

Статика 3 – Описание API

Версия 1.3 от 01.07.2020

Оглавление

Описание взаимодействия	2
Требования к внешней системе	2
Протокол обмена.....	2
Список команд:	2
Авторизация (cmd_id: 1).....	3
Синхронизация	4
Товары (cmd_id: 2)	5
Контрагенты (cmd_id: 3)	5
Автомобили (cmd_id: 4)	6
Прицепы (cmd_id: 5)	6
Вагоны (cmd_id: 6)	7
Водители (cmd_id: 7)	7
Отправка результатов взвешивания (cmd_id: 8).....	8
Ответ в виде ошибки (cmd_id: 9).....	11

Описание взаимодействия

ПО «Статика 3» обладает возможностью синхронизации справочников и отправки результатов взвешивания по http-запросу во внешнюю систему посредством встроенного REST API. Любая команда, исходящая от ПО «Статика 3» во внешнюю систему выполняет POST-запрос.

Информация о настройке REST API представлена в руководстве администратора.

Требования к внешней системе

Внешняя система должна обеспечивать реализацию встроенного в ПО «Статика 3» протокола и предоставлять доступ на использование веб-запросов со стороны ПО «Статика 3» с использованием или без использования SSL. При необходимости, внешняя система может потребовать использования авторизации и наличия токена, который она должна выдавать после успешной аутентификации.

Протокол обмена

Протокол обмена базируется на JSON-протоколе. Главной парой служит **cmd_id**, где значение – число, номер команды. Эта пара передается в запросе, а так же ожидается в ответе.

Список команд:

1. Авторизация;
2. Синхронизация товаров;
3. Синхронизация контрагентов;
4. Синхронизация ТС;
5. Синхронизация прицепов;
6. Синхронизация вагонов;
7. Синхронизация водителей;
8. Отправка результатов взвешивания;
9. Ошибка (возможный ответ на любую из вышеопределенных команд).

Авторизация (cmd_id: 1)

Авторизация на сервере производится отправкой команды авторизации каждый раз, когда требуется отправка иной команды при отсутствии токена или при необходимости авторизации.

Если авторизация проходит без токена, она будет применяться каждый раз перед отправкой основной команды. При наличии токена, он будет отправляться в посылке с ключом **token** в каждом пакете любой команды, кроме **1** и **9**.

Пример отправки пакета для авторизации:

```
{  
  "cmd_id": 1,  
  "login": "test",  
  "password": "12345"  
}
```

Где:

login: имя пользователя, строка;

password: пароль, строка.

В ответ интегрируемая система должна продублировать номер команды и прислать токен, если используется.

Пример ответа с токеном:

```
{ "cmd_id": 1, "token": "52390723r5tgn89ng92" }
```

Синхронизация

Синхронизация происходит посредством выполнения запроса от ПО «Статика 3» к интегрируемой системе. В ответ на команду синхронизации, система должна прислать свой справочник согласно протоколу (зеркально), либо ошибку. При этом справочник может быть пустым, либо не иметь в ответе каких-то ключей.

Пример запроса синхронизации:

```
{  
  "cmd_id": 2,  
  "goods":  
  [  
    {  
      "name": "Лук",  
      "code": "12345",  
      "coeff": 0  
    }  
  ]  
}
```

Товары (cmd_id: 2)

goods: массив, где каждый элемент это объект ключ-значение.

Элемент массива **goods**:

name: имя груза, строка;

code: баркод, строка;

coeff: коэффициент, число с плавающей точкой.

Контрагенты (cmd_id: 3)

companies: массив, где каждый элемент это объект ключ-значение.

Элемент массива **companies**:

name: имя контрагента, строка;

law_addr: юридический адрес, строка;

fact_addr: фактический адрес, строка;

contacts: контакты, строка;

inn: ИНН, строка;

kpp: КПП, строка;

ogrn: ОГРН, строка.

Автомобили (cmd_id: 4)

cars: массив, где каждый элемент это объект ключ-значение.

Элемент массива **cars**:

number: номер ТС, строка;

mark: марка, строка;

model: модель, строка;

tare: тара, число с плавающей точкой;

rfid: RFID-метка, строка;

downs: скатность, массив чисел от первой до последней оси;

susps: типы подвесок, массив чисел от первой до последней оси, где 0 – механическая, 1 – пневматическая подвеска;

betweens: межосевое расстояние, массив чисел.

