



Кассовый сервер QKkmServer

Описание протокола управления

версия от 07.06.19

Оглавление

Введение.....	3
Формат команды протокола.....	3
Формат ответа протокола.....	3
Проверка успешности печати чека.....	4
Алгоритм печати чека с реакцией на ошибки.....	5
Транспорты.....	6
FileAPI.....	6
FtpAPI.....	6
WebAPI.....	6
Использование Токена сессии.....	7
Скрипт запроса задания.....	7
Скрипт получения ответа.....	8
Усиленное подтверждение ответа.....	8
AjaxAPI.....	9
Команды протокола.....	10
Выбрать кассу.....	10
Открыть новую смену.....	10
Печать строки текста.....	10
Печать множества строк текста (SLIP-чек).....	10
Печать строки заданным шрифтом.....	11
Печать строки жирным шрифтом.....	11
Отчет без гашения (X-отчет).....	11
Отчет с гашением (Z-отчет).....	11
Отчет по секциям ККМ.....	11
Открыть чек.....	11
Добавить товар в чек.....	12
Аннулирование чека.....	13
Продолжить печать.....	13
Передать TLV-реквизит в ККМ.....	13

Оплатить чек (короткая форма) Не соответствует ФФД 1.1.....	14
Оплатить чек (длинная форма) Не соответствует ФФД 1.1.....	14
Оплатить чек Соответствует ФФД 1.1.....	15
Внесение денег в кассу.....	15
Выемка денег из кассы (инкассация).....	16
Протяжка ленты.....	16
Отрезка чека.....	16
Запрос денежного регистра.....	16
Печать копии последнего чека.....	17
Открыть денежный ящик.....	17
Чек коррекции на сумму.....	17
Запрос фискального признака и фискального номера чека.....	17
Получить номер чека и номер смены.....	18
Печать заголовка чека.....	18
Печать рекламного заголовка.....	19
Печать изображения из памяти ККМ.....	19
Запрос полного статуса ККМ.....	19
Примеры.....	22
Наиболее частые ошибки.....	23
Общие комментарии.....	23
Контактные данные:.....	23

Введение

1. Протокол управления кассовым сервером представляет собой текстовый файл, сформированный в соответствии со стандартом CSV – поля разделяются символом «точка с запятой».
2. Перенос строк осуществляется в UNIX-Like формате ('/n')
3. Кодировка строковых данных – UTF8 (Unicode)
4. Одна строка представляет собой, как правило, одну команду для ККМ.
5. В файле может присутствовать любое количество строк-команд.
6. Внутри текстовой строки не допускается использование специальных символов, символов переноса строки, символа «точка с запятой», двойных кавычек. Использование данных символов приведет к ошибкам в разборе команды парсером и нарушению логики работы ККМ.
Рекомендуемый набор символов: [a-яA-ЯA-za-z01234567890-+=@#%?._,|](#)
Использование любых других символов не рекомендуется. Лучше их удалять или заменять на пробел (иной подходящий по смыслу символ).

Формат команды протокола

Команда представляет собой CSV строку, первым параметром в которой всегда идет наименование команды. Во всех последующих позициях располагаются параметры.

1. Наименование_команды
2. Если необходимо – параметры.

Примеры команд:

```
x;  
b;0;ФИО Кассира;0;
```

Формат ответа протокола

После исполнения команды Кассовый сервер формирует ответ. Ответ состоит из набора CSV-строк, организованных следующим образом:

Наименование_команды;код_ответа;текст_ответа;ответ_ядра;

1. Наименование_команды – то же самое, что и в команде-запросе.
2. Код_ответа – число.
 - a. 0 -- Отсутствие ошибки ;
 - b. 255 -- отсутствие связи с ККМ;
 - c. Остальные коды – в соответствии с моделью. У АТОЛа значения кода имеют отрицательное значение. У Штрих-М – положительное.
3. Текст_ответа – строка. Текстовое описание ошибки. Можно использовать для отображения пользователю.

