

# БИТ драйвер ККТ

## АПИ фискализации чеков для облачных решений вер. 1.0

### 1. Назначение

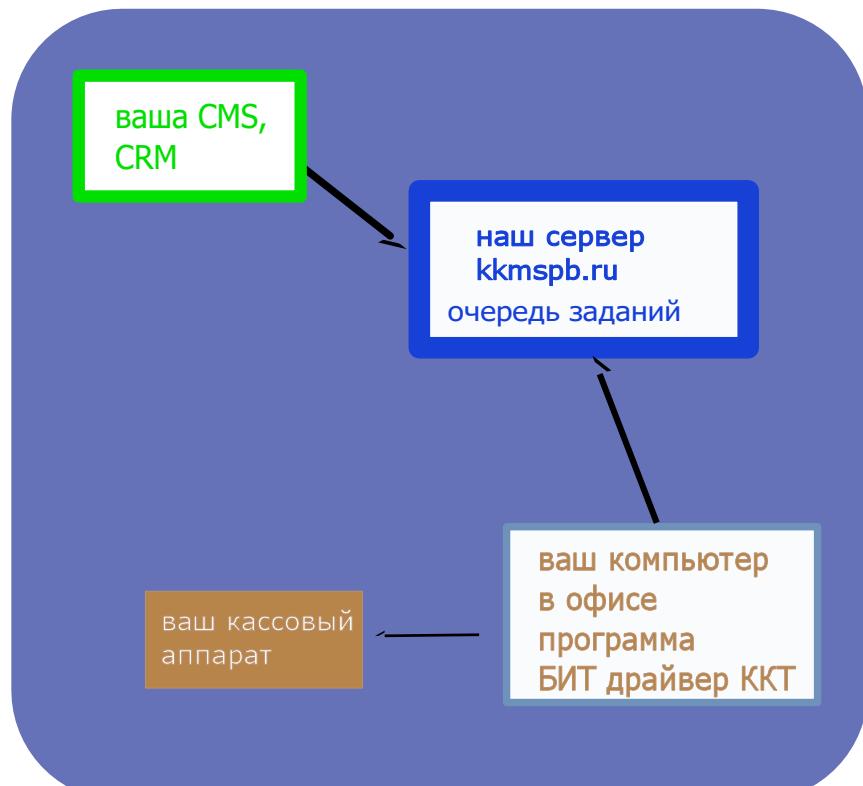
АПИ представлен в виде открытого ресурса в сети интернет. Это обычная страница на сайте [kkmspb.ru: https://kkmspb.ru/api/payment-dlg.php](https://kkmspb.ru/api/payment-dlg.php) (далее **страница оплаты**).

Вы в своей CMS/CRM подготавливаете данные для пробития чека в формате POST запроса и переходите на страницу оплаты.

Далее происходит проверка связи с ккт и предлагается принять оплату наличными с фискализацией чека.

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**.

### 2. Общая схема работы



### **3. Пример передачи данных на страницу оплаты:**

```
BIT_ACCOUNT_ID: 896
BIT_KKT_TOKEN: f039001210451fae2f18c2f6d75a5cc3
BIT_TRANSACTION_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_JSON: {
    "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"
        },
        {
            "productName_1030" : "товар 234",
            "price_1079" : 22.00,
            "qty_1023" : 1.00,
            "unit_2108" : 0,
            "paymentFormCode_1214" : 2,
            "productTypeCode_1212" : 3,
            "tax_1199" : 6
        }
    ],
    "cashierName_1021": "Пупкин Иван",
    "cashierInn_1203": "",
    "payments": {
        "cash_1031" : 11.00,
        "ecash_1081" : 2.00
    },
    "taxationType_1055" : 5,
    "receiptType_1054" : 0
}
```

В нашем протоколе используется следующая логика: названия ключей (например cashierName\_1021) состоит из двух элементов, разделенных знаком подчеркивания. Справа это номер тега в соответствии с законом ФЗ54, 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 ФЗ54. В данном случае строка.

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".

#### **4. Формат данных представлен следующими частями:**

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

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

**BIT\_TRANSACTION\_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](http://kkmspb.ru). Это ваша страховка, что никто не подделает от вашего имени чек.

#### **5. Генерация подписи**

BIT\_SIGNATURE md5 по сумме всех передаваемых параметров:

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

В примере выше используется md5 библиотеки jquery. Далее все параметры передаются в составе POST блока. Ниже дана ссылка на гитхаб, где приложен файл [test.php](#) для примера посылки правильного запроса.

## **6. Кодирование передаваемых данных**

Данные передаются как обычная форма методом POST. Но есть нюансы. Перед отправкой содержание данных параметра BIT\_JSON (это сам json текст чека) надо предвариатально закодировать в **encodeURIComponent**, а потом еще в **base64** (см.btoa). Это важно, так как иначе спецсимволы такие как перенос строки \r\n не будут переданы 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. Предлаем 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://kkmspb.ru/software/Mercury-KKT-test-driver/download/> последнюю программы **БИТ драйвер ККТ** (версию не ниже 1.18.xxx) и установить на свой ПК. Демо период 14 дней, если надо дольше мы продлим.

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/Mercury/](https://kkmspb.ru/software/BIT_driver_KKT/attach-kkt/Mercury/).

## **8. Пример тестовой страницы**

### **9.**

Ниже приложена тестовая страница на php. Используется библиотека jquery.

Тестовую страницу с необходимыми библиотеками можно скачать с гитхаба:

<https://github.com/PavelDorofeev/API-receipt-fiscalization-for-CMS-and-CRM>