

# ПТК АЗС API

## Получение данных в формате JSON

версия: 13.27

© ООО «АВТОМАТИКА плюс»

### Содержание

<b>Установка</b>	<b>3</b>
<b>Настройка</b>	<b>3</b>
<b>Получение данных</b>	<b>9</b>
Демонстрационная версия . . . . .	9
Авторизация . . . . .	9
Использование . . . . .	9
<b>Запросы получения данных</b>	<b>10</b>
Список АЗС . . . . .	10
Список видов оплаты ГСМ . . . . .	11
Список видов топлива . . . . .	13
Список смен . . . . .	14
Список отделов магазина . . . . .	15
Список товаров магазина . . . . .	16
Список видов оплаты магазина . . . . .	18
Список контрагентов магазина . . . . .	19
Список складов магазина . . . . .	21
Список контрагентов ГСМ . . . . .	22
Список резервуаров . . . . .	23
Список контрагентов расчётного центра . . . . .	24
Состав услуг . . . . .	25
Товарно-транспортные накладные . . . . .	26
Список заявок . . . . .	29
Отливы . . . . .	30
Приходы/откачки . . . . .	32
Операции магазина . . . . .	35
Партии товаров . . . . .	39
<b>Запросы изменения данных</b>	<b>43</b>
Создание товарно-транспортной накладной . . . . .	43
Создание заявки . . . . .	45

<b>Дополнительно</b>	<b>46</b>
Ошибки . . . . .	47
Ограничения . . . . .	47
<b>История изменений</b>	<b>48</b>

## Установка

Для работы с запросами пользователей используется программа «APIService.exe». Она устанавливается в составе ПТК АЗС в папку «C:\DEXE». Данный файл содержит службу для работы с запросами и программу настройки и управления службой. Для полноценной работы со службой необходимо наличие на ключе защиты опции «Поддержка выгрузки в 1С». Если опции на ключе защиты нет, служба будет работать в демонстрационном режиме с ограничениями по количеству отдаваемых данных.

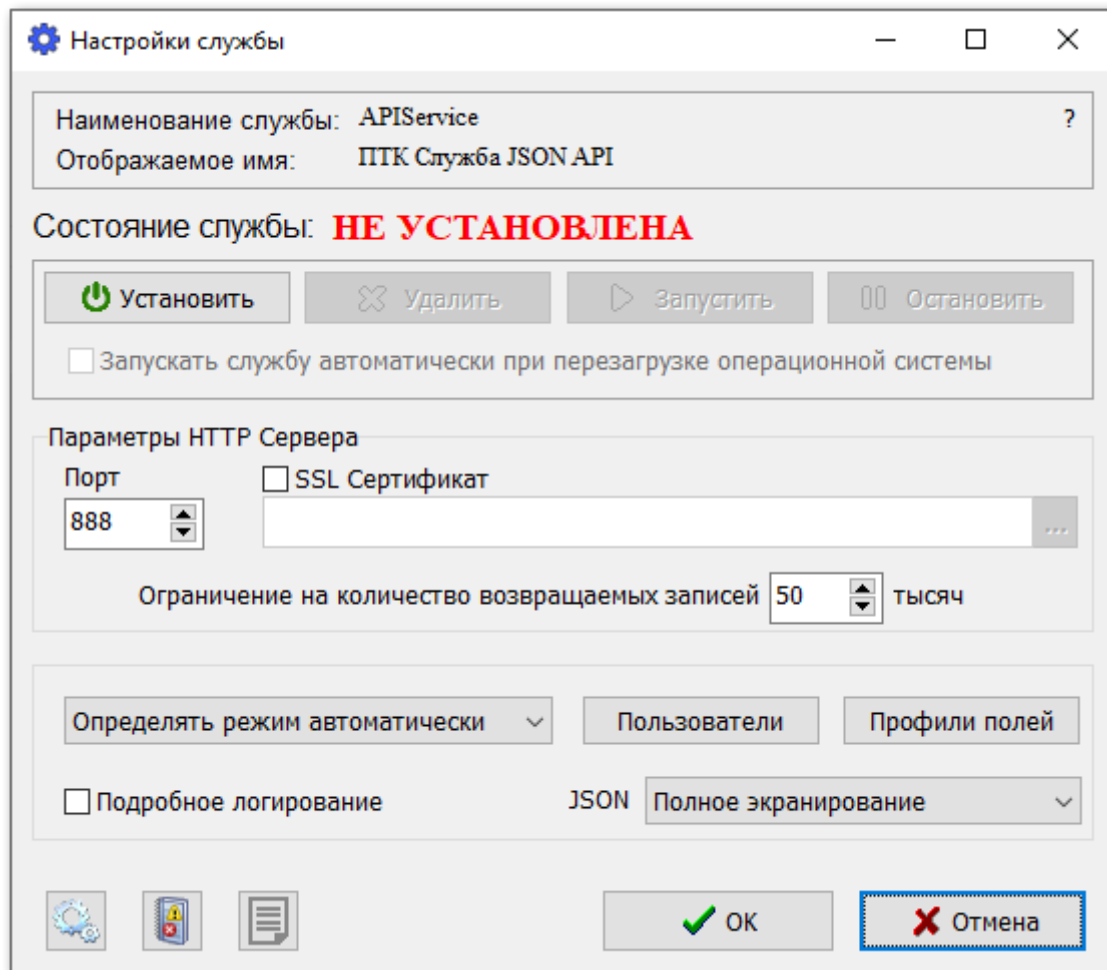


Рис. 1: Главное окно программы настройки.

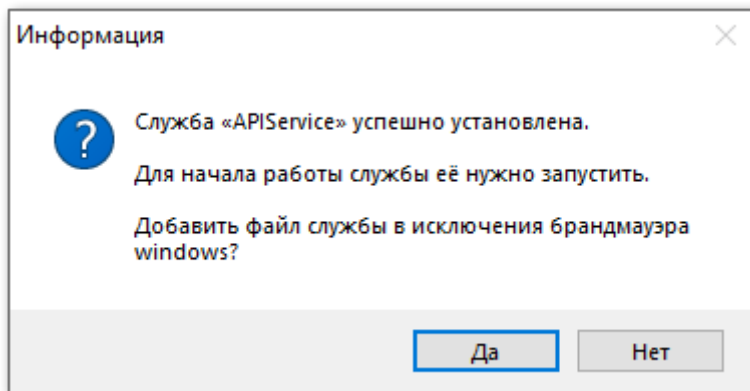


Рис. 2: Служба установлена.

## Настройка

Для настройки необходимо запустить файл: «C:\DEXE\APIService.exe» от имени администратора. В результате будет открыто главное окно программы (рис. 1)

Для установки службы нажмите кнопку «Установить» (рис. 3)[5]. В случае успешной установки появится информационное окно (рис. 2), в котором будет предложено добавить необходимые исключения для встроенного брандмауэра windows. Вы можете доверить это программе установки или настроить брандмауэр самостоятельно позже.

После установки службы будут доступны дополнительные средства управления службой (рис. 3).

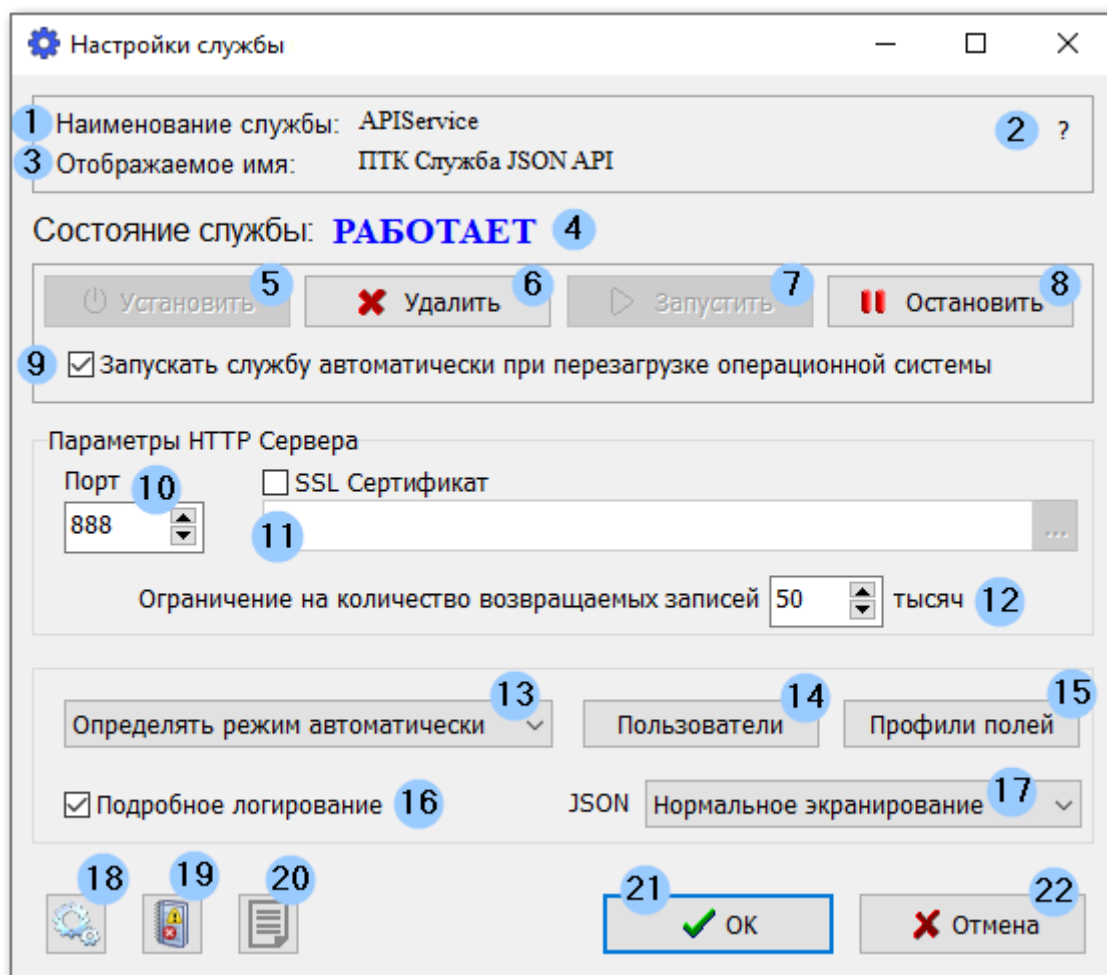


Рис. 3: Элементы управления.

Рассмотрим подробнее элементы окна программы настройки (рис. 3):

1. **Наименование службы:** Для управления службой средствами операционной системы необходимо знать её наименование.
2. **[?]:** Если нажать на данную кнопку можно увидеть список зависимостей службы. Другими словами это список служб от которых зависит запуск данной службы. Для успешного запуска необходимо иметь установленными и запущенными все службы, от которых зависит данная служба. Если службы установлены, но не запущены программа попытается запустить их самостоятельно перед запуском службы. Если эти службы не установлены, необходимо установить их, иначе служба работать не сможет.
3. **Отображаемое имя:** Данное наименование отображается в стандартной острнастке windows.