4. Ответ_ядра – строка. Ответ сервера в формате XML кодируется по алгоритму BASE64. Может понадобиться для получения некоторых значений возвращаемых на команду или для глубокого анализа состояния ККМ.
5. Раскодированные из пункта 4 параметры, записанные в виде пар ключ=значение через «;». Количество параметров зависит от команды.

```
d;0;0шибок нет;PENvb{... вырезано
BASE64.. }YmxlPg==;getDeviceStatus.inn=1656087840;getDeviceStatus.isCapOpen=0;ge
tDeviceStatus.isDecimalPointPosition=0;getDeviceStatus.isLowerSensorPresent=0;ge
tDeviceStatus.flagsFR=46;getDeviceStatus.isOpticalSensorCheckTape=0;getDeviceSta
tus.isOpticalSensorOperationJournal=0;getDeviceStatus.isErrorLeftSensor=255;getD
eviceStatus.buildSoftFR=103;getDeviceStatus.isLeverThermalHeadCheckTape=0;getDev
iceStatus.currentDocNumber=15;getDeviceStatus.dateSoftFiscalMemory=;getDeviceSta
tus.isBufferNotEmpty=255;getDeviceStatus.countLeftRefiscalizations=0;getDeviceSt
atus.isUpperSensorPresent=214;getDeviceStatus.numberLastCl
ousedSession=108;getDeviceStatus.buildSoftFiscalMemory=108;getDeviceStatus.opera
torNumber=30;getDeviceStatus.subModeFR=0;getDeviceStatus.isOnline=1;getDeviceSta
tus.isEk1zFull=255;getDeviceStatus.isRollCheckTapePresent=1;getDeviceStatus.isEk
1zPresent=0;getDeviceStatus.countFreeRecordsInFiscalMemory=0;getDeviceStatus.ful
lSerialNumber=0458210010002651;getDeviceStatus.deviceErrorCode=0;getDeviceStatus
.portFR=2;getDeviceStatus.isRollOperationJournalPresent=255;getDeviceStatus.isLe
verThermalHeadControlTape=0;getDeviceStatus.dateSoftFR=;getDeviceStatus.countRef
iscalizations=0;getDeviceStatus.dateFR=2019.05.25;getDeviceStatus.timeFR=14:50:0
8;getDeviceStatus.versionSoftFR=;getDeviceStatus.isMoneyBoxOpen=0;getDeviceStatu
s.isErrorRightSensor=255;getDeviceStatus.versionSoftFiscalMemory=;getDeviceStatu
s.serialNumber=2651;
```

Таким образом, получение всех ответных параметров становится простой задачей с точки зрения программирования.

Проверка успешности печати чека

Для проверки успешности печати чека необходимо:

1. Сравнить количество строк в файле-задании и файле-ответе. Оно должно совпадать.
2. Во всех строках на позиции 1 (нумерация с нуля) должен быть код 0 – отсутствие ошибок.

При выполнении этих 2-х условий можно быть уверенным в том, что чек успешно распечатался.

Если произошла ошибка, то её код и описание будут в позициях 1 и 2 последней строки. Кассовый сервер останавливает выполнение файла-задания как только случается ошибка.

Ещё одним способом является проверка фискального номера и фискального признака документа. Можно запросить номер документа ДО пробития чека и ПОСЛЕ. Если номер изменился, значит все фискальные операции прошли успешно.

```
get_fiscal_mark;0
b;0;ФИО_Кассира;0
set_tlv;1008;5;adres@post.mail
smde;НАЗВАНИЕ_ТОВАРА_УСЛУГИ;12345;123456;2;1;1;1;3;4
```

```
tmde;1524064;0;0;0;0;8  
get_fiscal_mark;0
```

Алгоритм печати чека с реакцией на ошибки

В процессе работы ККМ могут возникать ошибки. Как правило, это:

- Обрыв бумаги
- Просроченная смена
- Потеря связи с ккм
- Ошибка в параметрах чека

Ниже приводим рекомендуемый алгоритм реакции внешнего ПО на возникновение ошибки в процессе печати чека:

1. Сформировать задание
2. Послать задание в Кассовый сервер удобным транспортом
3. Получить ответ.
 - а. Если ответ не получен, значит связи с Кассовым сервером нет.
 - б. Если ответ получен, то проверить, что чек успешно распечатался.
 - i. Если чек распечатался верно, то перейти к п.0.
 - ii. Если обнаружена ошибка, то:
 1. Ошибка 255 – нет связи с кассой.
 - а. Сформировать запрос с командой d; -- запрос статуса.
 - б. Повторять отправку запроса статуса до тех пор, пока касса не ответит. Так же необходимо устранить причину отсутствия связи с кассой (её могли обесточить, к примеру).
 - с. Если связь с кассой отсутствует долго, возможно имеет смысл пометить этот чек как не распечатанный и прекратить попытку печати до устранения неисправности ККМ.
 2. Ошибка, связанная с истекшей сменой.
 - а. Заккрыть смену
 - б. Повторить печать чека
 3. Иная ошибка
 - а. Послать команду продолжения печати после обрыва ленты
 - б. Аннулировать чек
 - с. Повторить печать чека
 - д. Если после N попыток чек не распечатан, то скорее всего имеется или ошибка в формате чека (в параметрах) или касса требует обслуживания и не может распечатать чек. Прекращаются попытки печати чека до устранения неисправности ККМ.

Транспорты

Управлять Кассовым сервером можно несколькими способами. Они отличаются друг от друга только способом доставки набора строк-команд от внешней информационной системы к Кассовому серверу, и ответа в обратном направлении.

FileAPI

Для использования FileAPI необходимо сохранить в файл с расширением txt набор строк-команд. Файл помещается в отслеживаемую Кассовым сервером папку. После исполнения файл-задание удаляется.

Результат исполнения файла-задания помещается в файл, имеющий точно такое же имя, как и у исходного файла-задания, но к которому добавили суффикс .answer.

Строки-ответы помещаются в файл строго в той последовательности, в которой они были в оригинальном файле-задании.

Если в Кассовом сервере включена опция «Контролировать уникальность имен файлов-заданий», то дополнительно будет отслеживаться, чтобы 2 раза файл с одинаковым названием не исполнялся бы Кассовым сервером.

