



**АвтоРесурс**  
Интернет-магазин запчастей



## REST API



Для системы «Веб-АвтоРесурс» версии 6.46.0 и выше



**Трэйдсофт**

tradesoft.ru

## Содержание

<b>1. Общая информация</b>	<b>5</b>
<b>2. Запросы</b>	<b>5</b>
2.1. Токены REST API v1	5
2.2. Формат запросов REST API v1	5
2.3. Формат запросов REST API v2	5
2.4. Методы запросов	5
2.5. Результаты запроса	6
<b>3. Особенности обработки данных</b>	<b>6</b>
3.1. Гео-поля	6
<b>4. Сущности, для которых реализован REST API v1</b>	<b>6</b>
4.1. API клиенты	6
4.1.1. Создание/редактирование клиента	6
4.1.2. Получение информации о клиенте (клиентах)	8
4.1.3. Удаление информации о клиенте (клиентах)	8
4.2. API категории клиентов	8
4.2.1. Создание категории клиента	8
4.2.2. Получение информации о категориях	9
4.2.3. Удаление информации о категории	9
4.3. API группы клиентов	9
4.3.1. Создание/редактирование группы	9
4.3.2. Получение информации о группах	9
4.3.3. Удаление информации о группах	9
4.4. API автомобили клиентов	9
4.4.1. Создание/редактирование автомобиля клиента	9
4.4.2. Получение информации об автомобилях	10
4.4.3. Удаление информации об автомобилях	11
4.5. API плательщики клиентов	11
4.5.1. Создание/редактирование плательщика	11
4.5.2. Получение информации о плательщиках	11
4.5.3. Удаление информации о плательщиках	11
4.6. API адреса доставки клиентов	12
4.6.1. Создание/редактирование адреса доставки	12
4.6.2. Получение информации об адресах доставки	12
4.6.3. Удаление информации об адресах доставки	12
4.7. API балансы клиентов	12
4.7.1. Обновление баланса	12
4.7.2. Получение информации о финансовых документах	13
4.8. API заказы клиентов	13
4.8.1. Создание/редактирование заказа	13
4.8.2. Получение информации о заказах	14
4.8.3. Удаление информации о заказе (заказах)	14
4.9. API позиций заказов клиентов	14
4.9.1. Создание/редактирование позиции заказа клиента	14
4.9.2. Получение информации о позициях заказов	16
4.9.3. Удаление информации о позиции заказа клиента	16

4.10. API позиции прайс-листов (остатков).....	16
4.10.1. Загрузка остатков .....	16
4.11. API поиск по коду .....	17
4.11.1. Получение списков брендов .....	17
4.11.2. Получение списка доступных к заказу деталей по артикулу и бренду .....	18
4.12. API производители.....	18
4.12.1. Создание/редактирование производителя.....	19
4.12.2. Получение информации о производителях.....	19
4.12.3. Удаление информации о производителях.....	19
4.13. API справочник деталей .....	19
4.13.1. Создание/редактирование детали .....	20
4.14. API справочника аналогов и ошибочных аналогов.....	20
4.14.1. Создание/редактирование аналога.....	20
4.14.2. Получение информации об аналогах.....	20
4.15. API валюты .....	21
4.15.1. Создание/редактирование валюты .....	21
4.15.2. Получение информации о валютах.....	21
4.16. API курсы валют .....	21
4.16.1. Создание курса валют .....	21
4.16.2. Получение информации о курсах валют.....	21
<b>5. Сущности, для которых реализован API V2 .....</b>	<b>22</b>
5.1. API токена для клиентских запросов.....	22
5.1.1. Получение информации о токене клиентских запросов.....	22
5.2. API клиенты .....	22
5.2.1. Создание клиента (регистрация).....	22
5.2.2. Редактирование данных клиента .....	23
5.2.3. Получение информации о данных клиента .....	23
5.3. API финансовые документы.....	24
5.3.1. Получение информации о финансовых документах клиента .....	24
5.4. API заказы.....	24
5.4.1. Получение информации о заказах клиента .....	24
5.5. API позиции заказов клиентов .....	25
5.5.1. Получение информации о позициях заказов клиента .....	25
5.6. API оформление заказа.....	25
5.6.1. Создание нового заказа.....	25
5.7. API корзина .....	25
5.7.1. Удаление корзины клиента .....	25
5.8. API позиции в корзине .....	26
5.8.1. Получение информации о позициях в корзине клиента.....	26
5.8.2. Редактирование позиции в корзине клиента .....	26
5.8.3. Удаление позиции из корзины клиента .....	26
5.9. API торговые точки.....	26
5.9.1. Получение информации о торговых точках.....	26
5.10. API менеджеры торговых точек .....	27
5.10.1. Получение информации о менеджерах торговых точек.....	27
5.11. API настройки системы.....	27

