

REST API

Общее

REST API используется для взаимодействия терминалов с товароучетной системой посредством **HTTP** запросов.

Для взаимодействия терминалов сбора данных с товароучетной системой на стороне последней должен быть реализован интерфейс **Rest API** для принятия **HTTP** запросов. Для ограничения доступа необходимо использовать обычную **Basic** авторизацию - пара логин:пароль закодированная в **Base64** в заголовке запроса **Authorization**.

Все методы интерфейса в ответ на запрос возвращают структуру **JSON**:

- meta - описание метаданных
- type - тип ответа
- mediaType - тип данных, содержащихся в реквизите media
- media - данные ответа

Интерфейс использует следующие коды ответа:

Код ответа	Значение
200	Запрос успешно обработан
201	Запрос успешно обработан, данных для формирования ответа не найдено
403	Недостаточно прав для выполнения запроса
404	Запрашиваемые данные отсутствуют
500	Запрос имеет некорректную структуру

Точки подключения (endpoint) интерфейсов

HTTP сервис товароучетной системы должен обеспечивать следующие интерфейсы:

- [интерфейс рукопожатия](#) - для знакомства терминала и товароучетной системы. Терминал сбора данных обращается к интерфейсу рукопожатия каждый раз при запуске бизнес-процесса. При первом соединении в товароучетной системе сохраняются имя и идентификатор терминала для последующего взаимодействия. При последующих соединениях идет проверка наличия данного терминала в базе данных товароучетной системы и внесения изменений о терминале (при необходимости).
- [интерфейс получения задач для бизнес-процесса](#) - получение задач (одной или нескольких) из товароучетной системы в терминал.
- [интерфейс передачи результата выполнения задачи для бизнес-процесса](#) - передача задачи из терминала в товароучетную систему.

Интерфейс рукопожатия

GET handshake/{device_id}

URI параметры

- `device_id (required)` идентификатор терминала сбора данных.

Пример: 7737f6da-5fcf-4752-9cc1-dd268242612e

Параметры запроса

- `device_desc (required)`: человеко-читаемое наименование терминала сбора данных, установленное пользователем при регистрации терминала в облачной инфраструктуре

Пример:

```
curl -X GET  
"http://195.54.41.203/rtl/hs/alpha/handshake/7737f6da-5fcf-4752-9cc1-  
dd268242612e?device_desc=ТЦД_ChiperLab_RK25"  
-H "Authorization: Basic <Credentials>"
```

В случае успеха возвращает **JSON** файл с полем:

`success`: успешное выполнение

Пример JSON файла:

```
{
  "meta": {
    "type": "result",
    "mediaType": "application/json"
  },
  "media": {
    "success": true
  }
}
```

В случае неудачи, возвращается сообщение об ошибке.

Пример:

```
{
  "meta": {
    "type": "error",
    "mediaType": "text/plain"
  },
  "media": "Некорректная структура данных."
}
```

Интерфейс получения задач для бизнес-процесса

GET tasks/{device_id}/{task_type}

URL параметры:

- **device_id**: идентификатор терминала;
- **task_type**: идентификатор типа задачи в бизнес-процессе - произвольный текстовый параметр, который задается и на стороне терминала, и на стороне товароучетной системы.

Метод вызывается инструкцией **net.get.task**.

Пример:

```
curl -X GET
"http://195.54.41.203/rtl/hs/alpha/tasks/7737f6da-5fcf-4752-9cc1-dd268242612e/Inventorization"
-H "Authorization: Basic <Credentials>"
```

В случае успеха возвращает **JSON** файл со следующими полями:

erase (boolean): требуется ли предварительная очистка базы данных

barcodes: описание штрих-кодов товаров и ячеек. Каждый штрих-код представляет собой словарь, содержащий следующие поля:

- **uuid (string)**: идентификатор
- **barcode (string)**: штрих-код
- **product (string)**: идентификатор продукта
- **characteristic (string)**: идентификатор параметров продукта
- **cell (string)**: идентификатор ячейки

products: описание товаров. Каждый продукт представляет собой словарь, содержащий следующие поля:

- **uuid (string)**: идентификатор
- **code (string)**: код
- **article (string)**: артикул продукта
- **description (string)**: наименование продукта
- **marked (boolean)**: признак маркированного товара
- **price (double)**: цена продукта
- **rest (double)**: остаток на складе

characteristics: описание характеристик товаров. Каждая характеристика представляет собой словарь, содержащий следующие поля:

- **uuid (string)**: идентификатор
- **product (string)**: идентификатор продукта
- **description (string)**: описание характеристики

header (object): описание заголовков документов. Каждый заголовок представляет собой словарь, содержащий следующие поля:

- **uuid (string)**: идентификатор документа
- **type (string)**: тип документа
- **number (string)**: номер документа
- **date (string)**: дата документа в формате yyyy-MM-dd'T'HH:mm:ss
- **comment (string)**: комментарий к документу
- **author (string)**: автор документа

content (array): описание строк документа. Каждая строка представляет собой словарь, содержащий следующие поля:

- **uuid (string)**: идентификатор строки
- **index (int)**: номер строки

- `owner (string)`: идентификатор заголовка документа
- `product (string)`: идентификатор товара
- `characteristic (string)`: идентификатор характеристики
- `quantity (double)`: количество по строке
- `barcodes (array)`: штрих-коды по строке. Каждый штрих-код представляет собой словарь, содержащий следующие поля:
- `uuid (string)`: идентификатор строки
- `barcode (string)`: представление штрих-кода
- `quantity (double)`: количество по штрих-коду

Пример JSON ответа:

```
{
  "meta": {
    "type": "Классификаторы",
    "mediaType": "application/json"
  },
  "media": {
    "00000000-0000-0000-0010-000000000000 ": {
      "erase": false,
      "barcodes": [
        {
          "uuid": "6d4f070a-f555-393f-60ac-4d65a88ac363",
          "barcode": "2000019217961",
          "product": "bd72d930-55bc-11d9-848a-00112f43529a",
          "characteristic": "00000000-0000-0000-0000-000000000000"
        },
        {
          "uuid": "ce7197bc-0f46-9978-17bc-0e9bdc13434a",
          "barcode": "2000019357964",
          "product": "cbcf4994-55bc-11d9-848a-00112f43529a",
          "characteristic": "00000000-0000-0000-0000-000000000000"
        }
      ],
      "products": [
        {
          "uuid": "bd72d930-55bc-11d9-848a-00112f43529a",
          "code": "000000000004",
          "article": "KA-879",
          "description": "Кофеварка JACOBS (Австрия)",
          "price": 28400,
          "rest": 16
        },
        {
          "uuid": "cbcf4994-55bc-11d9-848a-00112f43529a",
          "code": "000000000005",
          "article": "M-545",

```

```

    "description": "Миксер SOLAC мод.545",
    "price": 21300,
    "rest": 38
  },
  ],
  "characteristics": [
    {
      "uuid": "00000000-0000-0000-0000-000000000000",
      "product": "bd72d930-55bc-11d9-848a-00112f43529a",
      "description": ""
    },
    {
      "uuid": "00000000-0000-0000-0000-000000000000",
      "product": "cbcf4994-55bc-11d9-848a-00112f43529a",
      "description": ""
    }
  ]
}
}
}
}

```

Интерфейс передачи результата выполнения задачи для бизнес-процесса

PUT tasks/{task_type}/{task_id}

URL параметры:

- **task_type**: идентификатор типа задачи или документа, полученный терминалом от товароучетной системы - произвольный текстовый параметр, который задается и на стороне терминала, и на стороне товароучетной системы;
- **task_id**: идентификатор задачи

Пример:

```

curl -X PUT
"http://195.54.41.203/rtl/hs/alpha/tasks/Inventorization/020-046-1785-000"
-H "Authorization: Basic <Credentials>"

```

Содержит параметр **Body** - тело запроса, внутри которого находится массив **JSON** с результатами выполнения задачи. Элементы массива имеют следующую структуру:

- **barcode (string)**: представление штрих-кода
- **quantity (double)**: количество по штрих-коду

В случае успеха возвращает **JSON** файл с полем:

success: успешное выполнение

Пример JSON файла:

```
{
  "meta": {
    "type": "result",
    "mediaType": "application/json"
  },
  "media": {
    "success": true
  }
}
```

В случае неудачи, возвращается сообщение об ошибке.

Пример:

```
{
  "meta": {
    "type": "error",
    "mediaType": "text/plain"
  },
  "media": "Некорректная структура данных."
}
```