Иначе говоря:

- Файл-0.txt -- исполнится, если это не повтор
- Файл-1.txt -- исполнится
- Файл-1.txt – не исполнится, т.к. предыдущий файл имеет такое же имя
- Файл-2.txt – исполнится, имя файла отличается от предыдущего

Кассовый сервер при сканировании каталога использует сортировку ПО ИМЕНИ ФАЙЛА. Это позволяет избавиться от проблемы рассинхронизации времени создания файлов в случае, когда файлы поступают с разных рабочих мест с разными настройками даты-времени (отличаются часовые пояса, к примеру).

Рекомендуется придерживаться следующего формата имени файла:

ГГГГММДДЧММСС-{НазваниеРМК}-{НомерДокументаВоВнешнейИС}.txt

FtpAPI

Полностью аналогичен FileAPI, только данные располагаются на FTP-сервере.

Сканирование FTP-сервера производится примерно 2-3 раза в минуту.

WebAPI

Данный вид транспорта создан для связи с интернет-магазинами(ИМ) и иными ИС, работающими в асинхронном режиме.

Кассовый сервер с заданной периодичностью опрашивает ИМ (скрипт запроса задания) на предмет новых чеков (и любых иных команд для исполнения на ККМ).

После получения задания, Кассовый сервер исполняет команды на ККМ. Формирует ответ.

Ответ передается в ИМ (скрипт ответа) [этап «Получи»].

(далее если активирована опция «Использовать усиленное подтверждение ответа»)

Кассовый сервер ожидает от ИМ в ответ положительную квитанцию об успешной доставке результата исполнения команд Кассовым сервером [этап « и распишись...»]

Использование Токена сессии

В качестве средства авторизации добавлена поддержка работы токена.

Принцип работы:

1. Делается запрос по адресу ПОЛУЧИТЬ ЗАДАНИЕ или ПЕРЕДАТЬ

ОТВЕТ. Если статус исполнения = 301, значит серверу для работы необходимо передать токен.

2. Производится обращение Устанавливаются заголовки к скрипту ПОЛУЧИТЬ_ТОКЕН.

Content-Type = application/json-patch+json

accept = text/plain

Содержимое POST:

```
{"userName": "ИМЯ_ПОЛЬЗОВАТЕЛЯ", "password": "ПАРОЛЬ"}
```

3. В ответ получаю значение accessToken

4. Полученное значение accessToken далее используется в работе всех скриптов:

Authorization = Bearer ТОКЕН

Content-Type = application/json

Скрипт запроса задания

Владелец ИМ (программист) создает скрипт, обратившись к которому Кассовый сервер может получить новое задание, или удостовериться, что его нет.

Передаваемые параметры в составе адреса (GET):

1. kkm_id -- идентификатор кассы. Должен совпадать в Кассовом сервере и в ИМ.

В ответ сервер должен прислать XML-сообщение (конверт, package):

<package id="0" />

в случае, если новых команд в очереди нет

<package id="X">

<data>ДАННЫЕ</data>

</package>

где

ДАННЫЕ – это кодированные алгоритмом BASE64 набор строк-команд в соответствии с FileAPI.

X – число, идентификатор задания на печать чека (внутренний ИД в очереди печати)

чеков ИМ). Так же как с в случае с FileAPI, производится контроль за уникальностью двух смежных ID package.

Полученные ДАННЫЕ Кассовый сервер раскодирует и приступит к исполнению.

Скрипт получения ответа

После исполнения полученных команд Кассовый сервер должен сообщить ИМ (или иной ИС) о результатах своей работы. Для этого он совершает запрос к Скрипту ответа.

Передаваемые параметры в составе адреса (GET):

1. kkm_id – идентификатор кассы
2. package_id – идентификатор задания, на который передается ответ.

В теле запроса (POST) передается переменная data, которая содержит в себе строки-ответы в соответствии с FileAPI, которые кодированы с применением BASE64URL.

Внимание! Отличие обычного BASE64 от BASE64URL состоит в используемом алфавите кодирования.

Пример функций кодирования и декодирования на языке PHP

```
function base64url_encode($data, $pad = null) {
    $data = str_replace(array('+', '/'), array('-', '_'), base64_encode($data));
    if (!$pad) {
        $data = rtrim($data, '=');
    }
    return $data;
}

function base64url_decode($data) {
    return base64_decode(str_replace(array('-', '_'), array('+', '/'), $data));
}
```

Усиленное подтверждение ответа

Использовать опцию «Усиленное подтверждение ответа» имеет смысл использовать в том случае, когда имеются неполадки в линиях связи, возможна некорректная работа Скрипта получения ответа или в целях обеспечения максимальной надежности транспортного уровня.

Если эта опция включена, то Кассовый сервер будет передавать ответ в ИМ (Скрипту получения ответа) до тех пор, пока ИМ в ответ не передаст положительную квитанцию о приеме и сохранении данных. В теле ответа должна присутствовать следующая запись:

<package id="ИД_ПАКЕТА" result="ok" />

Как только Кассовый сервер получает такой ответ, он считает что последовательность операций по печати чека и передачи данных о его печати завершена. Можно приступать к следующей итерации запроса данных на печать чека.

