

Методология подключения iККМ в Delphi

Методология подключения iККМ в Delphi	1
Пререквизиты	1
Основы работы	1
Основные функции из ikkm.pas	2
Получение состояния	2
Продажа и печать чека.....	2
Получение списка каталогов	2
Получение списка товаров.....	2
Получение списка налогов.....	2
Получение списка кассиров.....	3
Получение отчёта по сменам	3
Получение отчёта по чекам	4
Получение отчёта по продажам.....	4
Получение состояния чека.....	4
Список возможных ошибок	4

Пререквизиты

Для корректной работы с iККМ Вам потребуется:

1. Delphi версии 7 и старше, **кроме Delphi 8***.
2. Установленный набор компонентов Indy (Internet Direct), в частности, TidHTTP.
3. Добавленный в Ваш проект модуль ikkm.pas из папки /dist.

* теоретически ikkm.pas должен работать и в нём, просто он не был протестирован.

Основы работы

Взаимодействие с iККМ в режиме веб-кассы происходит посредством HTTP-запросов (в нашем случае — POST-запросов), с использованием описанной в **ikkm.pas** функции **http_post_request**.

Также в модуле **ikkm.pas** уже описаны основные функции для работы с iККМ: проверка состояния, печать чека и прочие (подробнее в разделе «Функции»). Некоторые из функций требуют определённого уровня доступа пользователя, который запустил веб-кассу (Страший кассир, Инспектор, пр.).

Для простоты понимания в папке **/dist** также представлен **iККМ_test.exe**, наглядно отображающий работу с iККМ.

Основные функции из ikkm.pas

Все описанные функции возвращают **строковые** значения и обязательно принимают параметры: ip-адрес iKKM, порт iKKM, ключ API. Все параметры также имеют **строковый** тип.

Получение состояния

check_state (ip, port, api_key: string): string;

Если ответ равен нулю, запрос выполнен удачно, iKKM готов к работе. Иначе смотрите раздел «Список возможных ошибок».

Продажа и печать чека

sale (ip, port, cost, income_cash, api_key: string; string_to_print: string = "; count_of_chars: byte = 32): string;

Параметры **cost**, **income_cash** являются строковыми представлениями чисел — цена покупки и сумма оплаты соответственно. Параметр **string_to_print** содержит строку, которая будет напечатана на чеке, **count_of_chars** — число символов из параметра string_to_print в одной строке на чеке (по умолчанию равно 32).

В случае успешного выполнения функция вернёт новый ключ API.

Получение списка каталогов

get_catalogs (ip, port, api_key: string): string;

Функция возвращает JSON-объект вида:

```
{
  "1": "Кухня",
  "2": "Бар",
  "3": "Салаты",
  ...
}
```

пример ответа

Получение списка товаров

get_items (ip, port, api_key: string): string;

Функция возвращает JSON-объект вида:

```
[[
  "uid": 11, # внутренний и уникальный код товара
  "code1": 888, # уникальный код товара
  "code2": "008123563127", # штрих-код
  "name": "Кофе", # наименование товара
  "price1": 77, # розничная цена
  "itemtax1": 2, # код налога (см справочник налоги)
  "itemtype": 0, # тип
  "measure": 0, # измерение
  "catalogid": 16 # код каталога (см справочник каталог)
],...
]]
```

пример ответа

Получение списка налогов

get_taxes (ip, port, api_key: string): string;

Функция возвращает JSON-объект вида:

```
{
  "1": "НДС",
  "2": "Без НДС",
  ...
}
```

пример ответа

Получение списка кассиров

get_cashiers (ip, port, api_key: string): string;

Функция возвращает JSON-объект вида:

```
{
  "100": "Алия Кассир",
  "201": "Наташа Старший Кассир",
  ...
}
```

пример ответа

Получение отчёта по сменам

get_shifts (ip, port, api_key: string): string;