4. **Состояние службы:** Здесь отображается статус или состояние службы. Служба может находиться в статусе: «Не установлена», «Остановлена», «Останавливается», «Запускается», «Работает»
  - Не установлена - служба не установлена в системе. Служба не зарегистрирована в операционной системе. Операционная система ничего не знает о данной службе. Службу нельзя запустить или остановить. Работать со службой в данном режиме невозможно.
  - Остановлена - служба установлена, но не запущена. Служба зарегистрирована в операционной системе. Службу можно запускать и останавливать с помощью данной программы, с помощью средств операционной системы или другими средствами.
  - Останавливается - служба находится в процессе остановки. В случае успешной остановки служба перейдет в статус «Остановлена»
  - Запускается - служба находится в процессе запуска. В случае успешного запуска служба перейдет в статус «Работает»
  - Работает - служба запущена. Это нормальный режим работы. В данном режиме можно работать с программой.
5. **Установить:** Установить службу в системе. Это действие необходимо выполнить один раз.
6. **Удалить:** Удалить службу из системы. Если не собираетесь больше использовать данную службу, её можно удалить из системы.
7. **Запустить:** Для работы со службой её необходимо запустить. Это нужно сделать после настройки всех необходимых параметров.
8. **Остановить:** Остановить службу. Служба останется установленной в системе, но не будет запущена. Если стоит флаг [Запускать службу автоматически при перезагрузке операционной системы](#), то после перезагрузки компьютера служба запустится автоматически.
9. **Запускать службу автоматически при перезагрузке операционной системы:** Если включена данная настройка служба будет автоматически запущена после перезагрузки компьютера и вам не нужно будет беспокоиться о её запуске.
10. **Порт:** Служба представляет собой HTTP сервер. Для его запуска необходимо выбрать порт ([рис. 3](#)). По умолчанию стоит значение «888», но вы можете использовать любой свободный порт на свое усмотрение. После изменения данного параметра необходимо перезапустить службу.
11. **SSL:** Так же можно установить TLS режим работы HTTP сервера. Для этого необходимо иметь соответствующий сертификат и указать его в настройках программы. Для получения сертификата обратитесь к своему системному администратору. После включения данной настройки необходимо использовать для запросов протокол HTTPS вместо HTTP.
12. **Ограничение на количество возвращаемых записей:** Для оптимального использования памяти получение данных ограничено данным значением. [Подробнее.](#)
13. **Режим работы:** Программа может работать в трёх режимах ([рис. 3](#)):
  - Режиме сервера, если она запущена на сервере ПТК АЗС.
  - Режиме офиса, если она запущена на компьютере с ПТК АЗС Офис.
  - Режиме магазина, если она запущена на компьютере только с ПТК АЗС Магазин.

Программа может автоматически включить **режим работы** исходя из имеющихся опций ключа защиты. Режимы включаются в следующем приоритетном порядке: Режим офиса, режим сервера, режим магазина. Или можно включить необходимый режим работы вручную. Для этого на ключе защиты должна быть соответствующая опция.

14. **Пользователи:** При нажатии данной кнопки открывается окно управления пользователями (рис. 4). Для работы программы необходимо создать хотя бы одного пользователя. Для настройки разграничения данных можно настроить необходимое количество пользователей и каждому пользователю назначить свой **профиль полей**.

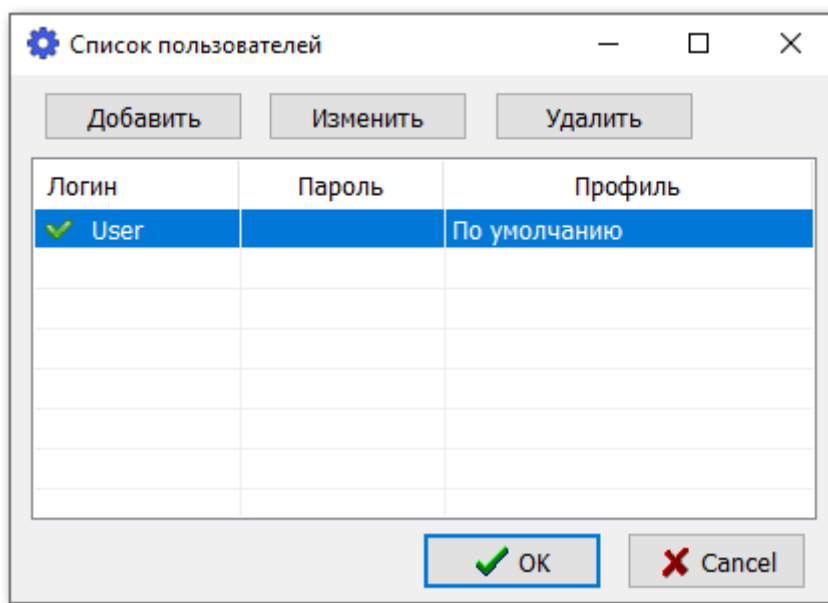


Рис. 4: Пользователи.

15. **Профили полей:** При нажатии данной кнопки открывается окно управления и настройки профилей полей (рис. 5). По умолчанию используется профиль «По умолчанию» с минимальным набором полей. Его нельзя удалить, но можно настроить. Имеется возможность создать и настроить любое количество дополнительных профилей. Для каждого **пользователя** можно выбрать свой профиль набора полей.
16. **Подробное логирование:** При установленной настройке в журнал работы программы пишется дополнительная информация для удобства обслуживания.
17. **JSON:** В результате работы программы формируется ответ в формате JSON. Формат подразумевает несколько вариантов экранирования. Подробнее об экранировании будет изложено [ниже](#).
18. **Службы:** Кнопка открывает системное окно управления службами.
19. **Журнал событий:** Кнопка открывает системный журнал событий.
20. **Протокол работы:** Кнопка открывает текущий протокол работы программы.
21. **ОК:** Закрытие программы с сохранением настроек. Если параметры были изменены и служба запущена программа предложит перезапустить службу с сохранением и учётом изменённых параметров (рис. 6).
22. **Отмена:** Закрытие программы без сохранения настроек.

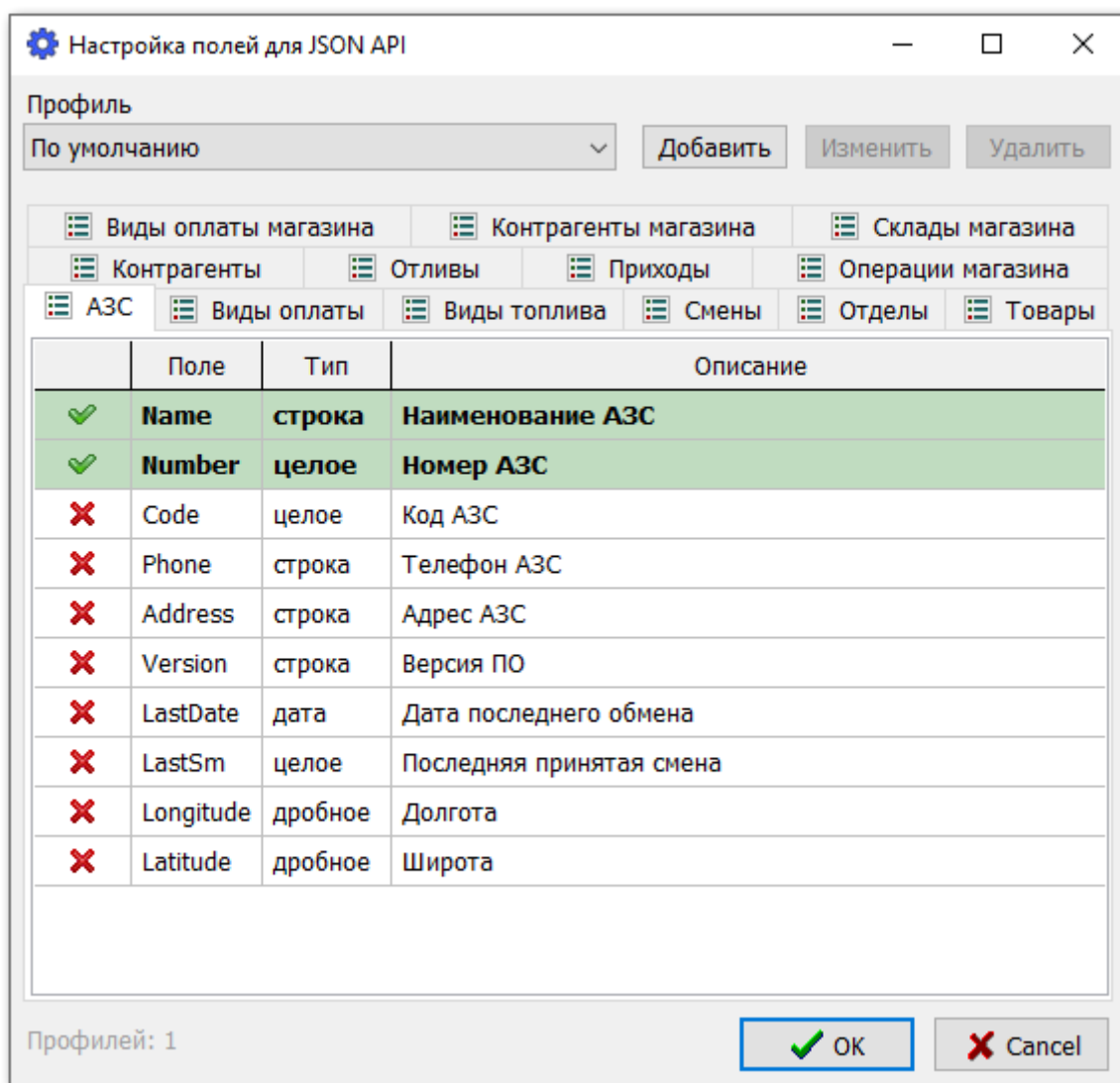


Рис. 5: Профили настройки полей.

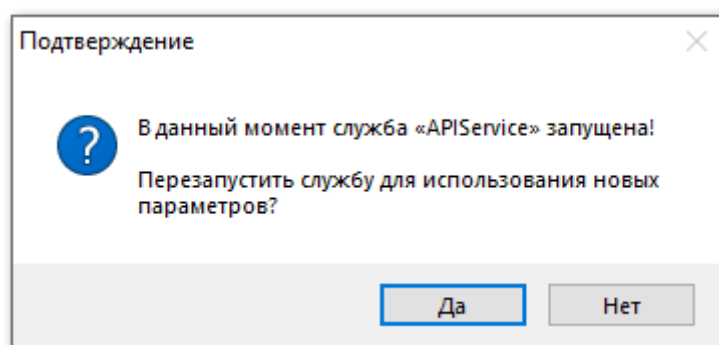


Рис. 6: Подтверждение применения параметров.

После настройки программы необходимо запустить службу с помощью кнопки «Запустить» (рис. 3)[pic7]. После успешного запуска службы в строке состояние службы появится статус «Работает». Если служба по каким либо причинам не смогла запуститься, ошибку можно посмотреть в протоколе работы программы (рис. 3)[20] или в системном журнале событий windows (рис. 3)[19].