Если по какой-то причине сервер не вернул данные в ответ на обращение к Скрипту получения ответа, или вернул отрицательную квитанцию

**<package id="ИД_ПАКЕТА" result="error"
error="ТЕКСТОВОЕ ОПИСАНИЕ ОШИБКИ" />**

AjaxAPI

Этот вид транспорта предназначен для организации печати чеков напрямую с WEB-страниц, Android-приложений и прочих мобильных систем.

В состав Кассового сервера встроен WEB-сервер, который осуществляет коммуникацию с WEB-страницей через AJAX.

Для отправки запроса необходимо обратиться по адресу <http://127.0.0.1:8888/req> и в параметре data POST-запроса передать строки-команды.

Например, так:

результат будет помещён в элемент с id «result». Дополнительно рекомендуется провести проверку успешности печати чека и оповестить пользователя о результате проверки.

```
printFR('b;0;Кассир Пупкин Иван Ильич;\nsmde;Очень ходовой
товар;56712;2323;0;3;\nsmde;Второй ходовой товар;25000;6000;0;3;\
ntmde;281742;0;');
function printFR(data) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("result").innerHTML =
this.responseText;
        }
    };
    xhttp.open("POST", "http://127.0.0.1:8888/req", true);
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhttp.send("data="+data);
}
```

В результате будет распечатан вот такой чек:



Для тестирования функционала AjaxAPI подготовлена тестовая страница http://rnditsoft.ru/test_ajax.html

Команды протокола

Выбрать кассу

Производит переключение на кассу, конфигурация которой описана в конфигурационном файле qkkmserver-<ИД_КАССЫ>.cfg.xml

| `connect;ИД_КАССЫ` |

При запуске используется стандартный конфигурационный файл qkkmserver.cfg.xml, но в процессе работы можно организовать управление несколькими ККМ с помощью одного СуперВизора.

Имейте ввиду, что переключение происходит не мгновенно. В среднем на перезапуск ядра уходит 3-7 секунды.

Например, имя файла-конфигурации qkkmserver-IP_Ivanov.cfg.xml

```
connect;IP_Ivanov
```

Открыть новую смену

Открывает новую кассовую смену

| `open_session` |

Печать строки текста

Выводит на чековую ленту строку произвольного текста. Нефискальная операция.

| `p;ТЕКСТ` |

1. Название команды
2. ТЕКСТ – строка. UTF-8.

Печать множества строк текста (SLIP-чек)

Выводит на чековую ленту строку произвольного текста. Нефискальная операция.

| `pt;ТЕКСТ` |

1. Название команды
2. ТЕКСТ – строка. UTF-8. Содержит в себе множество строк. Признаком переноса (разбиения строки на подстроки) является маркер #kkm_br#

Пример

```
pt;Строка-1#kkm_br#Строка-2#kkm_br#Строка-3
```

Печать строки заданным шрифтом

Выводит на чековую ленту строку произвольного текста заданным номером шрифта. Нефискальная операция.

| *print_font;НОМЕР_ШРИФТА;ТЕКСТ* |

1. Название команды
2. НОМЕР_ШРИФТА – целое число, номер встроенного в ККМ шрифта.
3. ТЕКСТ – строка. UTF-8.

Печать строки жирным шрифтом

Выводит на чековую ленту строку произвольного текста жирным шрифтом. Нефискальная операция.

| *print_bold;ТЕКСТ* |

1. Название команды
2. ТЕКСТ – строка. UTF-8.

Отчет без гашения (X-отчет)

Печатает промежуточный отчет без гашения. Смена не закрывается.

| *x* |

Отчет с гашением (Z-отчет)

Печатает отчет с гашением. Смена закрывается.

| *z* |

Отчет по секциям ККМ

Печатает отчет по секциям ККМ. Смена не закрывается.

| *rep_sections* |

Открыть чек

Открывает чек с заданными параметрами. Команда обязательная.

| *b;ТИП_ЧЕКА;КАССИР;ПЕЧАТЬ* |

1. Название команды
2. ТИП_ЧЕКА – число. Указывает на тип открываемого чека
 - a. 0 – чек продажи
 - b. 1 – чек возврата продажи
 - c. 2 – чек покупки

- d. 3 – чек возврата покупки
- 3. КАСИР – строка. ФИО и должность кассира. Печатается на чеке и передается в ОФД. Если значение не задано, то берётся наименование кассира №30 из настроек ККМ. Обычно это СИСТ.АДМИН.
- 4. ПЕЧАТЬ – число. Флаг необходимости печатать чек на бумаге. Корректно работает только если задать получателя электронного чека через set_tlv 1005.
 - a. 0 – не печатать чек
 - b. 1 – печатать чек (по-умолчанию).

b;0;ФИО_Кассира;0

Добавить товар в чек

Добавить товарную позицию в чек.