Функция возвращает JSON-объект вида:

```
[{
  "id": 12, # Номер смены
  "opened": 0, # 1 - смена открыта, 0 - смена закрыта
  "openDate": "2016-01-18 20:41:06", # дата открытия смены
  "closeDate": "2016-01-19 17:21:46", # дата закрытия смены
  "cashierId": 166, # код кассира
  "sale": 5210, # сумма всех продаж на начало смены
  "buy": 100, # сумма всех покупок на начало смены
  "saleReturn": 803, # сумма всех возвратов продаж на начало смены
  "buyReturn": 50 # сумма всех возвратов покупок на начало смены
},...
]
```

пример ответа

Получение отчёта по чекам

get_cheques (ip, port, no, api_key: string): string;

Функция возвращает JSON-объект вида:

```
[[
  "chequeId": 244, # номер чека
  "chequeDate": "2016-01-18 20:41:08", # дата открытия чека
  "chequeType": "1", # 1 - продажа, 2 - покупка, 3 - возврат продажи, 4 - возврат покупки
  "chequeSales": [[
    "id": 878, # код продажи
    "itemCode": "1001", # код товара (может быть пустой)
    "qty": 1, # количество (может быть Double)
    "price": 23, # цена за единицу
    "saleSum": 23, # сумма продажи
    "taxId": 2 # код налога
  ],...
],...
]]
```

пример ответа

Получение отчёта по продажам

get_sales (ip, port, no, api_key: string): string;

Функция возвращает JSON-объект вида:

```
[[
  "id": 878, # код продажи
  "chequeId": 244, # номер чека, сортировка результата по этому полю
  "itemCode": "1001", # код товара (может быть пустой)
  "qty": 2, # количество (может быть Double)
  "price": 30, # цена за единицу
  "saleSum": 60, # сумма продажи
  "taxId": 2 # код налога
],...]
```

пример ответа

Получение состояния чека

get_cheque (ip, port, cheque_api_key, api_key: string): string;

Параметр **cheque_api_key** — ключ API, с которым была выполнена операция.

Функция возвращает JSON-объект вида:

```
{
  "docid": "247", # номер чека
  "createDate": "2016-01-20 18:10:43", # дата создания
  "submitDate": "2016-01-20 18:11:12", # дата отправки
  "status": 1, # 0 - не отправлен в ОФД, 1 - отправлен
  "onlineCode": "020223112" # онлайн номер чека (может быть пустым)
}
```

пример ответа

Список возможных ошибок

В случае получения ошибки **HTTP 303** следует проверить верность пути запроса, или передаваемых параметров, в частности, ключа API.

В случае ошибки **HTTP 400** функция **http_post_request** возвращает одно из следующих описаний ошибки:

Ответ iККМ	Описание	Действия
- 1	неверный web-api ключ	настроить новый ключ на iККМ
- 2	происходит регистрация или банковская операция	повторите запрос позже
- 3	смена превысила 24 часа	необходимо закрыть смену (выполняется на iККМ)
- 4	оффлайн период более 72 часов	решить проблему связи и разблокировать ккм
- 5	низкий заряд батареи ккм	зарядить ккм
- 6	принтер не готов	проверить бумагу
- 7	ошибка запроса	проверьте передаваемые параметры
- 8	ошибка метода	только /api или /apicheck
- 9	ошибка значений параметров	передавать только цифры (кроме параметра print)
- 10	сумма cash (bank, tara, discount) меньше sale (buy...)	проверьте логику работы с API
- 11	смена открыта другим кассиром	закрыть смену
- 12	значение cash меньше чем расчетная итоговая сумма	может возникнуть если налог насчитывается поверх суммы
- 13	банк или тара или скидка больше чем итоговая сумма	проверьте логику работы с API
- 14	ошибка налога	указанный налог не найден
- 15	нет наличности в кассе (возврат продажи, покупка)	проверьте передаваемые параметры
- 16	операцию может проводить только старший кассир	проверить пользователя
- 99	ошибка iККМ	смотрите журнал событий на iККМ в меню состояние устройства
- 100 .. - 199	ккм заблокирован ОФД	см документацию ОФД (отнять100)