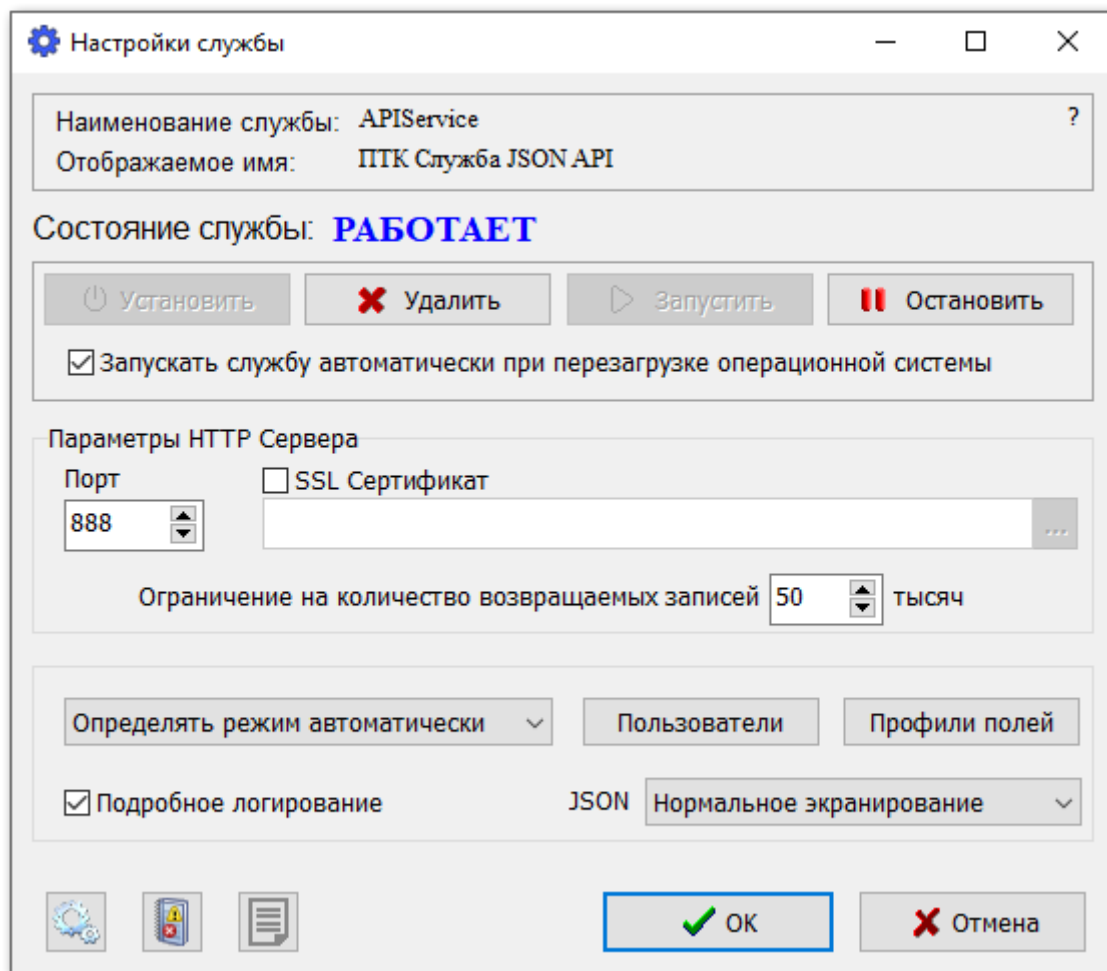


Рис. 7: Служба запущена.

## Получение данных

Для получения данных в формате JSON необходимо отправить HTTP запрос к соответствующей службе ПТК АЗС. Запрос должен быть передан с помощью методов GET или POST. Параметры строки запроса, в соответствии с форматом HTTP, разделяются символом «&». Некоторые символы должны экранироваться в формате % + код символа. Например, символ «+» меняется на «%2B» а код пробела «%20»

При использовании POST запроса можно передать параметры в теле запроса в виде JSON. Для этого «Content-Type» должен быть «application/json». **Внимание!** Нужно учитывать что формат JSON чувствителен к регистру символов.

Для удобства работы в условиях нескольких часовых поясов все даты программа выдает в формате международного стандарта ISO 8601. Подробнее об этом формате можно узнать, например, [здесь](#) или [здесь](#).

**Внимание!** Большие объемы передаваемых данных приводят к значительной нагрузке на сеть. Текстовые данные имеют очень хороший коэффициент сжатия. Поэтому для разгрузки сети имеет смысл использовать сжатие передаваемых по сети данных. Для сжатия отправляемых сервером данных в заголовке «Accept-Encoding» запроса необходимо передать тип подходящего алгоритма сжатия. Сервер поддерживает «gzip» и «deflate» алгоритмы сжатия.

## Демонстрационная версия

---

Демонстрационная версия (Swagger UI):

<http://www.automatikaplus.ru/azs/swagger/index.html>

## Авторизация

[авторизация](#)

Для работы с web сервером необходима авторизация. Для авторизации нужно использовать либо базовую авторизацию в заголовке запроса или в каждом запросе дополнительно передавать параметры **Login** и **Password** необходимого **пользователя**.

[авторизация](#)

## Использование

---

Возвращаемые данные условно делятся на две категории. Есть справочники со списком элементов и со всеми их характеристиками. И есть данные, которые имеют связь с соответствующими справочниками. Например, есть справочник товаров и есть операции магазина по товарам и они связаны между собой с помощью идентификатора товара. В запросе операций возвращаются только параметры операций, а характеристики товара, отдела, АЗС и т.д. находятся в соответствующих справочниках. Если нужен номер смены или оператор, то эта информация есть в справочнике смен.

Каждая смена характеризуется временем открытия и временем закрытия, а каждая продажа имеет время продажи. По этим данным можно определить в какую смену попадает продажа.

Так же необходимо учитывать что JSON это текстовый формат и не может содержать в составе текстовых полей некоторые символы. Например, символ перевода строки. При формировании строк такие символы экранируются или, другими словами, заменяются на так называемые escape-последовательности. В настройках можно выбрать либо полное экранирование и тогда все управляющие символы и символы выходящие за пределы латинского алфавита меняются на шестнадцатеричный код, либо нормальное экранирование с заменой девяти критических для формата символов (0x0, 0x8, 0x9, 0xA, 0xC, 0xD, 0x22, 0x2F, 0x5C), либо минимальное экранирование и тогда экранируются только три самые необходимые escape-последовательности (0x0, 0x22, 0x5C).

После получения ответа на запрос рекомендуется в первую очередь проверить параметр «**Success**». Если он равен True, ответ на запрос выполнен корректно, а если параметр «**Success**» равен False, запрос выполнен с ошибкой и нет смысла обрабатывать остальные данные, кроме сообщения об ошибке.

**Внимание!** Если ваша программа имеет версию ниже той, которая указана в данной инструкции некоторые функции работать не будут или могут работать иначе. Обновите версию программы до актуальной. Версия программы указана в заголовке данного документа.

## Запросы получения данных

Для получения данных запрос должен быть вида:

**протокол://адрес\_сервера:порт/GetJson? + параметры.**

Обязательные параметры запроса:

- **Login** – имя пользователя для авторизации
- **Password** – пароль пользователя для авторизации
- **JType** – тип запроса.

Список возможных значений этого параметра приведен ниже:

---

### Список АЗС

- **JType = 1** Список АЗС (Только в режиме сервера или офиса)

Необязательные параметры

- **NUM** – список номеров АЗС для фильтрации. *(Только в режиме офиса)*

Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?JType=1&NUM=2,4:7,9,10
```

- **CODE** – список кодов АЗС для фильтрации. *(Только в режиме офиса)*

Список из одного или нескольких кодов АЗС или диапазонов кодов АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные АЗС с учетом остальных фильтров.

Пример:

```
...&CODE=2,4:7,9,10
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
Name,	строка	Наименование АЗС
Number,	целое	Номер АЗС
Code,	целое	Код АЗС
Address,	строка	Адрес АЗС
Phone,	строка	Телефон АЗС
Version,	строка	Версия ПО
LastDate,	строка	Дата последнего обмена ISO 8601
LastSm,	целое	Последняя принятая смена
Longitude,	дробное	Долгота
Latitude	дробное	Широта
},...]		
}		

## Список видов оплаты ГСМ

- **JType = 2** Список видов оплаты ГСМ *(Только в режиме сервера или офиса)*

Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. *(Только в режиме офиса)*

Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **NUM** – список номеров видов оплаты для фильтрации

Список из одного или нескольких номеров видов оплаты или диапазонов номеров видов оплаты через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные виды оплаты с учетом остальных фильтров.

Пример:

```
...&NUM=1,4:7,101:104
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив видов оплаты
Num,	целое	Номер вида оплаты
Name,	строка	Наименование вида оплаты
BaseNum,	целое	Базовый номер вида оплаты
CheckType,	целое	Тип чека вида оплаты
Enabled,	логика	Статус вида оплаты
PrintKKM,	логика	Печать произвольных документов на кассе перед пробитием чека
PrintKKMpred,	логика	Печать произвольных документов на кассе перед отпуском «до полного»
DscOptions,	логика	Кто рассчитывает скидку (0 - есть скидка; 1 - запрашивать на сервере; 2 - нет скидки; 3 - есть скидка с учетом дополнительного параметра)
KKM,	целое	Номер ККМ
NDS	дробное	Ставка НДС
NalogSystem	целое	Тип налоговой системы
SelfNDS	дробное	Собственный НДС
AccessLevel	целое	Уровень доступа
},...]		
},...]		

}

## Список видов топлива

- **JType = 3** Список видов топлива (*Только в режиме сервера или офиса*)

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (*Только в режиме офиса*)  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **NAME** – наименование вида топлива для фильтрации  
При заполнении этого параметра будут возвращены виды топлива совпадающие с заданным значением. Если параметр будет пустой или параметра не будет совсем, будут выданы данные по всем доступным видам топлива с учетом остальных фильтров.

Пример:

```
...&NAME=АИ-95
```

### Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС ( <i>только в режиме сервера или офиса</i> )
AZSNum,	целое	Номер АЗС ( <i>только в режиме офиса</i> )
Data[{	массив	Массив видов топлива
FullName,	целое	Полное наименование вида топлива
Name,	строка	Наименование вида топлива
Price,	целое	Цена вида топлива
Type,	целое	Тип вида топлива
Measure,	логика	Единица измерения
AltTable,	логика	Используется ли таблица соответствий
AltName,	строка	Наименование из таблицы соответствий
},...]		

