



## АПИ фискализации чеков для облачных решений вер. 1.2

Работает с версией программы БИТ драйвер ККТ не ниже 1.21.

### 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. Программа выполняется в браузере (на вашем сайте).

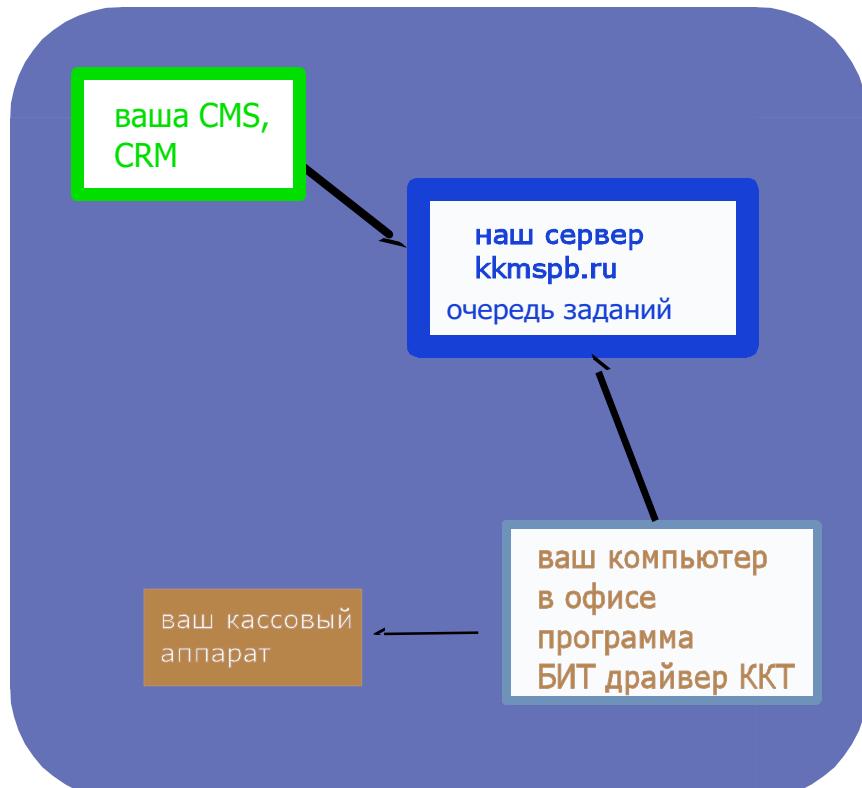
#### 1.2 Режим оплаты в облаке

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.

#### 1.3 Режим курьера

Это режим только для пробития чеков из авшего личного кабинета на нашем сайте [kkmspb.ru](http://kkmspb.ru). Позволяет фискализировать чеки на выезде к клиенту.

# Общая схема работы



## 2 Тестовая страница

### 2.1 Сначала мы обсудим *режим товароучетки*.

У нас создана тестовая страница для проверки фискализации чеков: <https://kkmspб.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](#)) у вас две кнопки (для двух режимов).

Из вашей облачной программы (на примере [test.php](#)) вы передаете пакет заданий например на прием банковской карты на банковском терминале и

на фискализацию чека на кассовом аппарате.

Этот пакет вы передаете нап <http://localhost:44735> и ждете завершения выполнения команд.

На <http://localhost:44735> у вас находится программа БИТ драйвер ККТ (соответственно запущенная под Windows).

В ответе вы получаете полную информацию о результатах выполнения команд.

### **3 Для режима оплаты в облаке**

Это режим только для фискализации чеков ( но не для принятия оплаты через банковский терминал ).

В режиме [оплаты в облаке](#) вы также со своего сайта (подготавливаете такие же POST данные и посыпаете 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 ( просто 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 фискализовать чек как можно быстрее, то можно указывать в пакете POST параметр **BIT\_PROG\_URL**. Это внешний статический ip адрес (с портом) вашего роутера (формат <https://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 случится так, что чек завершился не успешно. Вы в любом случае получите причину ошибки, так как отдает сам кассовый аппарат (в соответсвие со своим протоколом).

## 4 Формат передаваемых данных

### 4.1 Пример передачи данных на страницу оплаты:

BIT\_ACCOUNT\_ID: "здесь значение из вашего лк на kkmspb.ru"  
BIT\_KKT\_TOKEN: "здесь значение из вашего лк на kkmspb.ru"  
BIT\_BNK\_TRM\_TOKEN: "здесь значение из вашего лк на kkmspb.ru"  
BIT\_ORDER\_ID: 1237  
BIT\_SIGNATURE: 949f8b85f8c1b6762ddd951681363b7b  
BIT\_CALLBACK\_SUCCESS: <http://vspbkassa.ru/api/callback/success.php>  
BIT\_CALLBACK\_FAILED: <http://vspbkassa.ru/api/callback/failed.php>  
BIT\_DATA: [  
  {  
    "name": "1. Оплата по 'карте'",  
    "type": "bnkCardPayment",  
    "data":  
      {  
        "sum": 12  
      }  
  },  
  {  
    "name": "2. Фискализируем чек",  
    "type": "kktReceiptFiscalization",  
    "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 Трофимович",
    "cashierInn_1203": "",
    "payments": {
        "cash_1031" : 10.00,
        "ecash_1081" : 1.00,
        "prepayment_1215" : 0,
        "credit_1216" : 0,
        "barter_1217" : 0
    },
    "taxationType_1055" : 5,
    "receiptType_1054" : 1,
    "sendToEmail_1008" : "kkmspb2008@yandex.ru",
    "electronically":true
}
]

```

## **4.2 Единый протокол передачи содержания чека для разных ККТ**

Обратите внимание, что передается спитсок [ ... ] . Это список команд на выполнение по порядку. Если команда завершится не удачно, то следующая выолняться не будет.

Обязательным параметрами любой команды являются следующие:

- **name** : это произвольное ваше название команды;
- **type** : это зарезервированное значения из списка *kktReceiptFiscalization* или *bnkCardPayment* для идентификации команды в программе БИТ драйвер ККТ;
- **data** : здесь содержание команды;

Далее обратите внимание, что вы передаете содержание *команды пробития чека* в формате, где у всех реквизитов присутствует номер тега (после символа \_).

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 Меркурий (список будет расширяться по вашим запросам).

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

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

#### **4.3 Формат данных представлен следующими частями:**

**BIT\_ACCOUNT\_ID** — уникальный номер вашего аккаунта в личном кабинете на сайте [kkmspб.ru](http://kkmspб.ru);

**BIT\_KKT\_TOKEN** — уникальный код вашего кассового аппарата (смотрите в личном кабинете);

**BIT\_BNK\_TRM\_TOKEN** — уникальный код вашего банковского терминала (пин-пада) (смотрите 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 kkmspб.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\_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 также сразу (актуально только для режима оплаты в облаках и режима курьера).

## 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.

Актуально только для режима оплаты в облаке. Для режима товароучетки не имеет значения.

## 6 Кодирование передаваемых данных

Только для режима оплаты в облаке.

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 корректно 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 расположен в офисе и подключен к ПК с Windows).

Вам надо скачать с сайта <https://kkmspб.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/](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>