Прицепы (cmd_id: 5)

trailers: массив, где каждый элемент это объект ключ-значение.

Элемент массива **trailers**:

number: номер прицепа, строка;

mark: марка, строка;

model: модель, строка;

downs: скатность, массив чисел от первой до последней оси;

susps: типы подвесок, массив чисел от первой до последней оси, где 0 – механическая, 1 – пневматическая подвеска;

betweens: межосевое расстояние, массив чисел.

Вагоны (cmd_id: 6)

trains: массив, где каждый элемент это объект ключ-значение.

Элемент массива **trains**:

number: номер вагона, строка;

tare: вес тары, число с плавающей точкой;

max_weight: грузоподъемность, число с плавающей точкой.

Водители (cmd_id: 7)

drivers: массив, где каждый элемент это объект ключ-значение.

Элемент массива **drivers**:

fio: ФИО, строка;

reg_number: документ, строка (по нему определяется водитель в справочнике при синхронизации).

Отправка результатов взвешивания (cmd_id: 8)

При отправке результатов взвешивания отправляется вся информация стандартного журнала ПО «Статика 3» от конкретного взвешивания. Это значит, что даже если используются автомобильные весы, в пакете все равно будет находиться ключ с номером вагона. Поэтому на стороне интегрируемого ПО обязательно должна присутствовать фильтрация. Ответом на успешно принятую команду служит пакет с номером команды: { "cmd_id": 8 }

Пример запроса передачи показаний:

```
{ "cmd_id": 8,
  "records":
  [
    {
      "j_id": 2344,
      "tare_date": "12.05.2019 12:32:22",
      "brutto_date": "12.05.2019 13:02:44",
      "tare": 1243.0,
      "brutto": 3454.0,
      "netto": 2211.0,
      "good": "Лук",
      "sender": "ООО Павлов",
      "recipient": "ООО Романов",
      "payer": "ООО Романов",
      "car_number": "A232BB99",
      "trailer_number": "EH8789",
      "train_number": "",
      "driver": "Сидоров Павел Петрович",
      "rfid_code": "",
      "photos_tare": [ "data:image/jpeg;base64,<...>" ],
      "photos_brutto": [ "data:image/jpeg;base64,<...>" ],
      "axles_brutto": [],
      "axles_tare": [],
      "susps": [],
      "downs": [],
      "bets": [],
      "additional": { "<col_name>": "<value>" }
    }
  ] }
```

Где:

records: массив, где каждый элемент это объект ключ-значение.

Элемент массива **records:**

j_id: идентификатор из журнала ПО «Статика 3», число;

tare_date: дата взвешивания тары, строка;

brutto_date: дата взвешивания брутто, строка;

tare: вес тары, число с плавающей точкой;

brutto: вес брутто, число с плавающей точкой;

good: груз, строка;

sender: отправитель, строка;

recipient: получатель, строка;

payer: плательщик, строка;

car_number: номер ТС, строка;

trailer_number: номер прицепа, строка;

train_number: номер вагона, строка;

driver: ФИО водителя, строка;

rfid_code: данные с RFID-метки или QR-кода, строка;

photos_tare: фото взвешивания тары, массив, где каждый элемент строка с массивом байт формата base64;

photos_brutto: фото взвешивания брутто, массив, где каждый элемент строка с массивом байт формата base64;

axles_brutto: вес брутто по осям, массив, где каждый элемент число с плавающей точкой;

axles_tare: вес тары по осям, массив, где каждый элемент число с плавающей точкой;

susps: типы подвесок по осям, массив, где 0 – механическая, 1 – пневматическая;

downs: скатность по осям, массив, каждый элемент число;

bets: межосевое расстояние, массив, каждый элемент число;

additional: дополнительные поля, заведенные в ПО Статика 3, JSON-объект, где ключом выступает название колонки дополнительного поля, а значением строка, относящаяся к выбранной в соответствии с записью журнала.

Ответ в виде ошибки (cmd_id: 9)

При получении любой из команд от ПО «Статика 3», интегрируемая система вправе ответить ошибкой, отослав текст ошибки обратно в ключе **error_text**. Пример:

```
{  
    "cmd_id": 9,  
    "error_text": "Ошибка авторизации: неверный логин или пароль"  
}
```