```
}, ...]  
}
```

## Список смен

- **JType = 4** Список смен (Только в режиме сервера или офиса)

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)

Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **NUM** – список номеров смен

Список из одного или нескольких номеров смен или диапазонов номеров смен через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные смены с учетом остальных фильтров.

Пример:

```
...&NUM=1,2,4,90:100
```

- **BDateTime** – начало интервала (в формате ISO 8601)

Не забываем экранировать символ «+». Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Если заданы параметры диапазона дат запрос вернет все смены, хотя бы частично входящие в заданный диапазон. Если любой параметр диапазона будет пустой или его не будет совсем, будут выданы все возможные смены с учетом остальных фильтров.

Пример:

```
...&BDateTime=2023-08-11T14:52+03:00
```

- **EDateTime** – окончание интервала (в формате ISO 8601)

Не забываем экранировать символ «+». Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Если заданы параметры диапазона дат запрос вернет все смены, хотя бы частично входящие в заданный диапазон. Если любой параметр диапазона будет пустой или его не будет совсем, будут выданы все возможные смены с учетом остальных фильтров.

Пример:

```
...&EDateTime=2023-08-11T23:59:59+03:00
```

- **FilterType** – тип фильтра по датам

Есть три способа ограничить временной интервал запрашиваемых данных.

0 — Полный. В этом случае в параметрах `BDateTime` и `EDateTime` передается начало и окончание интервала. В выборку попадают все смены хотя бы частично перекрываемые данным интервалом.

1 — Ограниченный. В этом случае в параметрах `BDateTime` и `EDateTime` передается начало и окончание интервала. В выборку попадают только те смены, которые полностью попадают в данный интервал.

2 — Суточно сменный способ фильтрации. В этом случае передается начало интервала и окончание интервала в параметрах `BDateTime` и `EDateTime`, но используется только дата без времени. Интервал начинается с началом первой смены в сутках, заданных началом интервала и заканчивается последней сменой, начатой (но не обязательно завершённой) в сутках, заданных окончанием интервала.

Если значение параметра не задано или параметр отсутствует считается что `FilterType=0`

Параметр учитывается только если задано начало и окончание интервала с помощью параметров `BDateTime` и `EDateTime`

Пример:

```
...&FilterType=1
```

Формат возвращаемых данных:

Поле	Тип	Описание
<code>RepSource,</code>	строка	Источник данных
<code>DateTime,</code>	строка	Дата и время выполнения запроса ISO 8601
<code>Encoding,</code>	строка	Кодировка ответа
<code>Success,</code>	логика	Признак выполнения запроса
<code>Request{&gt;,</code>	строка	Список параметров запроса
<code>RepName,</code>	строка	Наименование запроса
<code>Data[{</code>	массив	Массив данных
<code>AZSName,</code>	строка	Наименование АЗС (только в режиме сервера или офиса)
<code>AZSNum,</code>	целое	Номер АЗС (только в режиме офиса)
<code>Data[{</code>	массив	Массив смен
<code>FullName,</code>	целое	Полное наименование вида топлива
<code>Name,</code>	строка	Наименование вида топлива
<code>Price,</code>	целое	Цена вида топлива
<code>Type,</code>	целое	Тип вида топлива
<code>Measure,</code>	логика	Единица измерения
<code>AltTable,</code>	логика	Используется ли таблица соответствий
<code>AltName,</code>	строка	Наименование из таблицы соответствий
<code>},...]</code>		
<code>},...]</code>		
<code>}</code>		

- **JType = 5** Список отделов магазина

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. *(Только в режиме офиса)*  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **DEP** – список кодов отделов. *(Только в режиме офиса и если они заполнены)*  
Список из одного или нескольких кодов отделов или диапазонов кодов отделов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем отделам с учетом остальных фильтров.  
**ВНИМАНИЕ!** Фильтр работает только для отделов с заполненными кодами. Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив отделов
DepName,	строка	Наименование отдела
DepCode,	целое	Код отдела (только в режиме офиса и если он задан)
},...]		
},...]		
}		

## Список товаров магазина

- **JType = 6** Список товаров магазина

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. *(Только в режиме офиса)*  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **DEP** – список кодов отделов. *(Только в режиме офиса и если они заполнены)*  
Список из одного или нескольких кодов отделов или диапазонов кодов отделов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем отделам с учетом остальных фильтров.  
**ВНИМАНИЕ!** Фильтр работает только для отделов с заполненными кодами. Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

- **ID** – список идентификаторов товаров.  
Список из одного или нескольких идентификаторов товаров или диапазонов идентификаторов товаров через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные товары с учетом остальных фильтров.

Пример:

```
...&ID=3,5,8,9:100, 200:300
```

### Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data{	массив	Массив отделов
DepName,	строка	Наименование отдела



Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

Формат возвращаемых данных:

Поле	Тип	Описание
<code>RepSource,</code>	строка	Источник данных
<code>DateTime,</code>	строка	Дата и время выполнения запроса <i>ISO 8601</i>
<code>Encoding,</code>	строка	Кодировка ответа
<code>Success,</code>	логика	Признак выполнения запроса
<code>Request{}</code> ,	строка	Список параметров запроса
<code>RepName,</code>	строка	Наименование запроса
<code>Data{</code>	массив	Массив данных
<code>AZSName,</code>	строка	Наименование АЗС (только в режиме сервера или офиса)
<code>AZSNum,</code>	целое	Номер АЗС (только в режиме офиса)
<code>Data{</code>	массив	Массив отделов
<code>DepName,</code>	строка	Наименование отдела
<code>DepCode,</code>	целое	Код отдела (только в режиме офиса и если он задан)
<code>Data{</code>	массив	Массив видов оплаты
<code>Name,</code>	строка	Наименование вида оплаты
<code>Num,</code>	целое	Номер вида оплаты
<code>Type,</code>	целое	Тип вида оплаты
<code>Check,</code>	целое	Тип чека
<code>GSMVO,</code>	строка	Код товара
<code>Deleted,</code>	логика	Категория
<code>},...</code>		
<code>},...</code>		
<code>},...</code>		
<code>}</code>		

## Список контрагентов магазина

- **JType = 8** Список контрагентов магазина

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **DEP** – список кодов отделов. (Только в режиме офиса и если они заполнены)

Список из одного или нескольких кодов отделов или диапазонов кодов отделов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем отделам с учетом остальных фильтров.

**ВНИМАНИЕ!** Фильтр работает только для отделов с заполненными кодами. Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

- **ID** – список идентификаторов контрагентов.

Список из одного или нескольких идентификаторов контрагентов или диапазонов идентификаторов контрагентов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные контрагенты с учетом остальных фильтров.

Пример:

```
...&ID=3,5,8,9:100
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив отделов
DepName,	строка	Наименование отдела
DepCode,	целое	Код отдела (только в режиме офиса и если он задан)
Data[{	массив	Массив контрагентов
Name,	строка	Наименование
FullName,	строка	Полное наименование
INN,	строка	Идентификационный номер налогоплательщика
Id,	целое	Идентификатор контрагента
IsSupplier,	логика	Признак поставщика
IsManufacturer,	логика	Признак производителя
Deleted,	логика	Признак удаленного контрагента
Address,	строка	Адрес
Phones,	строка	Телефоны
Info,	строка	Контактная информация
},...]		

```

    }, ...]
  }, ...]
}

```

## Список складов магазина

- **JType = 9** Список складов магазина

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. *(Только в режиме офиса)*  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **DEP** – список кодов отделов. *(Только в режиме офиса и если они заполнены)*  
Список из одного или нескольких кодов отделов или диапазонов кодов отделов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем отделам с учетом остальных фильтров.  
**ВНИМАНИЕ!** Фильтр работает только для отделов с заполненными кодами. Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

- **ID** – список идентификаторов складов.  
Список из одного или нескольких идентификаторов складов или диапазонов идентификаторов складов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные склады с учетом остальных фильтров.

Пример:

```
...&ID=3,5,7,90:100
```

### Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса

<code>Request{}</code> ,	строка	Список параметров запроса
<code>RepName</code> ,	строка	Наименование запроса
<code>Data[{</code>	массив	Массив данных
<code>AZSName</code> ,	строка	Наименование АЗС (только в режиме сервера или офиса)
<code>AZSNum</code> ,	целое	Номер АЗС (только в режиме офиса)
<code>Data[{</code>	массив	Массив отделов
<code>DepName</code> ,	строка	Наименование отдела
<code>DepCode</code> ,	целое	Код отдела (только в режиме офиса и если он задан)
<code>Data[{</code>	массив	Массив складов
<code>Name</code> ,	строка	Наименование
<code>Id</code> ,	целое	Идентификатор склада
<code>Internal</code> ,	логика	Признак внутреннего склада
<code>Deleted</code> ,	логика	Признак удаленного склада
<code>Kod</code> ,	строка	Код склада
<code>INN</code> ,	строка	Идентификационный номер налогоплательщика
<code>KPP</code>	строка	Код причины постановки на учет
<code>},...]</code>		
<code>},...]</code>		
<code>},...]</code>		
<code>}</code>		

## Список контрагентов ГСМ

- **JType = 10** Список контрагентов ГСМ (Только в режиме сервера или офиса)

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)

Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **ID** – список идентификаторов контрагентов.

Список из одного или нескольких идентификаторов контрагентов или диапазонов идентификаторов контрагентов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные контрагенты с учетом остальных фильтров.

Пример:

```
...&ID=3,5:8,90:100
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{}	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив контрагентов
Name,	строка	Наименование
FullName,	строка	Полное наименование
INN,	строка	Идентификационный номер налогоплательщика
Id,	строка	Идентификатор контрагента
Deleted,	логика	Признак удаленного контрагента
Address,	строка	Адрес
Phone,	строка	Телефон
Email,	строка	Адрес электронной почты
},...]		
},...]		
}		

## Список резервуаров

- **JType = 11** Список резервуаров (Только в режиме сервера или офиса)

Запрос списка резервуаров и их текущее состояние. В режиме сервера состояние резервуаров запрашивается непосредственно от ПКТ АЗС сервера поэтому он должен быть запущен. В режиме офиса информация о состоянии резервуаров зависит от обмена офиса с сервером.

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных

<code>DateTime,</code>	строка	Дата и время выполнения запроса <i>ISO 8601</i>
<code>Encoding,</code>	строка	Кодировка ответа
<code>Success,</code>	логика	Признак выполнения запроса
<code>Request{}</code> ,	строка	Список параметров запроса
<code>RepName,</code>	строка	Наименование запроса
<code>Data[{</code>	массив	Массив данных
<code>AZSName,</code>	строка	Наименование АЗС (только в режиме сервера или офиса)
<code>AZSNum,</code>	целое	Номер АЗС (только в режиме офиса)
<code>Data[{</code>	массив	Массив резервуаров
<code>Name,</code>	строка	Наименование резервуара
<code>GSMName,</code>	строка	Наименование вида топлива
<code>Time,</code>	строка	Время получения данных <i>ISO 8601</i>
<code>IsBuffer,</code>	логика	Признак промежуточного буфера
<code>MaxVolume,</code>	дробное	Объём резервуара, л
<code>Volume,</code>	дробное	Текущий объём топлива, л
<code>VolumeCalc,</code>	дробное	Расчётный объём топлива, л
<code>Level,</code>	дробное	Текущий уровень топлива, мм
<code>LevelGauge,</code>	логика	Наличие уровнемера
<code>WaterLevel,</code>	дробное	Уровень подтоварной воды, мм
<code>Signal1,</code>	дробное	Порог сигнализации 1, л
<code>Signal2,</code>	дробное	Порог сигнализации 2, л
<code>Plotn,</code>	дробное	Плотность, т/м <sup>3</sup>
<code>Pressure,</code>	дробное	Давление, кгс/см <sup>2</sup>
<code>Massa,</code>	дробное	Масса, кг (только в режиме офиса)
<code>Temper,</code>	дробное	Температура, °C
<code>SmVolumeP,</code>	дробное	Объём прихода за смену, л
<code>SmMassaP,</code>	дробное	Масса прихода за смену, кг
<code>SmVolumeR,</code>	дробное	Объём расхода за смену, л
<code>SmMassaR,</code>	дробное	Масса расхода за смену, кг
<code>},...]</code>		
<code>},...]</code>		
<code>}</code>		

## Список контрагентов расчётного центра

- **JType = 12** Список контрагентов расчётного центра (*Только в режиме офиса*)

Аналог файла `BezNaICardINN.DBF` для поиска контрагента по номеру карты. Данные берутся из расчётного центра и не привязаны к какой либо АЗС.

### Необязательные параметры

- **Card** – полный номер карты.

*Полный номер карты состоящий из четырёх цифр эмитента, трёх цифр эмиссии и восьми цифр номера карты.*

Пример:

```
...&Card=000100100000001
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{ },	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
Card,	строка	Полный номер карты
Name,	строка	Наименование контрагента
INN,	строка	ИНН контрагента
Holder,	строка	Держатель карты
}, ...]		
}		

## Состав услуг

- **JType = 13** Состав услуг

Состав услуг на текущий момент времени.