5.11.1. Получение настроек системы .....	27
5.12. API страницы сайта.....	27
5.12.1. Получение контентных страниц сайта .....	28
<b>6. Пример работы с REST API на PHP .....</b>	<b>29</b>
<b>7. Изменения в документации к модулю REST API .....</b>	<b>30</b>

# 1. Общая информация

**REST API** (в дальнейшем API) предназначен для обмена информацией с внешними системами (1С, другие сайты (v1), мобильное приложение (v2) и т.д.). Для использования API внешняя система должна посылать http-запросы на определённый url сайта с определённым набором параметров.

## 2. Запросы

Данные для запроса (для **POST** и **PUT** методов) должны передаваться в **json-формате** с заголовком "Content-Type": "application/json".

### 2.1. Токены REST API v1

Каждый запрос rest api v1 должен содержать GET-параметр **token**.

Создать токен можно в административной части системы «Веб-АвтоРесурс» в разделе **Настройки** ► **Настройки REST API**;

Примечание	Раздел <b>Настройки REST API</b> исходно доступен только для пользователей группы <b>Администраторы сайта</b> .
------------	---

Система позволяет использовать один или несколько токенов.

Добавление токена производится с помощью кнопки **Добавить** (Рис. 1).

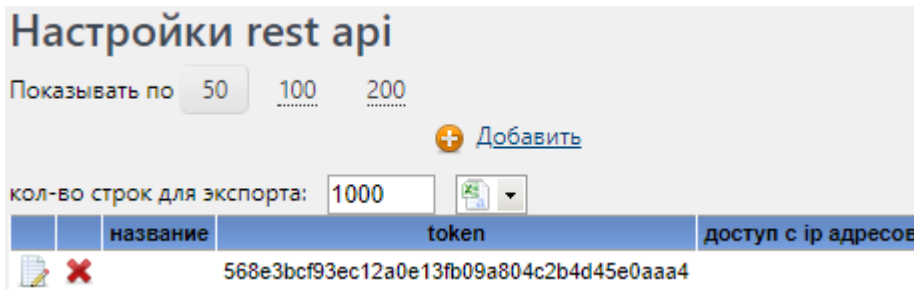


Рис. 1. Список текущих токенов системы.

Неиспользуемые токены удаляются с помощью кнопки **Удалить** **X** в строке соответствующего токена.

### 2.2. Формат запросов REST API v1

Запросы предназначены для работы с основными сущностями сайта.

Запросы посылаются на url вида

**http://www.domain.ru/api/v1/entity/[entityID]/[?token=some\_token&some\_filter=some\_value&some\_filter2=some\_value2]**

Пример	Ссылка запроса для получение информации о клиенте с id=2: <a href="http://www.domain.ru/api/v1/customers/2/?token=1ae6682c9619f22229d453df44ed26022fada053">http://www.domain.ru/api/v1/customers/2/?token=1ae6682c9619f22229d453df44ed26022fada053</a> Ссылка запроса для получения списка клиентов менеджера managerLogin=manager: <a href="http://www.domain.ru/api/v1/customers/?token=1ae6682c9619f22229d453df44ed26022fada053&amp;managerLogin=manager">http://www.domain.ru/api/v1/customers/?token=1ae6682c9619f22229d453df44ed26022fada053&amp;managerLogin=manager</a>
--------	---

### 2.3. Формат запросов REST API v2

Запросы предназначены для работы с основными сущностями клиентской части в рамках мобильного приложения.

Запросы посылаются на url вида

**http://www.domain.ru/api/v2/entityGroup/entity/[entityID]/[?some\_filter=some\_value&some\_filter2=some\_value2]**

### 2.4. Методы запросов

- **GET** - получение информации о сущностях (клиенты, адреса доставки и т.д.). Есть возможность фильтрации данных;

Внимание	Если выборка для <b>GET</b> запроса получилась больше <b>1000 строк</b> , то в ответе также придёт указатель на следующую строку <b>nextStartRow</b> - это значение соответственно нужно передать в следующем запросе (если нужно получить больше данных) в GET-параметре <b>startRow</b> .
	<ul style="list-style-type: none"><li>• <b>POST</b> - создание сущности;</li><li>• <b>PUT</b> - обновление сущности (возможно использование и POST-метода для этого, если в массиве данных указано поле с id сущности);</li><li>• <b>DELETE</b> - удаление сущности.</li></ul>
Внимание	Если в описании сущности, не описан какой-то метод запросов, то его поведение не определено и может привести к падению проекта и потере данных.

