

БИТ драйвер ККТ

АПИ фискализации чеков для облачных решений вер. 1.1

Работает с версией программы БИТ драйвер ККТ не ниже 1.19.

1 Режимы работы

Надо сразу разделить два режима использования АПИ для фискализации чеков.

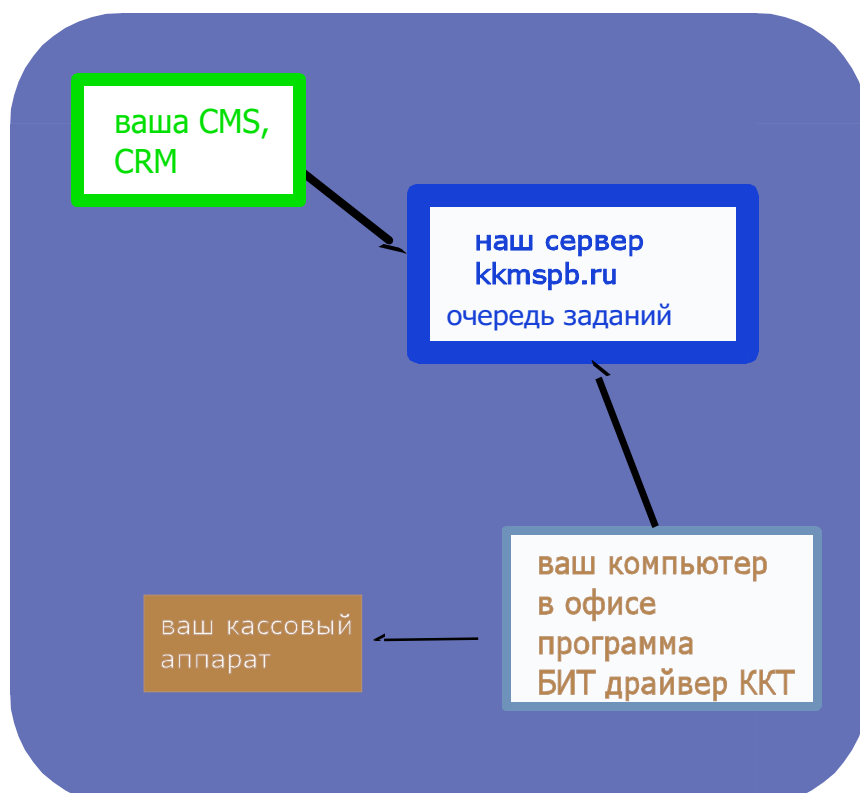
1.1 *Режим товароучетки*

Режим товароучетки будем называть когда сам продавец принимает оплату например наличными или по банковской карте (через банковский терминал) на локально подключенном оборудовании.

1.2 *Режим оплаты в облаке*

Если оплата происходит в облаке, то есть автоматически без участия продавца (например через платежный сервис, прикрученный к сайту), то этот режим будем называть - **режим оплаты в облаке**. Это режим когда покупатель самостоятельно оплачивает товар без участия продавца.

Общая схема работы



2 Тестовая страница

2.1 Сначала мы обсудим *режим товаручетки*.

У нас создана тестовая страница для проверки фискализации чеков: <https://kkmspb.ru/api/test.php> . По факту ее правильнее разместить на вашем сайте (но это не принципиально).

Вы в своей облачной программе (CMS/CRM) подготавливаете данные для пробития чека в формате POST запроса.

Далее (на примере [test.php](https://kkmspb.ru/api/test.php)) у вас две кнопки (для двух режимов).

Из вашей облачной программы (на примере [test.php](https://kkmspb.ru/api/test.php)) вы передаете задание на фискализацию чека. По сути происходит обычный переход на страницу

нашего сервера <https://kkmspb.ru/api/payment-dlg.php> для оплаты (далее **стр а ница оплаты**).

На странице оплаты происходит проверка связи с ккт и предлагается принять оплату через локально подключенное оборудование.

В случае успешного пробития чека мы вас возвращаем на указанное вами (в качестве параметра) значение **BIT_CALLBACK_SUCCESS** или в случае не успешного результата мы вас возвращает на **BIT_CALLBACK_FAILED**.