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. *(Только в режиме офиса)*  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **DEP** – список кодов отделов. *(Только в режиме офиса и если они заполнены)*  
Список из одного или нескольких кодов отделов или диапазонов кодов отделов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем отделам с учетом остальных фильтров.  
**ВНИМАНИЕ!** Фильтр работает только для отделов с заполненными кодами. Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

- **ID** – список идентификаторов услуг.

Список из одного или нескольких идентификаторов услуг или диапазонов идентификаторов услуг через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные услуги с учетом остальных фильтров.

Пример:

```
...&ID=-2147483332, -2147483334,
-2147483336:-2147483339
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса <i>ISO 8601</i>
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив отделов
DepName,	строка	Наименование отдела
DepCode,	целое	Код отдела (только в режиме офиса и если он задан)
Data[{	массив	Массив услуг
Id,	целое	Идентификатор услуги (ссылка на справочник товаров [6])
BegTime,	строка	Дата последнего изменения <i>ISO 8601</i>
EndTime,	строка	Дата будущего изменения, если оно запланировано <i>ISO 8601</i>
Ingredients[{	массив	Массив ингредиентов услуги
Id,	строка	Идентификатор товара (ссылка на справочник товаров [6])
Count,	дробное	Количество в услуге
ZPrice,	дробное	Закупочная цена
RPrice,	дробное	Цена продажи
},...]		
},...]		
},...]		
},...]		
}		

## Товарно-транспортные накладные

- **JType = 14** Товарно-транспортные накладные

Справочник товарно-транспортных накладных. Товарно-транспортную накладную можно создать с помощью запроса ([1])

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. *(Только в режиме офиса)*

*Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.*

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **ID** – список идентификаторов ТТН.

*Список из одного или нескольких идентификаторов ТТН или диапазонов идентификаторов ТТН через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные ТТН с учетом остальных фильтров.*

Пример:

```
...&ID=212, 474, 74:83
```

### Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив документов
Id,	целое	Идентификатор ТТН
TTNSER,	строка	ТТН Серия
TTNNUMBER,	строка	Номер
TTNDATE,	дробное	Дата и время ISO 8601
TTNAPPLIED,	логика	Проведено
GO_ID,	целое	Идентификатор грузоотправителя (ссылка на справочник контрагентов [10])
GP_ID,	целое	Идентификатор грузополучателя (ссылка на справочник контрагентов [10])
PER_ID,	целое	Идентификатор перевозчика (ссылка на справочник контрагентов [10])
PLAT_ID,	целое	Идентификатор плательщика (ссылка на справочник контрагентов [10])
NADB,	дробное	Наценка %

SKLADRASHOD,	дробное	Расходы (склад., трансп.)
ALLSUMMANUM,	дробное	Всего к оплате
ALLSUMMATEXT,	строка	Всего к оплате, прописью
PRILPAGESNUM,	целое	Приложение (листов)
PRILPAGESTEXT,	строка	Приложение (листов) прописью
MNETTONUM,	дробное	Масса нетто, т
MNETTOTEXT,	строка	Масса нетто, т (прописью)
MBRUTTONUM,	дробное	Масса брутто, т
MBRUTTOTEXT,	строка	Масса брутто, т (прописью)
ALLMEST,	строка	Всего мест (прописью)
BOSSPOST,	строка	Отпуск разрешил, должность
BOSSFIO,	строка	Отпуск разрешил, ФИО
BUHFIO,	строка	ФИО главного (старшего) бухгалтера
DOVERNUM,	строка	Доверенность перевозчика, номер
DOVERDATE,	строка	Доверенность перевозчика, дата ISO 8601
GOPOST,	строка	Отпуск произвел, должность
GOFIO,	строка	Отпуск произвел, ФИО
DOSTDATE,	строка	Срок доставки груза ISO 8601
CARMARKA,	строка	Марка автомобиля
CARNUM,	строка	Государственный номер автомобиля
DRIVERPOST,	строка	Водитель, должность
DRIVERFIO,	строка	Водитель, ФИО
DRIVERUDOST,	строка	Водитель, номер удостоверения
PLIST,	строка	Путевой лист
LICCARD,	целое	Лицензионная карточка (0 - Не зачеркивать; 1 - Стандартная; 2 - Ограниченная)
TRANSPKIND,	строка	Вид перевозки
PKOD,	строка	Код перевозки
PUNKTPOGRUZ,	строка	Пункт погрузки
PUNKTRAZGRUZ,	строка	Пункт разгрузки
READDRESS,	строка	Переадресовка
PMARSHRUT,	строка	Маршрут
PRMARKA1,	строка	Прицеп, марка
PRNUM1,	строка	Прицеп, гос. знак
PRGARNUM1,	строка	Прицеп, гаражный номер
SFNUMBER,	строка	Номер счета-фактуры
SFDATE,	строка	Дата счета-фактуры ISO 8601
SFPLATDOC,	строка	Номер и дата платежного документа
SFGOTYPE,	целое	Поле Грузоотправитель и его адрес (0 - Вывести текст полностью; 1 - Вывести текст "он же"; 2 - Вывести прочерк)
SFBOSSFIO,	строка	Руководитель организации
SFINDPREDFIO,	строка	Индивидуальный предприниматель
SFBUHFIO,	строка	Главный бухгалтер
SFGOSREG,	строка	Реквизиты свид-ва о гос. регистрации
TNNUMBER,	строка	Номер товарной накладной TOPГ12
TNDATE,	строка	Дата товарной накладной TOPГ12 ISO 8601
TNPLAT_ID,	целое	Идентификатор плательщика TOPГ12 (ссылка на справочник контрагентов [10])
TNPOST_ID,	целое	Идентификатор поставщика TOPГ12 (ссылка на справочник контрагентов [10])
TNOSNNAME,	строка	Основание TOPГ12

TNOSNUMBER,	строка	Основание, номер ТОРГ12
TNOSNDATE,	строка	Основание, дата ТОРГ12 ISO 8601
TNTRNUMBER,	строка	Транспортная накладная, номер
TNTRDATE,	строка	Транспортная накладная, дата ISO 8601
TNOPERATION,	строка	Транспортная накладная, вид операции
KVITNUMBER,	строка	Квитанция об отгрузке, номер
KVITDATE,	строка	Квитанция об отгрузке, дата ISO 8601
GO_FORWARDER,	логика	Является ли грузоотправителем экспедитором
CAROWNERTYPE,	целое	Тип владения транспортным средством (0 - собственность; 1 - аренда; 2 - лизинг)
Deleted,	логика	Признак удаления
},...]		
},...]		
}		

## Список заявок

- **JType = 15** Список заявок (*Только в режиме офиса*)

Заявки на создание документов. Запрос возвращает список заявок созданных с помощью запроса создания заявки ([2])

### Необязательные параметры

- **ID** – список идентификаторов заявок.

*Список из одного или нескольких идентификаторов заявок или диапазонов идентификаторов заявок через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные заявки с учетом остальных фильтров.*

Пример:

```
...&ID=0, 1, 3:5
```

### Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{}	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
Id,	строка	Идентификатор заявки
Date,	строка	Дата создания ISO 8601
Type,	строка	Тип документа (0 - приход; 1 - возврат; 2 - отмена)
ExtId,	строка	Идентификатор внешней системы
AzsNum,	дробное	Номер АЗС

<code>Status,</code>	дробное	Статус заявки (0 - создана; 1 - в очереди; 2 - отправлена; 3 - применена; 4 - отменена; 5 - ошибка)
<code>Error,</code>	строка	Текстовое описание ошибки
<code>Modify,</code>	строка	Дата последнего изменения ISO 8601
<code>},...]</code>		
<code>}</code>		

---

## Отливы

---

- **JType = 101** Отливы (Только в режиме сервера или офиса)

### Обязательные параметры

- **FilterType** – тип фильтрации

Есть три способа ограничить временной интервал запрашиваемых данных.

0 — Фильтр по датам. В этом случае в обязательных параметрах `BDateTime` и `EDateTime` передается начало и окончание периода.

1 — Фильтр по сменам. В этом случае параметры `BDateTime` и `EDateTime` не используются. А в обязательном для этого способа параметре `SmNum` передается список номеров или диапазонов номеров смен через запятую.

2 — Суточно сменный способ фильтрации. В этом случае передается начало интервала и окончание интервала в обязательных параметрах `BDateTime` и `EDateTime`, но используется только дата без времени. Интервал начинается с началом первой смены в сутках, заданных началом интервала и заканчивается последней сменой, начатой (но не обязательно завершенной) в сутках, заданных окончанием интервала.

Если значение параметра не задано или параметр отсутствует считается что `FilterType=0`

Пример:

```
.../GetJson?JType=101&FilterType=1
```

- **BDateTime** – начало запрашиваемого интервала (в формате ISO 8601)

Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если `FilterType` равен 0 или 2. При `FilterType=2` используется только дата без времени. Параметр обязателен при `FilterType` равном 0 или 2. Не забываем экранировать символ «+».

Пример:

```
...&BDateTime=2022-06-01T00:00+03:00
```

- **EDateTime** – окончание запрашиваемого интервала (в формате ISO 8601)

Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если `FilterType` равен 0 или 2. При `FilterType=2` используется только дата без времени. Параметр обязателен при `FilterType` равном 0 или 2. Не забываем экранировать символ «+».

Пример:

```
...&EDateTime=2022-07-25T23:59:59+03:00
```

- **SmNum** – список номеров смен

Список из одного или нескольких номеров смен или диапазонов номеров смен через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные с учетом остальных фильтров. Параметр используется только если FilterType=1. Параметр обязателен при FilterType=1

Пример:

```
...&SmNum=1:10,15,17:20
```

## Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)

Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

## Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив отливов
Date,	строка	Время начала отлива ISO 8601
Kolvo,	дробное	Количество топлива
Summa,	дробное	Сумма
VO,	целое	Вид оплаты
SubVO,	целое	Подвид оплаты
GSMName,	строка	Наименование вида топлива (ссылка на справочник видов топлива [3])
SmNum,	целое	Номер смены
BasePrice,	дробное	Базовая цена
GSMPrice,	дробное	Цена по виду топлива
BaseSumma,	дробное	Базовая Сумма

TRKNum,	целое	Номер ТРК
TankNum,	целое	Номер резервуара
Pokupatel,	строка	Покупатель
Plotn,	дробное	Плотность
Massa,	дробное	Масса
Contragent,	строка	Идентификатор контрагента для ведомости (ссылка на справочник контрагентов [10])
User	строка	Авторизованный пользователь. Вывод данного столбца увеличивает время запроса. Используйте только при необходимости.