## 2.5. Результаты запроса

Результатом обработки данных будет **JSON-массив**, содержащий информацию:

- **'error'** - технический код ошибки (например, `invalid_token`);
- **'errorMessage'** - текстовое описание ошибки (например, "Неправильный токен");
- **'result'** - результат (например, при создании/обновлении сущности здесь будет id сущности в системе, при получении информации - массив сущностей с их полями).

# 3. Особенности обработки данных

## 3.1. Гео-поля

В системе «Веб-АвтоРесурс» привязка сущностей идёт только к полю город и, хотя поля `country/region/city/district` в запросе являются текстовыми, система будет искать соответствие им во внутренней гео-базе (справочник географии), записывая найденный идентификатор города в карточку сущности. Соответственно, если в справочнике город по указанной связке `country/region/city/district` не найден, то запрос вернёт ошибку.

# 4. Сущности, для которых реализован REST API v1

- Клиенты (раздел 4.1);
- Категории клиентов (раздел 4.2);
- Группы клиентов (раздел 4.3);
- Автомобили клиентов (раздел 4.4);
- Плательщики клиентов (раздел 4.5);
- Адреса доставки клиентов (раздел 4.6);
- Балансы клиентов (раздел 4.7);
- Заказы клиента (раздел 4.8);
- Позиции заказов клиентов (раздел 4.9);
- Позиции прайс-листов (раздел 4.10);
- Поиск по коду (раздел 4.11);
- Производители (раздел 4.12);
- Справочник деталей (раздел 4.13);
- Аналоги и ошибочные аналоги (раздел 4.14);
- Валюты (раздел 4.15);
- Курсы валют (раздел 4.16).

## 4.1. API клиенты

`/api/v1/customers/[customerID/]`

### 4.1.1. Создание/редактирование клиента

Для создания клиента используется метод **POST** без указания поля **customerID** в данных запроса.

Для обновления данных клиента используется метод **PUT/POST** с указанием поля **customerID** в данных запроса.

Список принимаемых параметров (если требуется создать несколько клиентов, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
customerID	нет	int(11)	id клиента (только для обновления информации)
customerExternalID	нет	varchar(255)	внешний id клиента
userType	нет	tinyint(1)	тип регистрации: 1 - физическое лицо, 2 - юр. лицо
managerID	нет	int(11)	id менеджера клиента. Является более приоритетным нежели office
managerExternalID	нет	varchar(255)	внешний id менеджера клиента. Является более приоритетным нежели office
managerLogin	нет	varchar(255)	логин менеджера клиента. Является более приоритетным нежели office
officeID	нет	int(11)	id торговой точки клиента. Если не указан manager - устанавливается менеджер по умолчанию для выбранной торговой точки. Если введен некорректный идентификатор менеджера для выбранной торговой точки - возвращается ошибка
officeExternalID	нет	varchar(255)	внешний id торговой точки клиента. Если не указан manager - устанавливается менеджер по умолчанию для выбранной торговой точки. Если введен некорректный идентификатор менеджера для выбранной торговой точки - возвращается ошибка
officeName	нет	varchar(50)	название торговой точки клиента. Если не указан manager - устанавливается менеджер по умолчанию для выбранной торговой точки. Если введен некорректный идентификатор менеджера для выбранной торговой точки - возвращается ошибка
categoryID	нет	int(11)	id категории клиента
categoryExternalID	нет	varchar(255)	внешний id категории клиента
categoryName	нет	varchar(80)	название категории клиента
name	да* (если не указано поле organizationName)	varchar(50)	имя клиента
secondName	нет	varchar(50)	отчество клиента
surname	нет	varchar(50)	фамилия клиента
fullName	нет	varchar(152:50+1+50+1+50)	полное имя клиента в формате ФИО. При создании или обновлении данных клиента поле fullName: <ul style="list-style-type: none"> <li>используется, если не заданы поля name, secondName, surname;</li> <li>будет автоматически разбито по пробелам на поля фамилии, имени и отчества соответственно.</li> </ul>
login	да	varchar(50)	логин клиента
password	нет	varchar(50)	пароль клиента (при создании/изменении)
passwordHash	нет	varchar(32/40)	хеш пароля клиента (при создании/изменении вместо password)
phone	да	varchar(20)	контактный телефон
mobilePhone	нет	varchar(20)	мобильный телефон
email	нет	varchar(50)	e-mail
icq	нет	varchar(50)	ICQ
skype	нет	varchar(50)	Skype
country	нет	varchar(50)	страна
region	нет	varchar(50)	регион/область
city	нет	varchar(50)	город/район
district	нет	varchar(50)	район/нас. пункт
organizationName	да* (если не указано поле name)	varchar(255)	название организации для юр. лиц
businessID	нет	int(11)	id вида деятельности