3 Для режима оплаты в облаке

В режиме **оплаты в облаке** вы также со своего сайта (или на примере test.php) подготавливаете такие же POST данные и посылаете их на страницу <https://kkmspb.ru/api/fiscal/create-receipt.php> , и вам возвращается результат постановки в очередь на пробитие чека.

Для демонстрации использования нашего сервиса в режиме оплаты в облаке мы используем (для наглядности процесса) страницу нашего сайта <https://kkmspb.ru/api/create-receipt-dlg.php> . Изначально переход на нее происходит со страницы (test.php), далее вы наблюдаете процесс в режиме онлайн.

На странице <https://kkmspb.ru/api/create-receipt-dlg.php> (это уже на стороне нашего сервера) мы ставим чек на пробитие в очередь. Далее программа БИТ драйвер ККТ, установленная на вашем ПК, примерно 1 раз в 5 минут забирает из очереди с сервера чеки на фискализацию и пробивает чек на кассовом аппарате, подключенном к вашему ПК.

Как пробивать чек быстро без задержек

Если вы хотите фискализировать чек как можно быстрее, то можно указывать в пакете POST дополнительный параметр **BIT_PROG_URL**. Это внешний статический ip адрес (с портом) вашего роутера (формат <https://xxx.xxx.xxx.xxx:44735>). Далее на роутере надо пробросить порт **44735** на компьютер во внутренней сети, на котором установлена программа БИТ драйвер ККТ, к которой подключена ваша касса. На этот url будет приходить команда для немедленного выполнения пробития чека (вне очереди) и результат также будет возвращаться вам сразу по факту пробития.

Для режима товароучетки параметр **BIT_PROG_URL** не актуален, так как там обращение из браузера идет на url <http://localhost:44735>, то есть

минуя интернет, сразу на ваш ПК.

Конечно может случиться так, что чек завершился не успешно. Вы в любом случае получите причину ошибки, так как отдает сам кассовый аппарат (в соответствии со своим протоколом).

4 Формат передаваемых данных

4.1 Пример передачи данных на страницу оплаты:

```
BIT_ACCOUNT_ID: 896
BIT_KKT_TOKEN: f039001210451fae2f18c2f6d75a5cc3
BIT_ORDER_ID: 1237
BIT_SIGNATURE: 949f8b85f8c1b6762ddd951681363b7b
BIT_CALLBACK_SUCCESS: http://kkmspb.ru/api/callback/success.php
BIT_CALLBACK_FAILED: http://kkmspb.ru/api/callback/failed.php
BIT_DATA: {
  "purchases": [
    {
      "productName_1030" : "товар 123",
      "price_1079" : 11.00,
      "qty_1023" : 1.00,
      "unit_2108" : 10,
      "paymentFormCode_1214" : 2,
      "productTypeCode_1212" : 3,
      "tax_1199" : 6,
      "additionalAttribut_1191": "что-то дополнительное"
    }
  ],
  "cashierName_1021": "Пупкин Иван",
  "cashierInn_1203": "",
  "payments": {
    "cash_1031" : 11.00,
    "ecash_1081" : 2.00
  },
  "taxationType_1055" : 5,
  "receiptType_1054" : 0
}
```

4.2 Единый протокол передачи содержания чека для разных ККТ

Имеется ввиду, что вы формируете json пакет в параметре **BIT_DATA** в нашем (удобном на наш взгляд) формате, где у всех реквизитов присутствует номер тега.

То есть используется следующая логика: названия ключей (например cashierName_1021) состоит из двух элементов, разделенных знаком подчеркивания. Справа это номер тега в соответствии с законом Ф354, слева может быть все, что вам покажется разумным. Примечание: знак подчеркивания должен быть только один.

Значение ключа (например "Пупкин Иван") это всегда значение как указано в законе Ф354. В данном случае строка.

Таким образом вы можете применять единый протокол для управления разными моделями кассовых аппаратов. На сегодня поддерживаются Атолы и Меркурии (список будет расширяться по вашим запросам).