```

},...]
},...]
}

```

## Приходы/откачки

- **JType = 102** Приходы/откачки (Только в режиме сервера или офиса)

### Обязательные параметры

- **FilterType** – тип фильтрации

Есть три способа ограничить временной интервал запрашиваемых данных.

0 — Фильтр по датам. В этом случае в обязательных параметрах `BDateTime` и `EDateTime` передается начало и окончание периода.

1 — Фильтр по сменам. В этом случае параметры `BDateTime` и `EDateTime` не используются. А в обязательном для этого способа параметре `SmNum` передается список номеров или диапазонов номеров смен через запятую.

2 — Суточно сменный способ фильтрации. В этом случае передается начало интервала и окончание интервала в обязательных параметрах `BDateTime` и `EDateTime`, но используется только дата без времени. Интервал начинается с началом первой смены в сутках, заданных началом интервала и заканчивается последней сменой, начатой (но не обязательно завершенной) в сутках, заданных окончанием интервала.

Если значение параметра не задано или параметр отсутствует считается что `FilterType=0`

Пример:

```
.../GetJson?JType=101&FilterType=1
```

- **BDateTime** – начало запрашиваемого интервала (в формате *ISO 8601*)

Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если `FilterType` равен 0 или 2. При `FilterType=2` используется только дата без времени. Параметр обязателен при `FilterType` равном 0 или 2. Не забываем экранировать символ «+».

Пример:

```
...&BDateTime=2022-06-01T00:00+03:00
```

- **EDateTime** – окончание запрашиваемого интервала (в формате *ISO 8601*)  
Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если FilterType равен 0 или 2. При FilterType=2 используется только дата без времени. Параметр обязателен при FilterType равном 0 или 2. Не забываем экранировать символ «+».

Пример:

```
...&EDateTime=2022-07-25T23:59:59+03:00
```

- **SmNum** – список номеров смен  
Список из одного или нескольких номеров смен или диапазонов номеров смен через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные с учетом остальных фильтров. Параметр используется только если FilterType=1. Параметр обязателен при FilterType=1

Пример:

```
...&SmNum=1:10,15,17:20
```

### Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

### Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса <i>ISO 8601</i>
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив приходов/откачек
BegTime,	строка	Время начала операции <i>ISO 8601</i>
EndTime,	строка	Время окончания операции <i>ISO 8601</i>
GSMName,	строка	Наименование вида топлива (ссылка на справочник видов топлива [3])
SmNum,	целое	Номер смены

OpNum,	целое	Порядковый номер операции
TankNum,	целое	Номер резервуара
Contragent,	целое	Идентификатор контрагента (ссылка на справочник контрагентов [10])
Operator,	строка	Оператор
Seals	строка	Пломба
Measurer,	строка	Измеритель
MeasureTime,	строка	Время измерения ISO 8601
Description,	строка	Примечание
OperationInfo: {		
Direction,	логика	True-приход, False-откачка
IsAuto,	логика	Признак автоматической операции
IsMove,	логика	Признак операции «Перекачка»
}		
TTNInfo: {		
Num,	строка	Номер ТТН
Date,	строка	Дата ТТН ISO 8601
Volume,	дробное	Объем по ТТН
Plotn,	дробное	Плотность по ТТН
Massa,	дробное	Масса по ТТН
Temper	дробное	Температура по ТТН
Seals	строка	Пломба по ТТН
}		
CarInfo: {		
Name,	строка	Бензовоз
Section,	целое	Номер секции бензовоза
Volume,	дробное	Объем в бензовозе
Plotn,	дробное	Плотность в бензовозе
Massa,	дробное	Масса в бензовозе
Temper	дробное	Температура в бензовозе
}		
TankStateBefore: {		
Level,	целое	Уровень на начало операции
Volume,	дробное	Объем на начало операции
Plotn,	дробное	Плотность на начало операции
Massa,	дробное	Масса на начало операции
Temper	дробное	Температура на начало операции
}		
TankStateAfter: {		
Level,	целое	Уровень на конец операции
Volume,	дробное	Объем на конец операции
Plotn,	дробное	Плотность на конец операции
Massa,	дробное	Масса на конец операции
Temper	дробное	Температура на конец операции
}		
PrxResult: {		
Volume,	дробное	Объем фактически принятого
Massa,	дробное	Масса фактически принятого
Plotn,	дробное	Плотность фактически принятого
}		
},...]		
},...]		

}

## Операции магазина

---

- **JType = 103** Операции магазина

### Обязательные параметры

- **FilterType** – тип фильтрации

*Есть три способа ограничить временной интервал запрашиваемых данных.*

*0 — Фильтр по датам. В этом случае в обязательных параметрах `BDateTime` и `EDateTime` передается начало и окончание периода.*

*1 — Фильтр по сменам. В этом случае параметры `BDateTime` и `EDateTime` не используются. А в обязательном для этого способа параметре `SmNum` передается список номеров или диапазонов номеров смен через запятую.*

*2 — Суточно сменный способ фильтрации. В этом случае передается начало интервала и окончание интервала в обязательных параметрах `BDateTime` и `EDateTime`, но используется только дата без времени. Интервал начинается с началом первой смены в сутках, заданных началом интервала и заканчивается последней сменой, начатой (но не обязательно завершённой) в сутках, заданных окончанием интервала.*

*3 — Фильтр по дате последнего изменения документа. В этом случае в обязательных параметрах `BDateTime` и `EDateTime` передается начало и окончание периода последнего изменения документа. Дополнительно в этом режиме можно использовать параметры `ModifyDiff` и/или `DocID`.*

*Если значение параметра не задано или параметр отсутствует считается что `FilterType=0`*

*Если задан параметр `DocID`, то данный параметр не учитывается*

Пример:

```
.../GetJson?JType=101&FilterType=1
```

- **BDateTime** – начало запрашиваемого интервала (в формате *ISO 8601*)

*Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если `FilterType` равен 0 или 2. При `FilterType=2` используется только дата без времени. Параметр обязателен при `FilterType` равном 0 или 2. Не забываем экранировать символ «+».*

Пример:

```
...&BDateTime=2022-06-01T00:00+03:00
```

- **EDateTime** – окончание запрашиваемого интервала (в формате *ISO 8601*)

*Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если `FilterType` равен 0 или 2. При `FilterType=2` используется только дата без времени. Параметр обязателен при `FilterType` равном 0 или 2. Не забываем экранировать символ «+».*

Пример:

```
...&EDateTime=2022-07-25T23:59:59+03:00
```

- **SmNum** – список номеров смен

Список из одного или нескольких номеров смен или диапазонов номеров смен через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные с учетом остальных фильтров. Параметр используется только если FilterType=1. Параметр обязателен при FilterType=1

Пример:

```
...&SmNum=1:10,15,17:20
```

- **DocId** – идентификатор документа

Список из одного или нескольких идентификаторов документов или диапазонов идентификаторов документов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные с учетом остальных фильтров.

— Данный параметр имеет приоритетное значение и отменяет значения параметров диапазонов дат и смен

Пример:

```
...&DocId=15237:16589,17591,19112
```

## Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)

Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **DEP** – список кодов отделов. (Только в режиме офиса и если они заполнены)

Список из одного или нескольких кодов отделов или диапазонов кодов отделов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем отделам с учетом остальных фильтров.

**ВНИМАНИЕ!** Фильтр работает только для отделов с заполненными кодами. Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

- **Type** – тип операции.

Список из одного или нескольких типов операций или диапазонов типов операций через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем операциям с учетом остальных фильтров.

*Типы операций перечислены в конце раздела.*

Пример:

```
...&Type=0,2,4,8
```

- **Doc** – признак наличия документа.

Признак наличия документа в иерархии операций. Если этот параметр присутствует в запросе, операции выводятся в составе соответствующих документов. Если параметр в запросе отсутствует или равен нулю, выводятся только операции сплошным списком без упоминания документов. Достаточно наличие самого параметра без значения.

Пример:

```
...&Doc...
```

- **ModifyDiff** – разница между датой документа и датой последнего изменения.

*Данный параметр используется только в режиме **FilterType=3***

В этом параметре задается для дополнительной фильтрации разница между датой создания документа и датой последнего изменения документа в минутах. То есть в ответ попадут только документы с разницей между датами равной или превышающей значение, заданное этим параметром. Если значение будет пустым или параметра не будет совсем, будут выданы все возможные записи с учётом остальных фильтров.

Пример:

```
...&ModifyDiff=600
```

- **FoldSrv** – сворачивать услуги.

0 (Или если параметр отсутствует) - сворачивание услуг не выполняется. Запрос работает максимально быстро. Сама услуга и её состав располагается в документе отдельно на одном уровне.

1 - Услуги в запросе сворачиваются в одну строку, то есть выводятся без ингредиентов. Ингредиенты в ответе не выводятся совсем. Время обработки запроса и расход оперативной памяти дополнительно увеличивается пропорционально количеству услуг в запрашиваемых данных.

2- Услуги сворачиваются. Ингредиенты выводятся в составе услуги. Время обработки запроса и расход оперативной памяти максимально увеличивается пропорционально количеству услуг в запрашиваемых данных.

Пример:

```
...&FoldSrv=1
```

Формат возвращаемых данных:

Поле	Тип	Описание
------	-----	----------

{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{>,	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив отделов
DepName,	строка	Наименование отдела
DepCode,	целое	Код отдела (только в режиме офиса и если он задан)
Data[{	массив	Массив документов
DocId,	целое	Идентификатор документа
BaseId,	целое	Идентификатор документа основания. Например, при возврате это документ продажи. Заполняется при наличии значения.
DocNum,	строка	Номер документа
DocDate,	строка	Дата документа ISO 8601
DocType,	целое	Тип документа (значения перечислены ниже)
Modify,	строка	Дата последнего изменения документа ISO 8601
VO,	целое	Вид оплаты
SubVO,	целое	Подвид оплаты
VOGSM,	целое	Связанный вид оплаты ГСМ
Card,	строка	Карта
KKM,	целое	Номер ККМ
Note,	строка	Примечание. Заполняется при наличии значения.
Reason,	строка	Причина обмена с поставщиком (Для операций обмена с поставщиком). Заполняется при наличии значения.
ClientAddr,	строка	Для операции продажи: телефон или электронный адрес клиента для отправки чека. Для операций внутреннего перемещения: идентификатор перемещения. Заполняется при наличии значения.
SuppId,	целое	Идентификатор поставщика (ссылка на справочник контрагентов [8]). Заполняется при наличии значения.
SendId,	целое	Идентификатор отправителя (ссылка на справочник контрагентов или складов для внутреннего перемещения [8], [9]). Заполняется при наличии значения.
RecipId,	целое	Идентификатор получателя (ссылка на справочник контрагентов или складов для внутреннего перемещения [8], [9]). Заполняется при наличии значения.



`EDateTime` передается начало и окончание периода.

1 — Фильтр по сменам. В этом случае параметры `BDateTime` и `EDateTime` не используются. А в обязательном для этого способа параметре `SmNum` передается список номеров или диапазонов номеров смен через запятую.

2 — Суточно сменный способ фильтрации. В этом случае передается начало интервала и окончание интервала в обязательных параметрах `BDateTime` и `EDateTime`, но используется только дата без времени. Интервал начинается с началом первой смены в сутках, заданных началом интервала и заканчивается последней сменой, начатой (но не обязательно завершенной) в сутках, заданных окончанием интервала.

Если значение параметра не задано или параметр отсутствует считается что `FilterType=0`

Если задан параметр `ID`, то данный параметр не учитывается

Пример:

```
.../GetJson?JType=101&FilterType=1
```

- **`BDateTime`** – начало запрашиваемого интервала (в формате *ISO 8601*)

Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если `FilterType` равен 0 или 2. При `FilterType=2` используется только дата без времени. Параметр обязателен при `FilterType` равном 0 или 2. Не забываем экранировать символ «+».

Пример:

```
...&BDateTime=2022-06-01T00:00+03:00
```

- **`EDateTime`** – окончание запрашиваемого интервала (в формате *ISO 8601*)

Если часовой пояс не задан, будет использоваться локальный часовой пояс сервера. Параметр используется только если `FilterType` равен 0 или 2. При `FilterType=2` используется только дата без времени. Параметр обязателен при `FilterType` равном 0 или 2. Не забываем экранировать символ «+».

Пример:

```
...&EDateTime=2022-07-25T23:59:59+03:00
```

- **`SmNum`** – список номеров смен

Список из одного или нескольких номеров смен или диапазонов номеров смен через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные с учетом остальных фильтров. Параметр используется только если `FilterType=1`. Параметр обязателен при `FilterType=1`

Пример:

```
...&SmNum=1:10,15,17:20
```

- **`Id`** – идентификатор партии

Список из одного или нескольких идентификаторов партий или диапазонов идентификаторов партий через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем,

будут выданы все возможные данные с учетом остальных фильтров.  
— Данный параметр имеет приоритетное значение и отменяет значения параметров диапазонов дат и смен

Пример:

```
...&Id=15237:16589,17591,19112
```

## Необязательные параметры

- **AZS** – список номеров АЗС для фильтрации. (Только в режиме офиса)  
Список из одного или нескольких номеров АЗС или диапазонов номеров АЗС через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем АЗС с учетом остальных фильтров.

Пример:

```
.../GetJson?...&AZS=1,4:7,11, 20:25
```

- **DEP** – список кодов отделов. (Только в режиме офиса и если они заполнены)  
Список из одного или нескольких кодов отделов или диапазонов кодов отделов через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем отделам с учетом остальных фильтров.  
**ВНИМАНИЕ!** Фильтр работает только для отделов с заполненными кодами. Коды заполняются в программе ПТК-Офис Меню: Экспорт → Параметры экспорта для товаров → Коды товарных отделов.

Пример:

```
...&DEP=1:3,5,7:9
```

- **Type** – тип товаров партии.  
Список из одного или нескольких типов товаров или диапазонов типов товаров через запятую. Диапазон задается как два значения разделенные символом «:». Если список будет пустой или параметра не будет совсем, будут выданы все возможные данные по всем товарам с учетом остальных фильтров.

Пример:

```
...&Type=0,2,4,8
```

- **Doc** – признак наличия документа.  
Признак наличия документа в иерархии операций. Если этот параметр присутствует в запросе, операции выводятся в составе соответствующих документов. Достаточно самого параметра без значения. Если параметр в запросе отсутствует или равен нулю, выводятся только операции.

Пример:

```
...&Doc...
```

## Формат возвращаемых данных:

Поле	Тип	Описание
{		
RepSource,	строка	Источник данных
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак выполнения запроса
Request{ },	строка	Список параметров запроса
RepName,	строка	Наименование запроса
Data[{	массив	Массив данных
AZSName,	строка	Наименование АЗС (только в режиме сервера или офиса)
AZSNum,	целое	Номер АЗС (только в режиме офиса)
Data[{	массив	Массив отделов
DepName,	строка	Наименование отдела
DepCode,	целое	Код отдела (только в режиме офиса и если он задан)
Data[{	массив	Массив документов
DocId,	целое	Идентификатор документа
DocNum,	строка	Номер документа
DocDate,	строка	Дата документа ISO 8601
DocType,	целое	Тип документа (значения перечислены ниже)
Modify,	строка	Дата последнего изменения документа ISO 8601
SmNum,	целое	Номер смены. Вывод данного столбца увеличивает время запроса. Используйте только при необходимости.
User,	строка	Авторизованный пользователь. Вывод данного столбца увеличивает время запроса. Используйте только при необходимости.
Data[{	массив	Массив операций
Id,	целое	Идентификатор партии
GoodsId,	целое	Идентификатор товара (Ссылка на справочник товаров [6])
Type,	целое	0 - собственные товары; 1 - комиссионные товары
Count,	дробное	Количество товара
ZPrice,	дробное	Закупочная цена. Выводится только если нет закупочной суммы. Если есть закупочная сумма, то закупочная цена вычисляется как закупочная сумма делённая на количество.
ZSum,	дробное	Закупочная сумма. Заполняется при наличии значения.
Use,	строка	Срок годности. Заполняется при наличии значения.
SuppId,	целое	Идентификатор поставщика (ссылка на справочник контрагентов [8]) Заполняется при наличии значения.
NDS	дробное	Закупочный НДС. Значение НДС равное нулю означает «Без НДС»
},...]		
},...]		
},...]		

```
    }, ... ]
  }
```

## Запросы изменения данных

Для изменения данных POST запрос должен быть вида:

**протокол://адрес\_сервера:порт/SetJson?** + параметры.

Данные запроса передаются в теле запроса в формате JSON.

Обязательные параметры запроса:

- **Login** – имя пользователя для авторизации
- **Password** – пароль пользователя для авторизации
- **JType** – тип запроса.

Список возможных значений этого параметра приведен ниже:

## Создание товарно-транспортной накладной

- **JType = 1** Создание товарно-транспортной накладной (*Только в режиме сервера*)

Запрос создаёт в базе заголовочную часть документа товарно-транспортная накладная. Список созданных товарно-транспортных накладных можно получить с помощью запроса [14]

Тело запроса в формате JSON:

Поле	Тип	Описание
{		
"TTNSER":#,	стр20	ТТН Серия
"TTNNUMBER":#,	стр20	Номер
"TTNDATE":#,	строка	Дата и время ISO 8601
"TTNAPPLIED":#,	логика	Проведено
"GO_ID":#,	целое	Идентификатор грузоотправителя (ссылка на справочник контрагентов [10])
"GP_ID":#,	целое	Идентификатор грузополучателя (ссылка на справочник контрагентов [10])
"PER_ID":#,	целое	Идентификатор перевозчика (ссылка на справочник контрагентов [10])
"PLAT_ID":#,	целое	Идентификатор плательщика (ссылка на справочник контрагентов [10])
"NADB":#,	дробное	Наценка %
"SKLADRASHOD":#,	дробное	Расходы (склад., трансп.)

"ALLSUMMANUM":#,	дробное	Всего к оплате
"ALLSUMMATEXT":#,	стр100	Всего к оплате, прописью
"PRILPAGESNUM":#,	целое	Приложение (листов)
"PRILPAGESTEXT":#,	стр100	Приложение (листов) прописью
"MNETTONUM":#,	дробное	Масса нетто, т
"MNETTOTEXT":#,	стр100	Масса нетто, т (прописью)
"MBRUTTONUM":#,	дробное	Масса брутто, т
"MBRUTTOTEXT":#,	стр100	Масса брутто, т (прописью)
"ALLMEST":#,	стр100	Всего мест (прописью)
"BOSSPOST":#,	стр20	Отпуск разрешил, должность
"BOSSFIO":#,	стр100	Отпуск разрешил, ФИО
"BUHFIO":#,	стр100	ФИО главного (старшего) бухгалтера
"DOVERNUM":#,	стр20	Доверенность перевозчика, номер
"DOVERDATE":#,	строка	Доверенность перевозчика, дата ISO 8601
"GOPOST":#,	стр20	Отпуск произвел, должность
"GOFIO":#,	стр100	Отпуск произвел, ФИО
"DOSTDATE":#,	строка	Срок доставки груза ISO 8601
"CARMARKA":#,	стр20	Марка автомобиля
"CARNUM":#,	стр20	Государственный номер автомобиля
"DRIVERPOST":#,	стр20	Водитель, должность
"DRIVERFIO":#,	стр100	Водитель, ФИО
"DRIVERUDOST":#,	стр20	Водитель, номер удостоверения
"PLIST":#,	стр20	Путевой лист
"LICCARD":#,	целое	Лицензионная карточка (0 - Не зачеркивать; 1 - Стандартная; 2 - Ограниченная)
"TRANSPKIND":#,	стр100	Вид перевозки
"PKOD":#,	стр20	Код перевозки
"PUNKTPOGRUZ":#,	стр100	Пункт погрузки
"PUNKTRAZGRUZ":#,	стр100	Пункт разгрузки
"READDRESS":#,	стр100	Переадресовка
"PMARSHRUT":#,	стр20	Маршрут
"PRMARKA1":#,	стр20	Прицеп, марка
"PRNUM1":#,	стр20	Прицеп, гос. знак
"PRGARNUM1":#,	стр20	Прицеп, гаражный номер
"SFNUMBER":#,	стр20	Номер счета-фактуры
"SFDATE":#,	строка	Дата счета-фактуры ISO 8601
"SFPLATDOC":#,	стр100	Номер и дата платежного документа
"SFGOTYPE":#,	целое	Поле Грузоотправитель и его адрес (0 - Вывести текст полностью; 1 - Вывести текст "он же"; 2 - Вывести прочерк)
"SFBOSSFIO":#,	стр100	Руководитель организации
"SFINDPREDFIO":#,	стр100	Индивидуальный предприниматель
"SFBUHFIO":#,	стр100	Главный бухгалтер
"SFGOSREG":#,	стр100	Реквизиты свид-ва о гос. регистрации
"TNNUMBER":#,	стр20	Номер товарной накладной ТОРГ12
"TNDATE":#,	строка	Дата товарной накладной ТОРГ12 ISO 8601
"TNPLAT_ID":#,	целое	Идентификатор плательщика ТОРГ12 (ссылка на справочник контрагентов [10])
"TNPOST_ID":#,	целое	Идентификатор поставщика ТОРГ12 (ссылка на справочник контрагентов [10])
"TNOSNNAME":#,	стр100	Основание ТОРГ12
"TNOSNNUMBER":#,	стр20	Основание, номер ТОРГ12

"TNOSNDATE":#,	строка	Основание, дата TOPГ12 ISO 8601
"TNTRNUMBER":#,	стр20	Транспортная накладная, номер
"TNTRDATE":#,	строка	Транспортная накладная, дата ISO 8601
"TNOOPERATION":#,	стр20	Транспортная накладная, вид операции
"KVTITNUMBER":#,	стр20	Квитанция об отгрузке, номер
"KVTITDATE":#,	строка	Квитанция об отгрузке, дата ISO 8601
"GO_FORWARDER":#,	логика	Является ли грузоотправителем экспедитором
"CAROWNERTYPE":#,	целое	Тип владения транспортным средством (0 - собственность; 1 - аренда; 2 - лизинг)

