

# Описание методов (old)

## Запрос данных аккаунта (публичный)

GET

/api/lite/pub/account\_info

Документация обновляется.

## Получение информации по предзаказу (публичный)

GET

/api/lite/pub/order

Документация обновляется.

## Создание / обновление товарного предложения

POST

/api/lite/offer

создает или обновляет (если задан sku.humanID) товарное предложение

## Запрос:

```
{
  "sku": {
    "id": "12345", - опциональный параметр, только для обновления существующего
```

предложения. значение генерируется при создании предложения.

```
        "shortName": "краткое наименование",
        "fullName": "полное наименование",
        "article": "234234",
        "description": "описание"
    },
    "price": 33.45,
    "discountPrice": 20,
    "weight": 357, - единица измерения: грамм
    "dimensions": {
        "height": 12, - единица измерения: сантиметр
        "width": 20, - единица измерения: сантиметр
        "depth": 8 - единица измерения: сантиметр
    },
    "enabled": true,
    "pos": 123.456
}
```

В ответ возвращается

## Успех:

```
{
  "success": true,

  "humanId": "12345"
}
```

## Ошибка:

/api/lite/offer/pos

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "Текст ошибки"
    }
  ]
}
```

```
}  
]  
}
```

## Обновление позиции товарного предложения

**POST**

**/api/lite/offer/pos**

Обновляет значение позиции для сортировки предложений в списке

### Запрос:

```
{  
  "id": "1002",  
  "pos": 123.456  
}
```

### Ответ:

```
{  
  "success": true  
}
```

## Создание картинки к товарному

# предложению

POST

/api/lite/offer/image

Загружает на сервер картинку и привязывает изображение к товару.

Используется заголовок запроса **Content-type: multipart/form-data**.

В качестве параметров запроса передается:

- id - идентификатор товара
- upfile - картинка

На стороне сервера:

- Сохраняется оригинал картинки (но с ограничением в максимальное количество пикселей) - original
- Сохраняется обрезанное изображение 1x1 (ближе к центру) - cropper
- Сохраняется миниатюра обрезанного изображения - thumbnail

В ответ возвращается информация о загруженной картинке:

## Успех:

```
{
  "success": true,
  "image": {
    "orig": "https://img.kak2c.ru/g/d/123abc.jpg",
    "crop": "https://img.kak2c.ru/g/d/123abc_crop.jpg",
    "prev": "https://img.kak2c.ru/g/d/123abc_p.jpg",
    "id": "6F9619FF-8B86-D011-B42D-00CF4FC964FF",
    "pos": 123.456,
    "crop_p": [
      10,
      500,
      400,
      150
    ]
  }
}
```

Параметры:

- orig - прямая ссылка на оригинал картинки
- crop - прямая ссылка на кропнутое изображение
- prev - прямая ссылка
- id - идентификатор картинки (для последующего обновления или удаления)
- crop\_p - координаты кропнутого изображения относительно оригинала (topLeftX, topLeftY, bottomRightX, bottomRightY)

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "Текст ошибки"
    }
  ]
}
```

# Загрузка картинки без привязки к предложению

POST

/api/lite/image

Загружает на сервер картинку в исходных размерах без сжатия.

Используется заголовок запроса **Content-type: multipart/form-data**.

В качестве параметров запроса передается:

- upfile - картинка

На стороне сервера:

- Сохраняется оригинал картинки (без ограничений) - original

В ответ возвращается информация о загруженной картинке:

## Успех:

```
{
  "success": true,
  "orig": "https://img.kak2c.ru/g/d/123abc.jpg"
}
```

Параметры:

- orig - прямая ссылка на оригинал картинки

## ошибка:

```
{
  "success": false,
  "errors": [
    {
```

```
"code": 0,  
"message": "Текст ошибки"  
}  
]  
}
```

## Обновление кроя картинки к товарному предложению

**POST**

**/api/lite/offer/image/crop**

Обновляет параметры кроя (обрезки) у картинки товарного предложения.

### Запрос:

```
{  
  "id": "6F9619FF- 8B86- D011- B42D- 00CF4FC964FF",  
  "crop_p": [  
    10,  
    500,  
    400,  
    150  
  ]  
}
```

В ответ возвращается результат операции, id обновленной картинки и новый url на кропнутое изображение и превью.