| smde;НАЗВАНИЕ_ТОВАРА;ЦЕНА;КОЛВО;НАЛОГ;СЕКЦИЯ;ПСР;ППР |

1. Название команды
2. НАЗВАНИЕ – строка. Задаёт наименование товара / услуги.
3. ЦЕНА – целое число. Цена товара в копейках. Указывается с учетом всех скидок и наценок.
4. КОЛВО – целое число. Количество товара. Для указания 1шт или 1кг надо поставить значение 1000.
5. НАЛОГ – целое число. Номер налога из настроек ККМ «Таблица налогов».
 - Для Штрих-М
 - 0 — Из настроек ККМ
 - 1 — НДС 20%
 - 2 — НДС 10%
 - 3 — НДС 0%
 - 4 — Без НДС
 - Для АТОЛ
 - 0 — Из настроек ККМ
 - 1 — НДС 0%
 - 2 — НДС 10%
 - 3 — НДС 20%
 - 4 — БЕЗ НДС
6. СЕКЦИЯ – целое число. В какую секцию разнести данный товар.
7. ПСР — признак способа расчета.
 - (1) Предоплата 100%
 - (2) Предоплата
 - (3) Аванс
 - (4) Полный расчет
 - (5) Частичный расчет и кредит
 - (6) Передача в кредит
 - (7) Оплата кредита
8. ППР — признак предмета расчета
 - (1) Товар
 - (2) Подакцизный товар

- (3) Работа
- (4) Услуга
- (5) Ставка азартной игры
- (6) Выигрыш азартной игры
- (7) Лотерейный билет
- (8) Выигрыш лотереи
- (9) Предоставление РИД
- (10) Платеж
- (11) Агентское вознаграждение
- (12) Составной предмет расчета
- (13) Иной предмет расчета

Например, комиссия агента 100 руб, без НДС, при денежном переводе будет оформлена так:

```
smde;Комиссия агента;10000;1000;4;1;1;1;1
```

Аннулирование чека

Отмена ещё не закрытого (не оплаченного) чека

```
| g |
```

1. Название команды

Продолжить печать

Позволяет вывести ККМ из аварийного состояния после пропадания электропитания во время печати, обрыва ленты и прочей ошибки..

```
| continue_print |
```

1. Название команды

Передать TLV-реквизит в ККМ

Установить (передать) произвольный реквизит ФФД в ККМ.

```
| set_tlv;ТЭГ;ТИП;ДАННЫЕ |
```

1. Название команды
2. ТЭГ – целое число. ТЭГ в соответствии с ФФД
3. ТИП – целое число. Тип данных, устанавливаемых в результате передачи в ККМ ТЭГа.
 - a. 0 – массив бай
 - b. 1 – 1 байт
 - c. 2 – 2 байта
 - d. 3 – 4 байта
 - e. 5 – строка
4. ДАННЫЕ – непосредственно данные, которые надо записать в ККМ.

- a. Массив – кодируется как последовательность HEX-значений. Например, массив из 5 байт будет иметь вид: 010F03A405.
- b. 1 байт: в виде HEX-значения. Например, df.
- c. 2 байта: в виде HEX-значения. Например, df0a.
- d. 4 байта: в виде HEX-значения. Например, df0afa01.
- e. Строка: передается «как есть», в виде строка для печати

Пример:

Установить адрес покупателя для электронного чека:

```
set_tlv;1008;5;test@test.test
```

Установить СНО в ОСН

```
set_tlv;1055;01
```

Константы для СНО

- 01 – ОСН
- 02 – Упрощенная доходы
- 04 – Упрощенная доходы – расходы
- 08 – ЕНВД
- 16 – Единый сельхоз налог
- 32 -- Патент

Оплатить чек (короткая форма) **Не соответствует ФФД 1.1**

Закрытие чека одной формой оплаты.

```
| tmde;СУММА;НОМЕР_ОПЛАТЫ |
```

1. Название команды
2. СУММА – целое число. Сумма, которой производится закрытие чека.
 - a. Если закрытие происходит наличными, то СУММА должна быть не менее суммы всех стоимостей продаж по чеку. При превышении -- рассчитывается сдача.
 - b. При всех других формах оплаты СУММА должна точно совпадать с суммой стоимостей всех строк чека. Сдачи по безналичной (электронной) форме оплаты не существует.
3. НОМЕР_ОПЛАТЫ – целое число. Номер формы оплаты из настроек ККМ «Таблица формы оплаты»
 - a. 0 – всегда соответствует форме оплаты «наличными»

Оплатить чек (длинная форма) **Не соответствует ФФД 1.1**

Закрытие чека смешанное по формам оплаты.

```
| tmde;СУММА1; СУММА2; СУММА3; СУММА4;НОМЕР_НАЛОГА_1; НОМЕР_НАЛОГА_2;  
НОМЕР_НАЛОГА_3; НОМЕР_НАЛОГА_4 |
```

1. Название команды
2. СУММА1 – целое число. Сумма, которой производится закрытие чека. СУММА1 – всегда соответствует наличной форме оплаты. Если не оплачивается, то равно 0.
3. СУММА2 -- целое число. Сумма второй формы оплаты. Если не оплачивается, то равно 0.
4. СУММА3 -- целое число. Сумма 3-й формы оплаты. Если не оплачивается, то равно 0.
5. СУММА4 -- целое число. Сумма 4-й формы оплаты. Если не оплачивается, то равно 0.
6. НОМЕР_НАЛОГА_1 – целое число. Номер налога из таблицы «Налоги»
7. НОМЕР_НАЛОГА_2 – целое число. Номер налога из таблицы «Налоги»
8. НОМЕР_НАЛОГА_3 – целое число. Номер налога из таблицы «Налоги»
9. НОМЕР_НАЛОГА_4 – целое число. Номер налога из таблицы «Налоги»