Только ключ *"purchases"* имеет зарезервированное значение. Он используется для обозначение массива покупок (или продаваемых позиций) в чеке.

Соответственно все теги покупки должны находиться только в массиве *"purchases"*. В примере это все хорошо видно.

4.3 Формат данных представлен следующими частями:

BIT_ACCOUNT_ID — уникальный номер вашего аккаунта на сайте kkmspb.ru;

BIT_KKT_TOKEN – уникальный код вашего кассового аппарата (смотрите там же в личном кабинете);

BIT_ORDER_ID — это вами сгенерированный уникальный номер заказа (для защиты от дублирования оплат по одному заказу);

BIT_SIGNATURE — это ваша подпись, вычисляемая по алгоритму md5 для всех передаваемых (для гарантии защиты от подделки передаваемых ваших данных);

BIT_DATAINTEGRITY_CODE для вычисления хеша по всем данным вы должны еще использовать секретный (только вам известный) код проверки целостности данных, который вы сами задаете в личном кабинете на сайте kkmspb.ru. Это ваша страховка, что никто не подделает от вашего имени чек.

BIT_DATA — содержание чека в формате JSON. На тестовой странице вы его видите в нижней части.

BIT_CALLBACK_SUCCESSFUL – наименование вашей страницы (вашего

сайта) для получения успешного результата оплаты (пробития чека).

BIT_CALLBACK_FAILED - наименование вашей страницы (вашего сайта) для получения результата с ошибкой.

BIT_PROG_URL - наименование адреса в сети интернет для взаимодействия с программой БИТ драйвер ККТ напрямую из интернета. Можно указывать в режиме оплаты в облаке для ускорения полного процесса проведения фискализации чека до 5-10сек с получение результата пробития такжен сразу.

5 Генерация подписи

BIT_SIGNATURE вычисляется как **md5** по сумме следующих передаваемых параметров:

```
BIT_SIGNATURE = $.md5( $('#BIT_ACCOUNT_ID').val() +  
    $('#BIT_KKT_TOKEN').val() +  
    $('#BIT_ORDER_ID').val() +  
    $('#BIT_DATA').val() +  
    $('#BIT_CALLBACK_SUCCESS').val() +  
    $('#BIT_CALLBACK_FAILED').val() +  
    $('#BIT_DATAINTEGRITY_CODE').val())
```

В примере выше используется md5 библиотеки jquery. Все параметры передаются в составе POST блока.

6 Кодирование передаваемых данных

Данные передаются как обычная форма методом POST. Но есть нюансы. Перед отправкой содержание данных параметра **BIT_DATA** (это сам json текст чека) надо предварительно закодировать в **encodeURIComponent**, а потом еще в **base64** (см.btoa). Это важно, так как иначе спецсимволы такие как перенос строки \r\n будут переданы НЕ коррекно и данные json будет не раскодировать на строне сервера.

Надо еще отметить, что все остальные параметры формы передаются как есть и их дополнительно кодировать не нужно.

7 Программа БИТ драйвер ККТ

Конечно кто-то должен печатать чек. Предлагаем вам использовать свой ккт Атол или Меркурий (предположим он у вас есть и располложен в офисе и подключен к ПК с Windows).

Вам надо скачать с сайта <https://kkmspb.ru/software/BIT-driver-KKT/download/> последнюю программы **БИТ драйвер ККТ** (версию не ниже 1.19.xxx) и установить на свой ПК. Демо период 14 дней, если надо дольше мы можем продлить.

Инструкции и видео как подключать и настраивать кассовые аппараты к программе БИТ драйвер ККТ есть также на сайте:
https://kkmspb.ru/software/BIT_driver_KKT/attach-kkt/.

Есть важный нюанс при установке программа требует зарегистрироваться по email и надо указывать такой же как и email в вашем личном кабинете на kkmspb.ru.

8 Скачать АПИ

Все файлы АПИ с необходимыми библиотеками можно скачать с гитхаба:
<https://github.com/PavelDorofeev/API-receipt-fiscalization-for-CMS-and-CRM>