



БИТ драйвер ККТ

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

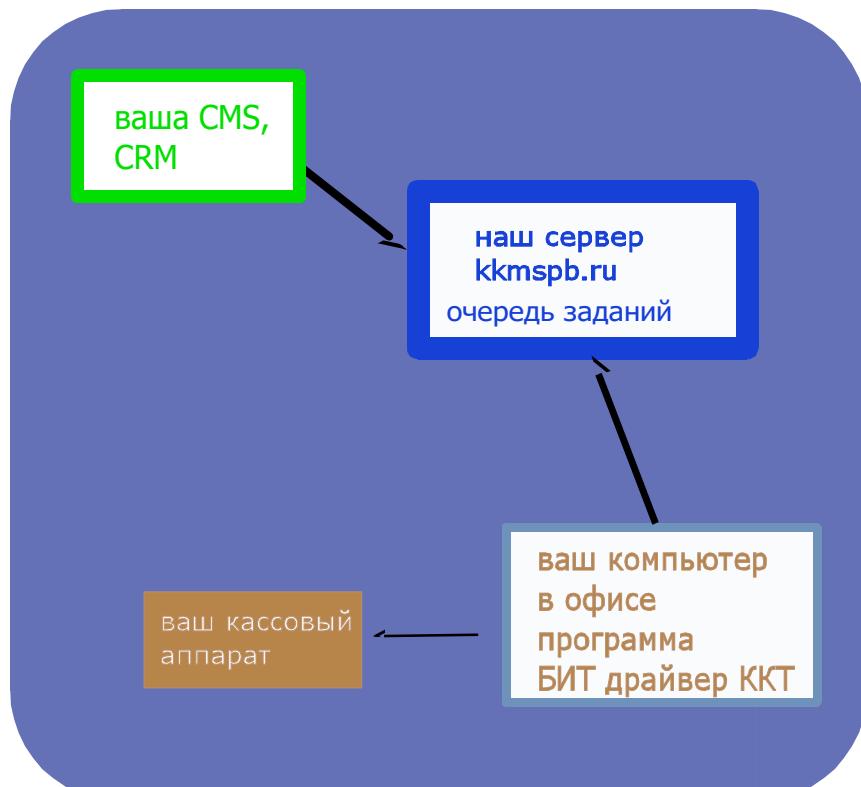
Если оплата происходит в облаке, то есть автоматически без участия продавца (например через платежный сервис, прикрученный к сайту), то этот режим будем называть - **режим оплаты в облаке**. Это режим когда покупатель самостоятельно оплачивает товар без участия продавца.

1.3 Режим курьера

Это режим только для пробития чеков из авшего личного кабинета на нашем сайте kkmspb.ru. Позволяет фискализировать чеки на выезде к клиенту, то есть удаленно на вашей кассе в вашем офисе.

Ниже речь только о режиме товароучетки

Общая схема работы



2 Тестовая страница

2.1 Режим товароучетки.

У нас создана тестовая страница для проверки фискализации чеков и приема банковских карт на тестовом сайте :

<https://vspbkassa.ru/api/test.php>. Это как-бы предположительно ваш сайт.

В своей облачной программе (CMS/CRM и т.д.) вы подготавливаете данные для пробития чека в формате POST запроса. Там формируется несколько общих параметров и json пакт команд на выполнение на локально подключенном оборудовании.

Вы передаете пакет команд: например на прием банковской карты по банковскому терминалу и (или) на фискализацию чека на кассовом аппарате.

Этот пакет вы передаете на <http://localhost:44735> (это компьютер пользователя вашей облачной программы) 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 всех команд.

3 Формат передаваемых данных

3.1 Пример передачи данных на localhost компьютеру:

Тело POST запроса:

```
json: [
  {
    "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
    }
}
]

```

Обязательные заголовки POST запроса.

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

XXXXXXXXXXXXXXXXXXXXXX: NN — наименование это значение токена кассового аппарата для пробития чеков и значение (NN) это номер ккт. Номер ккт пока не принципиален: это на будущее, если надо будет печатать на 2,3 ккт по состоянию готовности.

YYYYYYYYYYYYYYYYYYYYYY: NN — наименование это значение токена БАНКОВСКОГО ТЕРМИНАЛА для принятия оплат по картам. Значение (NN) это номер банк.терминала (по порядку). Номер банк.терминала пока не принципиален: это на будущее, если надо будет работать с 2,3 и более терминалами по состоянию готовности.

Вам надо передавать только хэши того оборудования, которое будете использовать к пакете команд. То есть если вам надо только принять наличные, то токен банковского терминала передавать не надо.

3.2 Единый протокол для управления разными ККТ и банковскими терминалами

Json пакет в теле POST запроса.

Обратите внимание, что передается список [...]. Это список команд на выполнение по порядку. Если команда завершится не удачно, то следующая выполняться 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 Меркурии (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 Кодирование передаваемых данных

Данные передаваемые в POST запросе кодируются автоматически при передаче через аjax запрос, то есть ничего дополнительно кодировать не надо. Вот примерно как это может выглядеть:

```
var hhh = {}

if( $('#BIT_BNK_TRM_TOKEN').val() != "" )
{
    hhh[ $('#BIT_BNK_TRM_TOKEN').val() ] = "1"
}

if( $('#BIT_KKT_TOKEN').val() != "" )
{
    hhh[ $('#BIT_KKT_TOKEN').val() ] = "1"
}

hhh['Action']='toLocalhost'
hhh['Bit_order_id']=$('#BIT_ORDER_ID').val()

data['json'] = $('#DATA').val()

$.ajax(
{
    url: url,
    method: 'POST',
    dataType: 'json',
    headers: hhh,
    data:data,
    async: false,
```

```
error: function( jqXHR , status, errorMsg)
{
},
success: function(data , status , jqXHR)
{
}
});
```

5 Программа БИТ драйвер ККТ

Конечно кто-то должен печатать чек и принимать оплаты бакновскими картами.

Вам надо скачать с сайта <https://kkmspb.ru/software/BIT-driver-KKT/download/> последнюю программы **БИТ драйвер ККТ** (версию не ниже 1.21.xxx) и установить на свой ПК.

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.

6 Скачать АПИ

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

7 Форум по АПИ и программе БИТ драйвер ККТ

<https://forum.kkmspb.ru>