Оплатить чек

Соответствует ФФД 1.1

Закрытие чека

| *tmde;СУММА1; СУММА2; СУММА3; СУММА4;СУММА5;СНО* |

1. Название команды
2. СУММА1 – целое число. Сумма, которой производится закрытие чека. СУММА1 – всегда соответствует наличной форме оплаты. Если не оплачивается, то равно 0.
3. СУММА2 -- оплата электронными средствами платежа (карты, электронные деньги и т.п.) Если не оплачивается, то равно 0.
4. СУММА3 -- предоплата. Если не оплачивается, то равно 0.
5. СУММА4 -- постоплата. Если не оплачивается, то равно 0.
6. СУММА5 -- оплата встречным. Если не оплачивается, то равно 0.
7. СНО — применяемая система налогообложения.

Константы для СНО

- 01 – ОСН
- 02 – Упрощенная доходы
- 04 – Упрощенная доходы – расходы
- 08 – ЕНВД
- 16 – Единый сельхоз налог
- 32 -- Патент

Внесение денег в кассу

Добавление денег в кассу (денежный ящик) не связанное с торговлей.

| *imde;СУММА* |

1. Название команды
2. СУММА – целое **положительное** число. Сумма наличных добавляемая в денежный ящик.

Выемка денег из кассы (инкассация)

Добавление денег в кассу (денежный ящик) не связанное с торговлей.

| *imde;СУММА* |

1. Название команды
2. СУММА – целое **отрицательное** число. Сумма наличных извлекаемых из денежного ящика.

Протяжка ленты

Продвигает ленту на указанное количество строк.

| *feed;КОЛВО_СТРОК* |

1. Название команды
2. КОЛВО_СТРОК – на какое количество строк (высоты стандартного шрифта) продвинуть ленту.

Отрезка чека

При наличии ножа-отрезчика, производит полную отрезку чека.

| *cut_check* |

1. Название команды

Запрос денежного регистра

Получает значение накоплений денежных средств по указанному регистру.

| *m;РЕГИСТР* |

1. Название команды
2. РЕГИСТР – номер денежного регистра, по которому производится запрос накоплений

Результат

Для получения результата, надо обратиться к 3-му параметру строки-ответа, раскодировать его из BASE64 и получить XML примерно такого вида:

```
<ControlProtocol messageType="answer">
<error text="Ошибок нет" id="0"/>
<getMoneyReg idReg="id" Value="Значение"/>
</ControlProtocol>
```

Далее необходимо получить Значение из атрибута Value.

Печать копии последнего чека

Выводит на чековую ленту дубликат последнего распечатанного чека.

| c

1. Название команды

Открыть денежный ящик

Подается сигнал на открытия денежного ящика.

| open_cash_box;НОМЕР_ЯЩИКА

1. Название команды
2. НОМЕР_ЯЩИКА – целое положительное число. Обычно = 0.

Чек коррекции на сумму

В случае, предусмотренном 54-ФЗ необходимо бывает пробить чек коррекции.

| correction;ТИП_ЧЕКА;СУММА

1. Название команды
2. ТИП_ЧЕКА – строка. Принимает значения:
 - a. docIn – чек продажи
 - b. docRetIn – чек возврата продажи
 - c. docOut – чек покупки
 - d. docRetOut – чек возврата покупки
3. СУММА – целое число. Сумма в копейках.

Запрос фискального признака и фискального номера чека

После регистрации чека на ОнЛайн касса формируется фискальный номер и признак документа .

| get_fiscal_mark;ФНДок

1. Название команды
2. ФНДок – число. Фискальный номер документа, по которому запрашивается фискальный признак. Если поставить 0, то будет извлечена информация о последнем документе.

Результат

Для получения результата, надо обратиться к 3-му параметру строки-ответа, раскодировать его из BASE64 и получить XML примерно такого вида:

```
<ControlProtocol messageType="answer">
<error text="Ошибок нет" id="0"/>
```

```
< GetFiskalMarkById
    fiskalDocId="НОМЕР_ЧЕКА_ИЗ_ФП"
    fiskalMark="ФИСКАЛЬНЫЙ ПРИЗНАК"
    date_time="дата-время печати чека"/>
</ControlProtocol>
```

Получить номер чека и номер смены

Для формирования электронной копии чека порой надо знать номер чека и номер смены. Существует 4 команды, позволяющие получить значения номера чека для 4-х типов операций: продажа, возврат продажи, покупка и возврат покупки

```
get_num_sale_check;

get_num_returnsale_check;

get_num_buy_check;

get_num_returnbuy_check;
```

1. Название команды

Результат

Для получения результата, надо обратиться к 3-му параметру строки-ответа, раскодировать его из BASE64 и получить XML примерно такого вида:

```
<ControlProtocol messageType="answer">
<error text="Ошибок нет" id="0"/>
<GetNumSaleCheck session="0" value="" />
</ControlProtocol>
```