```
}
```

Формат возвращаемых данных:

Поле	Тип	Описание
{		
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак успешного выполнения запроса
Request{>,	строка	Список параметров запроса
Body{>,	строка	Тело запроса (для контроля)
RepName,	строка	Наименование запроса
TTN_ID	целое	Идентификатор созданного документа (ссылка на справочник ТТН [14])
}		

## Создание заявки

- **JType = 2** Создание заявки (Только в режиме офиса)

Заявки на создание документов прихода и возврата поставщику. Заявка создаётся в офисе, затем отправляется на АЗС и исполняется. Статус созданной заявки можно узнать через справочник заявок [15]

Тело запроса в формате JSON:

Поле	Тип	Описание
{		
"externalId":#,	стр100	идентификатор внешней системы
"annulateId":#,	строка	Идентификатор заказа, который необходимо отменить
"azsId":#,	целое	Номер АЗС (из справочника АЗС [1])
"departmentId":#,	целое	Идентификатор единого справочника
"docType":#,	целое	Тип документа: 0 - приход, 1 - возврат поставщику (в primaryDoc можно указать документ прихода), 2 - отмена неприменённой операции (в annulateId передать externalId отменяемой операции)
"isCommision":#,	логика	Признак комиссионных товаров
"docNumber":#,	строка	Номер документа
"note":#,	строка	Примечание
"summa":#,	дробное	Сумма документа

"summaNds":#,	дробное	Сумма НДС
"supplierId":#,	целое	Идентификатор поставщика (из справочника контрагентов [8])
"supplierOriginalId":#,	целое	Идентификатор оригинального поставщика (из справочника контрагентов [8])
"primaryDoc": {	строка	Первичный документ
"number":#,	строка	Номер документа
"date":#,	строка	Дата документа ISO 8601
},		
"invoiceDoc":{	строка	Счёт-фактура
"number":#,	строка	Номер счёта-фактуры
"date":#,	строка	Дата счёта-фактуры ISO 8601
},		
"goods": [{	массив	Массив списка товаров
"id":#,	целое	Идентификатор товара (из справочника товаров [6])
"quantity":#,	дробное	Количество
"price":#,	дробное	Цена
"nds":#,	целое	Значение НДС
"summa":#,	дробное	Сумма
"summaNds":#	дробное	Сумма НДС
},...]		
}		

#### Формат возвращаемых данных:

Поле	Тип	Описание
{		
DateTime,	строка	Дата и время выполнения запроса ISO 8601
Encoding,	строка	Кодировка ответа
Success,	логика	Признак успешного выполнения запроса
Request{}	строка	Список параметров запроса
Body{}	строка	Тело запроса (для контроля)
RepName,	строка	Наименование запроса
DocId	целое	Идентификатор созданной заявки (ссылка на справочник заявок [15])
}		

## Дополнительно

### Типы документов:

(в скобках связанные типы операций)

- 0 - приход товара (0);
- 1 - возврат поставщику (1);
- 2 - продажа товара (2);
- 3 - возврат от покупателя (3);
- 4 - перемещение, расход (4);
- 5 - инвентаризация (21, 37);
- 8 - перемещение, приход (8);
- 9 - переоценка;
- 10 - заказ товара;
- 21 - списание со склада (21);
- 33 - производство (0, 21);
- 40 - обмен с поставщиком (22, 38);

### Типы операций:

(в скобках связанные типы документов):

- 0 - приход товара (0, 33);
- 1 - возврат товара поставщику (1);
- 2 - продажа товара (2);
- 3 - возврат от покупателя (3);
- 4 - перемещение, расход (4);
- 8 - перемещение, приход (8);
- 21 - инвентаризация, производство, расход (5, 21, 33);
- 22 - обмен с поставщиком, расход (40);
- 37 - инвентаризация, приход (5);
- 38 - обмен с поставщиком, приход (40);

## Ошибки

---

В случае ошибки возвращается JSON вида:

```
{"Success": False, "ErrorMsg": "Текст ошибки"}
```

## Ограничения

---

Количество возвращаемых данных в запросах задается в [настройках](#). По умолчанию ограничено 50 000 записей. Ограничение действует на объекты запроса, например, при запросе смен, ограничивается количество смен, при запросе товаров ограничение действует на количество товаров и так далее.

Если на ключе защиты нет опции «Поддержка выгрузки в 1С» программа работает в демонстрационном режиме и количество записей по всем запросам ограничено небольшими значениями для демонстрации возможностей.

## История изменений

### • Версия: 13.07

- Добавился новый запрос контрагентов из ГСМ базы (JType = [10]).
- В запросах данных ([101], [103]) добавилось поле «User» - авторизованный в данный момент пользователь. Вывод данного столбца увеличивает время запроса. Используйте только при необходимости.
- В запросах данных ([101], [103]) добавилось поле «SmNum» - номер смены. Запрос смен теперь отвязан от запроса отливов. Если не запрашивать номера смен, то запрос выполняется гораздо быстрее. Вывод данного столбца увеличивает время запроса. Быстрее и правильнее загрузить один раз список смен [4] и получать смену оттуда по временному интервалу.
- В запросе отливов добавилось поле «Contragent» идентификатор контрагента для ведомости
- В запросе приходов/откачек вместо структуры «Contragent» появилось поле «Contragent» в котором возвращается идентификатор контрагента (ссылка на справочник контрагентов [10]) Теперь по контрагенту можно получить любые данные (из справочника [10])
- В запросе [102] изменился тип столбца «OperationInfo.Direction». Теперь True - приход; False – откачка (Были текстовые значения «in» и «out»)
- В запросе списка АЗС [1] (офис) добавились поля: Version, LastDate, LastSm, Longitude, Latitude
- Изменился инструмент формирования JSON. Теперь он требует меньше времени и оперативной памяти для формирования JSON ответа.
- Исправлена ошибка фильтрации по отделам. Фильтрация работала неправильно.
- В запросе операций магазина [103] добавлено поле розничного НДС (RNDS)

### • Версия: 13.08

- Изменился исходный формат документации (L<sup>A</sup>T<sub>E</sub>X).
- Добавились разделы «Установка» и «Настройка». Теперь вся документация по API находится в одном (данном) файле.
- Изменился символ разделителя диапазонов целых чисел во входных параметрах. Был «-», стал «:». Нельзя было использовать отрицательные значения. Теперь отрицательные значения можно использовать как отдельно, так и в составе диапазона.

### • Версия: 13.09

- Исправлена ошибка сворачивания услуг.
- Добавлена возможность перепутать начало и окончание диапазонов при задании входных параметров целого типа. Будет браться диапазон от меньшего к большему.
- В запрос смен [4] добавился входной параметр типа фильтрации по датам «FilterType».
- Добавлен вариант базовой авторизации через заголовок запроса.
- Исправлены коды возвратов HTTP при ошибках. До этого при большинстве ошибок код возврата был равен нулю.
- Исправлен суточно-сменный фильтр. Неправильно возвращал при отсутствии смен в заданных сутках.

- Закупочная цена товара в запросе операций магазина ([103]) теперь берется из партии по формуле: если есть закупочная сумма, цена берется как сумма делённая на количество, иначе берется просто цена.
- Версия: 13.10
  - Увеличена мантиса закупочной цены товара в запросе операций магазина ([103]).
  - Исправлена ошибка задвоения (затроения при двухчековом возврате) отливов ([101]).
- Версия: 13.11
  - Переименовано поле вида оплаты товаров. Было «VidOpl» стало «VO» по аналогии с видом оплаты для ГСМ ([103]).
  - Добавлено поле подвида оплаты ГСМ «SubVO» ([101]).
  - Добавлено поле подвида оплаты товаров «SubVO» ([103]).
  - Добавлено поле связанного вида оплаты ГСМ для товаров «VOGSM» ([103]).
  - Переименовано поле идентификатора документа основания. Было «BaseDocId» стало «BaseId» ([103]).
  - Переименован входной параметр типа операции в запросе операций магазина. Был «OperType» стал «Type» ([103]).
  - Переименовано поле типа операции. Было «OperType» стало «Type» ([103]).
  - Добавлено поле идентификатора партии товаров «BatchId» в запрос операций магазина ([103]).
  - Добавлено поле идентификатора услуги «SrvId» в запрос операций магазина ([103]).
  - Добавлен запрос партий товаров ([104]).
- Версия: 13.13
  - Добавлен запрос списка резервуаров ([11]).
- Версия: 13.14
  - Добавлен запрос списка контрагентов из расчётного центра ([12]).
- Версия: 13.17
  - Добавлен запрос списка услуг с составом ([13]).
- Версия: 13.19
  - Исправлена ошибка, связанная с отсутствием часового пояса в датах запроса, если локальный часовой пояс отрицательный.
- Версия: 13.21
  - Добавлено поле идентификатора документа прихода для возврата поставщику «OrigId» в запросе операций магазина ([103]).
  - Исправлена ошибка заполнения групп в json в запросе приходов/откачек ([102]).
  - Добавлено поле «Fractional» разрешения дробного количества товара ([6]).
- Версия: 13.22
  - Добавлено поле «Marking» признак маркированного товара ([6]).

- Версия: 13.24

- Добавлены поля «TTNInfo.Seals, Seals, Measurer, MeasureTime» в запрос по приходам ([102]).
- Исправлена ошибка фильтрации по сменам запроса по приходам ([102]).
- Добавлен запрос списка товарно-транспортных накладных ([14]).
- Добавлена группа запросов изменения данных
- Добавлен запрос создания ТТН ([1]).
- Добавлен запрос создания заявки ([2]).
- Добавлен запрос списка заявок ([15]).

- Версия: 13.25

- Добавлены поля «BegTime, EndTime» в запрос по услугам ([13]).
- Исправлена ошибка запроса состава услуг. Некоторые услуги без состава не попадали в выборку ([13]).

- Версия: 13.26

- Исправлена ошибка запроса создания заявки ([2]). Неправильно обрабатывались вложенные json объекты в теле запроса).
- Добавлено поле «Balance» текущий остаток товара ([6]).
- Исправлена ошибка сворачивания услуг в запросе операций магазина ([103]). Иногда неправильно округлялось количество товара из нескольких партий в услуге.
- Исправлена ошибка запроса резервуаров ([11]). Иногда в режиме сервера происходила попытка подключения к офисной базе.

- Версия: 13.27

- Добавлена проверка наличия запущенного процесса сервера в запросе резервуаров [11] в режиме сервера.
- Исправлена ошибка распределения дополнительных кодов в запросе списка товаров [6]