businessExternalID	нет	varchar(255)	внешний id вида деятельности
businessName	нет	varchar(255)	название вида деятельности
creditLimit	нет	decimal(20,2)	кредитный лимит
managerComment	нет	varchar(255)	комментарий менеджера
contragentType	нет	tinyint(1)	тип контрагента (справочник): 1 - клиент, 2 - торговая точка, 3 - поставщик (типы по-умолчанию)
organizationFormID	нет	int(4)	ID ОПФ
organizationFormExternalID	нет	varchar(255)	внешний ID ОПФ
organizationFormName	нет	varchar(50)	название ОПФ
groupID	нет	tinyint(1)	id группы клиента
groupExternalID	нет	varchar(255)	внешний id группы клиента
groupName	нет	varchar(255)	название группы клиента
preferredCurrencyID	нет	int(11)	ID валюты
preferredCurrencyExternalID	нет	varchar(255)	внешний id валюты
preferredCurrencyISO	нет	varchar(10)	ISO валюты

#### Внимание

При передаче параметра **sendNotifies=1** в get-параметрах, создаваемым клиентам производится рассылка писем о регистрации.

#### 4.1.2. Получение информации о клиенте (клиентах)

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям, а также дополнительно по следующим критериям:

Параметр	Описание
dateRegistredStart	Начальная дата регистрации в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
dateRegistredEnd	Конечная дата регистрации в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
dateUpdatedStart	Начальная дата последнего обновления в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
dateUpdatedEnd	Конечная дата последнего обновления в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
officeMode	Тип торговой точки, к которой привязан клиент (0 - филиал ГО, 1 - дилер)

Результатом будет выборка по клиентам в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности (кроме поля пароля - оно хранится в базе в необратимом виде, поэтому его получение невозможно), а также дополнительно:

Параметр	Тип	Описание
balance	decimal(15,2)	Баланс клиента
deliveryAddresses	array	Адреса доставки (описание полей см. в соответствующем разделе). Возвращается при передаче GET параметра withDeliveryAddresses
payers	array	Плательщики клиента (описание полей см. в соответствующем разделе). Возвращается при передаче GET параметра withPayers
orders	array	Заказы клиента (описание полей см. в соответствующем разделе). Возвращается при передаче GET параметра withOrders

#### 4.1.3. Удаление информации о клиенте (клиентах)

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

### 4.2. API категории клиентов

/api/v1/categories/[categoryID/]

#### 4.2.1. Создание категории клиента

Для создания используется метод **POST** без указания поля **categoryID** в данных запроса.

Для обновления данных используется метод **PUT/POST** с указанием поля **categoryID** в данных запроса.



Список принимаемых параметров (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
categoryID	да (при редактировании)	int(11)	id категории
categoryExternalID	нет	varchar(255)	внешний id категории
categoryName	да (при создании)	varchar(80)	название категории
percent	нет	decimal(5,2)	наценка/скидка категории
default	нет	tinyint(1)	категория по-умолчанию
registrationDefault	нет	tinyint(1)	категория по-умолчанию при регистрации
minOrderSum	нет	decimal(9,2)	минимальная сумма заказа
debtSum	нет	decimal(9,2)	максимальная сумма долга
basketLifeTime	нет	int(11)	время жизни корзины для клиентов данной категории в днях

#### 4.2.2. Получение информации о категориях

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.

#### 4.2.3. Удаление информации о категории

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

### 4.3. API группы клиентов

/api/v1/customerGroups/[groupID/]

#### 4.3.1. Создание/редактирование группы

Для создания используется метод **POST** без указания поля **groupID** в данных запроса. Для обновления данных используется метод **PUT/POST** с указанием поля **groupID** в данных запроса. Список принимаемых параметров (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
groupID	да (при редактировании)	tinyint(3)	ID группы
groupExternalID	нет	varchar(255)	внешний ID группы
groupName	да (при создании)	varchar(255)	название группы
default	нет	tinyint(1)	группа по-умолчанию
registrationDefault	нет	tinyint(1)	группа по-умолчанию при регистрации

#### 4.3.2. Получение информации о группах

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.

#### 4.3.3. Удаление информации о группах

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

### 4.4. API автомобили клиентов

/api/v1/customers/[customerID]/cars/[carID/]

#### 4.4.1. Создание/редактирование автомобиля клиента

Для создания используется метод **POST** без указания поля **carID** в данных запроса.