### Пример успешного ответа:

```
{
```

```
"success": true,
"id": "6F9619FF-8B86-D011-B42D-00CF4FC964FF",
"crop": "https://img.kak2c.ru/g/d/123abc_cropped.jpg",
"prev": "https://img.kak2c.ru/g/d/123abc_preview.jpg"
}
```

## В случае ошибки:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "Картинки с указанным id не существует"
    }
  ]
}
```

## Удаление картинки у товарного предложения

**DELETE**

**/api/lite/offer/image**

Для удаления метки, выполняется следующий POST запрос с указанием идентификатора картинки.

## Пример запроса:

```
{
  "id": "6F9619FF-8B86-D011-B42D-00CF4FC964FF"
}
```

В ответ возвращается результат операции и id удаленной картинки.

## Пример успешного ответа:

```
{
  "success": true,
  "id": "6F9619FF-8B86-D011-B42D-00CF4FC964FF"
}
```

## В случае ошибки:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "Картинки с указанным id не существует"
    }
  ]
}
```

## Обновление позиции картинки

**POST**

**/api/lite/offer/image/pos**

Обновляет значение позиции для сортировки картинок в списке

## Запрос:

```
{
  "id": "6F9619FF-8B86-D011-B42D-00CF4FC964FF",
  "pos": 123.456
}
```

## Ответ:

```
{
  "success": true
}
```

## В случае ошибки:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

# Создание/обновление заказа

**POST**

**/api/lite/order**

Создает новый заказ в системе Кактус или обновляет существующий.

Доставка выгружается как позиция в items (id = "delivery").

Цены позиций задаются в запросе.

## Пример запроса:

```
{
  "order": {
    "comment": "",
    "confirmStatus": "NEED_CONFIRM",
    "paymentStatus": "NOT_PAID",
    "paymentMethodCode": "online",
    "totalOrderSum": 1212,
    "delivery": {
      "receiver": {
        "name": "Олег",
        "surname": "Коробов",
        "patronymic": "",
        "phone": "9153112255",
        "alternativePhone": "",
        "email": "korobov.oleg@gmail.com"
      },
      "deliveryCode": "courier",
      "timeFrom": "10:30",
      "timeTo": "20:00",
      "desiredDeliveryDate": "2018-10-18",
      "deliveryComment": "",
      "address": {
        "regionFias": "18133adf-90c2-438e-88c4-62c41656de70",
        "areaFias": "b502ae45-897e-4b6f-9776-6ff49740b537",
        "cityFias": "",
        "settlementFias": "",
        "country": "RU",
        "region": "Москва",
        "area": "",
        "city": "Москва",
        "settlement": "",
        "street": "Боровское ш.",
        "house": "20",
```

```
        "block": "",
        "building": "",
        "flat": "120",
        "fullAddress": "",
        "zip": ""
    },
    "pickupPointId": ""
},
"items": [
    {
        "num": 1,
        "id": "48488",
        "quantity": 1,
        "price": 1212
    }
]
}
}
```

В ответ возвращается

## успех:

```
{
  "success": true,

  "orderId": "12345"

}
```

## ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
```

```
"message": "Текст ошибки"
}
]
```

## Заказ → обновление контактной информации получателя

POST

/api/lite/order/receiver

Возможность обновить только контактную информацию

### Запрос:

```
{
  "orderId": "1234",
  "name": "Олег",
  "surname": "Коробов",
  "patronymic": "",
  "phone": "9153112255",
  "alternativePhone": "",
  "email": "korobov.oleg@gmail.com"
}
```

В ответ возвращается

### успех:

```
{
  "success": true
}
```

```
}
```

## ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "Текст ошибки"
    }
  ]
}
```

## Заказ → обновление товарной части

POST

/api/lite/order/items

Возможность обновить только табличную часть заказа

Необходимо в запросе передать новое желаемое состояние табличной части в виде непустого перечня позиций

## Запрос:

```
{
  "orderId": "1234",
  "items": [
    {
      "num": 1,
      "id": "48488",
      "quantity": 1,
      "price": 1212
    }
  ]
}
```

```
}  
]  
  
}
```

В ответ возвращается

## успех:

```
{  
  "success": true  
  
}
```

## ошибка:

```
{  
  "success": false,  
  "errors": [  
    {  
      "code": 0,  
      "message": "Текст ошибки"  
    }  
  ]  
}
```

## Заказ → отменить

POST

/api/lite/order/cancelled

Производит отмену заказа.