В зависимости от команды, на месте GetNumSaleCheck может быть

- GetNumSaleCheck для get_num_sale_check
- GetNumReturnSaleCheck для get_num_returnsale_check
- GetNumBuyCheck для get_num_buy_check
- GetNumReturnBuyCheck для get_num_returnbuy_check

В атрибуте:

- Session -- будет возвращен номер смены
- Value - будет возвращен номер последнего закрытого чека

Печать заголовка чека

Печатается стандартный заголовок чека.

```
| print_clishe
```

1. Название команды

Печать рекламного заголовка

Печатается стандартный рекламный заголовок чека.

| `print_reklama` |

1. Название команды

Печать изображения из памяти ККМ

Печатается картинка, предварительно загруженная в память ФР (логотип).

| `print_image;НАЧ;КОН` |

1. Название команды
2. НАЧ – целое число. Начальная линия (пиксель) изображения из памяти.
3. КОН – целое число. Конечная линия (пиксель) изображения из памяти.
В самом простом случае, НАЧ и КОН соответствуют высоте изображения (0 и ВЫСОТА).
Если надо напечатать часть изображения, то могут быть уже варианты.

Запрос полного статуса ККМ

Возвращается полный статус ККМ.

| `d` |

1. Название команды

Результат

Для получения результата, надо обратиться к 3-му параметру строки-ответа, раскодировать его из BASE64 и получить XML примерно такого вида:

```
<ControlProtocol messageType="answer">
<error text="Ошибок нет" id="0"/>
<getDeviceStatus
isOnline="1"
deviceErrorCode="0"
dateFR="2012.12.11"
timeFR="20:03:20"
inn="88888888"
serialNumber="123456"
versionSoftFR="A.1"
dateSoftFR="Пт мар 22 2002"
buildSoftFR="4563"
versionSoftFiscalMemory="1.1"
dateSoftFiscalMemory="Пн сен 17 2001"
numberFR="1"
modeFR="8"
subModeFR="0"
flagsFiscalMemory="5"
```

```
operatorNumber="30"
portFR="0"
countLeftRefiscalizations="14"
currentDocNumber="284"
numberLastClousedSession="631"

countRefiscalizations="2"
flagsFR="979"
buildSoftFiscalMemory="31"
countFreeRecordsInFiscalMemory="1469"
isBufferNotEmpty="0"
isCapOpen="0"
isRollOperationJournalPresent="1"
isOpticalSensorCheckTape="1"
isDecimalPointPosition="1"
isUpperSensorPresent="0"
isLeverThermalHeadCheckTape="1"
isLeverThermalHeadControlTape="1"

isEklzFull="0"
isErrorRightSensor="0"
isErrorLeftSensor="0"

isRollCheckTapePresent="1"
isMoneyBoxOpen="0"
isEklzPresent="0"
isOpticalSensorOperationJournal="1"
isLowerSensorPresent="0"
statusMessageHTML="HTML-описание статуса"
/>
</ControlProtocol>
```

В зависимости от модели ККМ, часть полей может не иметь заполненного значения.

ОПИСАНИЕ ПАРАМЕТРОВ ОТВЕТА

isOnline	ККМ на связи или нет
deviceErrorCode	код ошибки
dateFR	текущая дата в часах ККМ
timeFR	текущее время в часах ККМ
inn	ИНН
serialNumber	заводской номер
versionSoftFR	версия программного обеспечения
dateSoftFR	дата сборки прошивки
buildSoftFR	номер сборки прошивки

versionSoftFiscalMemory	версия прошивки фискальной памяти
dateSoftFiscalMemory	дата изготовления прошивки фискальной памяти
numberFR	логический номер ККМ в зале
modeFR	режим работы ККМ
subModeFR	номер подрежима работы ККМ
flagsFiscalMemory	флаги фискальной памяти
operatorNumber	номер оператора
portFR	использование внешнего порта ККМ
countLeftRefiscalizations	количество оставшихся операций рефискализации
currentDocNumber	номер открытого документа продажи
numberLastClousedSession	номер последней закрытой сессии
countRefiscalizations	количество фискализаций которые уже были
flagsFR	флаги ККМ
buildSoftFiscalMemory	номер сборки прошивки фискальной памяти
countFreeRecordsInFiscalMemory	количество свободных записей в фискальной памяти

Логические параметры (0 - ложь, 1- истина)

isBufferNotEmpty	буфер документа не пуст
isCapOpen	состояние крышки ККМ
isMoneyBoxOpen	состояние денежного ящика
isRollOperationJournalPresent	наличие ленты операционного журнала (контрольной ленты)
isLeverThermalHeadControlTape	положение заслонки (рычага) контрольной ленты
isOpticalSensorOperationJournal	контрольная лента вставлена
isRollCheckTapePresent	наличие чековой ленты
isOpticalSensorCheckTape	чековая лента заправлена (оптический датчик)
isLeverThermalHeadCheckTape	состояние заслонка термопринтера чековой ленты
isUpperSensorPresent	состояние верхнего датчика
isErrorRightSensor	ошибка правого датчика
isErrorLeftSensor	ошибка левого датчика
isDecimalPointPosition	положение десятичного разделителя
isEklzFull	состояние ЭКЛЗ (закончилась?)
isEklzPresent	ЭКЛЗ присутствует в ККМ?
isLowerSensorPresent	нижний датчик присутствует (подкладного документа)

Ряд возвращаемых параметров в связи с переходом на ОнЛайн кассы утратил смысл, но необходим для обратной совместимости со старыми ККМ (их используют часто как простые термопринтеры).

