додано сервіс часткового повернення акційної ОЧ **ServicePartlyRefPbPAct**
- змінені правила маскування pan у відповіді на сервісне повідомлення **getDiscountName**

v.1.0.2.8

- у п.4 додано опис коли при responseCode не 0000 передавати "error":false", винятком є responseCode 0010.

v.1.0.2.9

- додано метод **GetOTPpassword** (Введення OTP пароллю)

v.1.0.3.0

- змінено метод **GetOTPpassword** (Введення OTP пароллю)

v.1.0.3.1

- додано метод **GetReceiptInfo** (Отримання даних по чеку)
- змінено метод **SaleCompleteon**. Виправлена орфографічна помилка **SaleCompleti**on та у запиті на завершення додано поле RRN.