## Пример запроса

```
{  
  "orderId": "1029"  
}
```

В ответ возвращается

### успех:

```
{  
  "success": true  
  
}
```

### ошибка:

```
{  
  "success": false,  
  "errors": [  
    {  
      "code": 0,  
      "message": "Текст ошибки"  
    }  
  ]  
}
```

**Заказ → пометить оплаченным  
получателем**

**POST****/api/lite/order/payment**

Ставит статус оплаты покупателем.

## Пример запроса

```
{  
  "orderId": "1029"  
}
```

В ответ возвращается

## успех:

```
{  
  "success": true  
}
```

## ошибка:

```
{  
  "success": false,  
  "errors": [  
    {  
      "code": 0,  
      "message": "Текст ошибки"  
    }  
  ]  
}
```

# Заказ → снять пометку оплаты заказа получателем

DELETE

/api/lite/order/payment

Снимает статус оплаты покупателем.

## Пример запроса

```
{  
  "orderId": "1029"  
}
```

В ответ возвращается

## успех:

```
{  
  "success": true  
}
```

## ошибка:

```
{  
  "success": false,  
  "errors": [  
    {  
      "code": 0,  
      "message": "Текст ошибки"  
    }  
  ]  
}
```

```
}  
]  
}
```

## Пример ошибки:

Документация обновляется.

# Заказ → получить акт приема передачи

POST

/api/lite/order/takeout\_list

Получение печатной формы в pdf для указанного списка заказов

Печатная форма возвращается в формате base64.

## Пример ошибки:

```
{  
  "orders": ["1001", "1002", "1003"]  
}
```

## Успех:

```
{  
  "success": true,  
  "content" : "base64data"  
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "Текст ошибки"
    }
  ]
}
```

## Оплатить доставку заказа/заказов

POST

**/api/lite/order\_delivery\_payment**

Производит списание средств за доставку указанных заказов.

После успешного списания - создает заявку на сбор курьером на указанную дату. Если заявка уже есть - новая не создается. Если на указанную дату заявку оформить уже нельзя (опоздал) - автоматом создается заявка на ближайшую возможную дату.

```
{
  "orders": ["1005", "1008"],
  "date": "2018-01-01", - обязательные для всех способов отгрузки кроме Своих курьеров
  "timeFrom": "09:00", - см выше
  "timeTo": "14:00", - см выше
  "courierId": "aa23s" - обязательный только для способа отгрузки Свои курьеры
}
```

## Успех:

Возвращается дата и интервал запрошенного приезда курьера.

```
{
  "success": true,
  "date": "2018-01-01",
  "timeFrom": "09:00",
```

```
"timeTo": "14:00"
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

## Отменить оплату доставки заказа/заказов

**DELETE**

**/api/lite/order\_delivery\_payment**

Пока не реализуем этот метод.

Вариант:

Производит отмену списания с баланса доставки указанных заказов, если по ним не было статусов от GD по сбору. Если какие то статусы по заказу поступали от GD - значит они уже забрали груз - тогда все заказы данного реестра отгрузки автоматом помечаются отгруженными и вернуть сумму по ним нельзя.

## Оформить вызов курьера

**POST**

**/api/lite/courier\_call**

Создает заявку на сбор курьером на указанную дату. Если заявка уже есть - возвращает

ошибку, что заявка уже есть.

## Пример ошибки:

```
{
  "date": "2018-01-01",
  "timeFrom": "09:00",
  "timeTo": "14:00"
}
```

## Успех:

```
{
  "success": true
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

# Отменить вызов курьера

DELETE

/api/lite/courier\_call

Отменяет заявку на вызов курьера, если заявка на эту дату и интервал есть. если заявки нет - возвращает ошибку.

## Пример ошибки:

```
{
  "date": "2018-01-01",
  "timeFrom": "09:00",
  "timeTo": "14:00"
}
```

## Успех:

```
{
  "success": true
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

# Получить список транзакций

GET

/api/lite/transaction

Получение списка транзакций баланса аккаунта.

## Пример ошибки:

```
{
  "transactions": [
    {
      "date": "2019-03-01 17:53:23.538",
      "sum": 1500.50,
      "documentId": "1003",
      "documentType": "PAYMENT",
      "description": "Пополнение баланса по счету", // из справочника
      "orderId": null
    },
    {
      "date": "2019-03-02 19:57:23.538",
      "sum": -265.57,
      "documentId": "1050",
      "documentType": "ORDER_PAYMENT_DOC_FACT",
      "description": "Оплата доставки заказа 1024 (доставка)", // в скобках -
      "orderId": "1024"
    }
  ]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