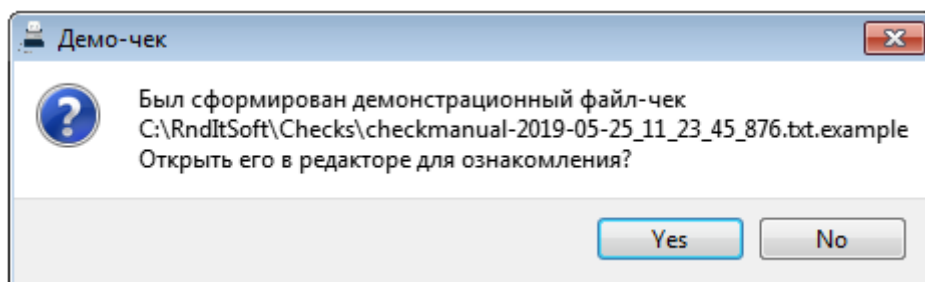
Примеры

Кассовый сервер построен таким образом, что он является максимально самодокументированным.

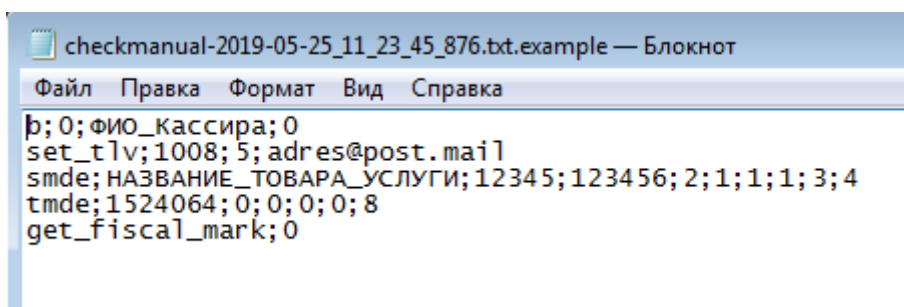
	Наименование	Цена	Количество	НДС	Сумма
1	НАЗВАНИЕ_ТОВАРА_УСЛУГИ	123,45	123,456	1385.51	15240.64

Например, в «Интерфейсе кассира» можно сформировать любой чек с нужными параметрами и, не печатая его на ККМ, посмотреть в текстовом редакторе.

Для этого необходимо сформировать чек, задать его параметры и, удерживая клавишу Ctrl, нажать на кнопку «Печать чека». В результате будет сформирован ПРИМЕР файла-задания, полностью идентичный «боевому», но отличающийся только окончанием имени файла .example.



Так же будет показан диалог, с помощью которого можно сразу же открыть файл в текстовом редакторе (Windows: notepad Linux: gedit)



Наиболее частые ошибки

1. Ошибки, связанные с долгим отсутствием связи с ОФД (более 30 дней).

Как правило, проявляются в виде отказа ККМ проводить любые фискальные операции.

Необходимо проверить на последнем фискальном или любом другом отчете строку «Количество не переданных документов в ОФД». В случае нормальной работы там будет число 0, т. е. все документы переданы.

Если там не 0, то надо проверить, почему отсутствует связь между ККМ и ОФД. Для этого обратитесь к документации на ККМ.

2. Отсутствует связь с ККМ

Проявляется в том, что в СуперВизоре в верхнем углу появляется надпись «СТОП ККМ! Нет связи с ККМ», «Ошибка ФР: Connection timed out» или «СТОП ККМ! Нет связи с сервером»

Это означает, что либо нет физического соединения с ККМ (обрыв провода, выключена касса и т. п.), либо выбраны не правильные параметры ККМ (порт, сетевой адрес, скорость, модель ККМ и т. п.).

Проверьте подключение, в том числе в интерфейсе «Настройка ККМ».

3. ККМ не выполняет команду

Команда подана, однако ККМ её игнорирует.

Причины:

а) формат записи команды не соблюден — тогда файл-задание удаляется из каталога заданий. В протоколе ничего не отображается (т. к. никакая кассовая команда выполнена не была)

б) не соблюдено количество или значение параметров — появляется запись в протоколе «Not correct message. Ignore.»

Общие комментарии

Проект «Кассовый сервер QKkmServer» находится в постоянном развитии. Особенно в последнее время с введением 54-ФЗ об ОнЛайн кассах.

Команда проекта будет признательна, если Вы будете сообщать об ошибках и неточностях в работе ПО в техподдержку.

Так же принимаются пожелания по внедрению нового функционала.

Контактные данные:

Сайт www.rnditsoft.ru

Почта rndit@mail.ru

skype: [knyazev_em](#)

telegram: [+79282701319](#)

tel: [+79282701319](#)

Руководитель проекта: Князев Евгений Михайлович