## Получить список транзакций по заказу

GET

**/api/lite/order/transaction**

Получение списка транзакций баланса оплаты доставки и услуг указанного заказа

запрос должен содержать параметр orderId с номером заказа, например

?orderId=1020

## Результат запроса:

```
{
  "transactions": [
    {
      "date": "2019-03-02 19:57:23.538",
      "sum": -265.57,
      "documentId": "1050",
      "documentType": "ORDER_EXPENSE_PAYMENT_DOC",
      "description": "Списание по расходам на оплату услуг заказа 1020 (доставка)",
      "orderId": "1020"
    },
    {
```

```
    "date": "2019-03-02 19:57:23.538",
    "sum": 265.57,
    "documentId": "1052",
    "documentType": "ORDER_PAYMENT_DOC_FACT",
    "description": "Оплата доставки заказа 1020 (доставка)",
    "orderId": "1020"
  }
]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

# Создать счет на оплату и получить информацию для оплаты

POST

/api/lite/payment

## Запрос:

```
{  
  "sum" : "222"  
}
```

## Ответ:

```
{  
  "success": true,  
  "errors": [],  
  "redirectData": {  
    "data": {  
      "entries": [  
        {  
          "num": 0,  
          "name": "MERCHANT",  
          "value": "kaktwocr"  
        },  
        {  
          "num": 1,  
          "name": "ORDER_REF",  
          "value": "25567175-8d84-e371-90fa-8a6e86a69aa4"  
        },  
        {  
          "num": 2,  
          "name": "ORDER_DATE",  
          "value": "2019-03-11 17:09:35"  
        },  
        {  
          "num": 3,  
          "name": "ORDER_PNAME[ ]",  
          "value": "Пополнение баланса аккаунта №6419855"  
        },  
        {  
          "num": 4,  
          "name": "ORDER_PCODE[ ]",  
          "value": "balance"  
        },  
        {
```

```
    "num": 5,
    "name": "ORDER_PRICE[ ]",
    "value": "222.00"
  },
  {
    "num": 6,
    "name": "ORDER_QTY[ ]",
    "value": "1"
  },
  {
    "num": 7,
    "name": "ORDER_VAT[ ]",
    "value": "0"
  },
  {
    "num": 8,
    "name": "PRICES_CURRENCY",
    "value": "RUB"
  },
  {
    "num": 9,
    "name": "DESTINATION_CITY",
    "value": ""
  },
  {
    "num": 10,
    "name": "DESTINATION_STATE",
    "value": ""
  },
  {
    "num": 11,
    "name": "DESTINATION_COUNTRY",
    "value": ""
  },
  {
    "num": 12,
    "name": "PAY_METHOD",
    "value": "CCVISAMC"
  },
  {
    "num": 13,
```

```
    "name": "ORDER_PRICE_TYPE[ ]",
    "value": "GROSS"
  },
  {
    "num": 14,
    "name": "ORDER_HASH",
    "value": "6fe466d056fcc82836a861c322070381"
  },
  {
    "num": 15,
    "name": "BILL_FNAME",
    "value": "Петр"
  },
  {
    "num": 16,
    "name": "BILL_LNAME",
    "value": "Петров"
  },
  {
    "num": 17,
    "name": "BILL_EMAIL",
    "value": "abc3@abc.ru"
  },
  {
    "num": 18,
    "name": "BILL_PHONE",
    "value": "111111"
  },
  {
    "num": 19,
    "name": "BILL_COUNTRYCODE",
    "value": "RU"
  },
  {
    "num": 20,
    "name": "BACK_REF",
    "value": ""
  },
  {
    "num": 21,
    "name": "LANGUAGE",
```

```

        "value": "RU"
    },
    {
        "num": 22,
        "name": "CURRENCY",
        "value": "RUB"
    }
],
"url": "https://sandbox.payu.ru/order/lu.php"
},
"paymentProvider": "PAYU"
}
}

```

Для работы в песочнице PayU использовать карты из <https://secure.payu.ru/docs/integration-sandbox/ru/>

## Пример формы, которую нужно сгенерировать на клиенте:

```

<html>
  <head></head>
  <body>
    <form name="example" method="POST" action="https://sandbox.payu.ru/order/lu.php">
      <input type="hidden" name="MERCHANT" value="kaktwocr">
      <input type="hidden" name="ORDER_REF" value="25567175-8d84-e371-90fa-
8a6e86a69aa4">
      <input type="hidden" name="ORDER_DATE" value="2019-03-11 17:09:35">
      <input type="hidden" name="ORDER_PNAME[]" value="Пополнение баланса аккаунта
И6419855">
      <input type="hidden" name="ORDER_PCODE[]" value="balance">
      <input type="hidden" name="ORDER_PRICE[]" value="222.00">
      <input type="hidden" name="ORDER_QTY[]" value="1">
      <input type="hidden" name="ORDER_VAT[]" value="0">
      <input type="hidden" name="PRICES_CURRENCY" value="RUB">
      <input type="hidden" name="DESTINATION_CITY" value="">
    </form>
  </body>
</html>

```

```
<input type="hidden" name="DESTINATION_STATE" value="">
<input type="hidden" name="DESTINATION_COUNTRY" value="">
<input type="hidden" name="PAY_METHOD" value="CCVISAMC">
<input type="hidden" name="ORDER_PRICE_TYPE[]" value="GROSS">
<input type="hidden" name="ORDER_HASH" value="6fe466d056fcc82836a861c322070381">
<input type="hidden" name="BILL_FNAME" value="Петр">
<input type="hidden" name="BILL_LNAME" value="Петров">
<input type="hidden" name="BILL_EMAIL" value="abc3@abc.ru">
<input type="hidden" name="BILL_PHONE" value="111111">
<input type="hidden" name="BILL_COUNTRYCODE" value="RU">
<input type="hidden" name="BACK_REF" value="">
<input type="hidden" name="LANGUAGE" value="RU">
<input type="hidden" name="CURRENCY" value="RUB">

<button type="submit">Оплатить</button>

</form>

</body>

</html>
```

## Получить список своих курьеров

GET

/api/lite/own\_courier

Получение списка своих курьеров.

### Пример ошибки:

```
{
  "couriers": [
    {
      "id": "123456abc",
      "fullName": "Василий",
      "phone": "+7(901)1122121"
    },
    {
      "id": "123457abc",
      "fullName": "Петр",
      "phone": "+7(901)1122122"
    }
  ]
}
```

```
]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

# Создать/обновить своего курьера

POST

/api/lite/own\_courier

Создание или обновление (если задан id) своего курьера

## Пример запроса:

```
{
  "id": "123456abc",
  "fullName": "Василий",
  "phone": "+7(901)1122121"
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
```

```
{
  "code": 0,
  "message": "текст ошибки"
}
]
```

## Удаление курьера

**DELETE****/api/lite/own\_courier**

Для удаления курьера, выполняется следующий POST запрос с указанием идентификатора курьера.

### Пример запроса:

```
{
  "id": "6F9619FF-8B86-D011-B42D-00CF4FC964FF"
}
```

В ответ возвращается результат операции

### Пример успешного ответа:

```
{
  "success": true
}
```

### В случае ошибки:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "курьера с указанным id не существует"
    }
  ]
}
```

```
}  
]  
}
```

## Заказ - отметить доставку

POST

/api/lite/delivery\_result

Для доставки своими курьерами - простановка результата доставки: доставлен/отказ/утеря.

### Пример запроса:

```
{  
  "orderId": "123456abc",  
  "result": "DELIVERED", //LOST, RETURNING, RETURNED  
  "comment": "abcd"  
}
```

### Ошибка:

```
{  
  "success": false,  
  "errors": [  
    {  
      "code": 0,  
      "message": "текст ошибки"  
    }  
  ]  
}
```

### Ответ:

Документация обновляется.

## Пример ошибки:

Документация обновляется.

# Записать данные собственного аккаунта Робокасса

POST

/api/lite/settings/payment/robokassa

Записать данные аккаунта робокасса для прямого зачисления оплат заказов покупателей.

## Пример запроса:

```
{
  "merchantId": "abcd",
  "accountSecret": "abcd",
  "accountSecret2": "abcd"
}
```

## Результат запроса:

```
{
  "success": true,
  "accountId": "abcd",
  "resultUrl": "abcd", - основной адрес приема ipn уведомлений (только оплаченные)
  "failUrl": "abcd", - адрес приема ipn уведомлений (только неоплаченные/отказ покупателя)
  "successesUrl": "abcd" - адрес редиректа в магазин после оплаты
}
```

## Ошибка:

Документация обновляется.

## Получить макет витрины(публичный)

GET

/api/lite/pub/settings/template

Получение макета витрины магазина аккаунта, в формате, предназначенном для ui витрины

## Результат запроса:

```
{
  "templates" : [
    {
      "id": "123456",
      "isDefault": true,
      "htmlContent": "<html>..."
    }
  ]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

```
}
```

# Получить макеты витрин

**GET****/api/lite/settings/template**

Получение перечня макетов витрин магазинов аккаунта, в формате, предназначенном для ui конструктора витрины

Для получения одного макета по id - использовать параметр в url вида ?id=123456

## Результат запроса:

```
{
  "templates" : [
    {
      "id": "123456",
      "isDefault": true,
      "content": "contentData..."
    }
  ]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

# Создать/обновить макет витрины

POST

/api/lite/settings/template

Создание/обновление данных о макете витрины в формате для ui конструктора

## Пример запроса:

```
{
  "id": "123456", // при наличии параметра - обновляет существующий макет по его id
  "isDefault": true,
  "content": "contentData...",
  "htmlContent": "html content Data"
}
```

## Результат запроса:

```
{
  "success": true,
  "id": "123456"
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

```
]
}
```

## Удалить макет витрины

**DELETE****/api/lite/settings/template**

Удаление одного макета. Использовать параметр в url вида `?id=123456`

### Результат запроса:

```
{
  "success": true
}
```

### Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

## Получить общие макеты витрин

**GET****/api/lite/settings/template/general**

Получение перечня стандартных макетов витрин магазинов в формате, предназначенном для ui конструктора витрины

Для получения одного макета по id - использовать параметр в url вида `?id=123456`

## Результат запроса:

```
{
  "templates" : [
    {
      "id": "123456",
      "isDefault": true,
      "content": "contentData..."
    }
  ]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

## Получить общие шаблоны блоков витрин

**GET****/api/lite/settings/template/block**

Получение перечня стандартных блоков витрин магазинов в формате, предназначенном для `ui` конструктора витрины

Для получения одного блока по `id` - использовать параметр в url вида `?id=123456`

## Результат запроса:

```
{
  "blocks": [
    {
      "id": "f4173840-f409-424d-a000-24542b391c5d",
      "name": "block name",
      "content": "contentData..."
    }
  ]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

## Получить заявки на сбор

**GET****/api/lite/gather**

Получение списка заявок на сбор, сортировка - свежие по полю date - сначала

## Результат запроса:

```
{
  "gathers": [
    {
      "success": true,
      "comment": null,
      "id": "1000-GD",
      "date": "2019-03-07 12:37:18",
      "desiredDate": "2019-03-15",
      "desiredHourStart": "09:00",
      "desiredHourEnd": "14:00",
      "status": "AWAITING_CARGO",    //CREATED, CARGO_SHIPPED, HOLD, CANCELED
      "trackingNumber": "100039439"
    }
  ]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

# Получить варианты курьерских пакетов

GET

/api/lite/pub/courier\_pack

Получение списка доступных типов курьерских пакетов с размерами

## Результат запроса:

```
{
  "courier_packs": [
    {
      "success": true,
      "type": "PACKET", // PAPERBOX
      "sizeType": "S",  // M/L
      "cargoWeight": 1500, // максимальный вес содержимого в граммах
      "dimensions": {
        "height": 12,
        "width": 20,
        "depth": 8
      }
    }
  ]
}
```

## Ошибка:

```
{
  "success": false,
  "errors": [
    {
      "code": 0,
      "message": "текст ошибки"
    }
  ]
}
```

## Создать пользователя для Expert

GET

**/api/lite/settings/go-expert**

Метод только для админов системы. Генерирует для аккаунта Lite нового пользователя чтобы под ним можно было работать в интерфейсе Expert.

Логин равен логину исходного аккаунта (тенанта) + суффикс "-expert", например:

**manager@shop.ru-expert**

Пароль равен паролю исходного аккаунта. При смене пароля исходного пользователя - expert логин останется с прошлым паролем.

Параметр URL: **?c=123** - проверочный код.

## Результат запроса:

```
{
  "success": true
}
```

## Ошибка:

```
{
```

```
"success": false,  
"errors": [  
  {  
    "code": 0,  
    "message": "текст ошибки"  
  }  
]  
}
```