Для обновления данных используется метод **PUT/POST** с указанием поля **carID** в данных запроса. Список принимаемых параметров (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
customerID	да	int(11)	id клиента, для которого создаётся автомобиль
customerExternalID	нет	varchar(255)	внешний id клиента, для которого создаётся автомобиль
carID	нет	int(11)	id автомобиля (только для обновления информации)
carExternalID	нет	varchar(255)	внешний id автомобиля
brandID	да (brandID или randName при создании)	int(11)	id марки авто
brandName	да (brandID или randName при создании)	varchar(255)	название марки авто
modelID	нет	int(11)	id модели авто
modelName	нет	varchar(255)	название модели авто
modificationID	нет	int(11)	id модификации авто
modificationName	нет	varchar(255)	название модификации авто
year	нет	varchar(4)	год выпуска
body	нет	varchar(50)	тип кузов
driveType	нет	varchar(30)	привод
enginePower	нет	int(11)	мощность двигателя
engineVolume	нет	decimal(9,1)	объём двигателя
engineCylindersCount	нет	tinyint(1)	кол-во цилиндров
engineValvesCount	нет	tinyint(1)	кол-во поршней
engineType	нет	varchar(30)	тип двигателя
engineCode	нет	varchar(50)	код двигателя
fuelType	нет	varchar(30)	тип топлива
fuelSupply	нет	varchar(30)	тип подачи топлива
brakeSystem	нет	varchar(30)	тип тормозной системы
brakeFront	нет	varchar(30)	тип передних тормозов
brakeRear	нет	varchar(30)	тип задних тормозов
VIN	нет	varchar(50)	VIN
steeringWheel	нет	varchar(1)	руль (L-левый, R-правый)
transmissionType	нет	varchar(30)	тип КПП
transmissionCode	нет	varchar(50)	номер КПП
transmissionCount	нет	tinyint(1)	число передач
catalystType	нет	varchar(50)	тип катализатора
ABS	нет	tinyint(1)	наличие ABS
ESP	нет	tinyint(1)	наличие ESP
powerSteering	нет	tinyint(1)	наличие усилителя руля
conditioner	нет	tinyint(1)	наличие кондиционера
catalyst	нет	tinyint(1)	наличие катализатора
turbo	нет	tinyint(1)	наличие турбо
ASR	нет	tinyint(1)	наличие ASR
ASD	нет	tinyint(1)	наличие ASD
ETS	нет	tinyint(1)	наличие ETS
doorsCount	нет	tinyint(1)	кол-во дверей
country	нет	varchar(40)	страна
registrationNumber	нет	varchar(50)	регистрационный (государственный) номер
additionalInfo	нет	text	доп. информация

#### 4.4.2. Получение информации об автомобилях

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

Дополнительные фильтры:

Параметр	Описание
showMode	режим вывода: отсутствие параметра/значения - все, deleted - только удалённые, actual - только актуальные

#### 4.4.3. Удаление информации об автомобилях

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

### 4.5. API плательщики клиентов

`/api/v1/customers/{customerID}/payers/{payerID/}`

#### 4.5.1. Создание/редактирование плательщика

Для создания используется метод **POST** без указания поля **payerID** в данных запроса.

Для обновления данных используется метод **PUT/POST** с указанием поля **payerID** в данных запроса.

Список принимаемых параметров (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
customerID	да	int(11)	id клиента, для которого создаётся плательщик
customerExternalID	нет	varchar(255)	внешний id клиента, для которого создаётся плательщик
payerID	нет	int(4)	id плательщика (только для обновления информации)
payerExternalID	нет	varchar(255)	внешний id плательщика (только для обновления информации)
payerName	да	varchar(255)	название плательщика
organizationFormID	нет	int(4)	ID ОПФ
organizationFormExternalID	нет	varchar(255)	внешний ID ОПФ
organizationFormName	нет	varchar(50)	название ОПФ
address	нет	varchar(255)	адрес
legalAddress	нет	varchar(255)	юр. адрес
inn	нет	varchar(20)	ИНН
kpp	нет	varchar(20)	КПП
bankName	нет	varchar(50)	Название банка
bik	нет	varchar(20)	БИК
ks	нет	varchar(30)	корр. счёт
rs	нет	varchar(50)	расчётный счёт
okpo	нет	varchar(255)	ОКПО
ogrn	нет	varchar(20)	ОГРН

#### 4.5.2. Получение информации о плательщиках

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.

Дополнительные фильтры:

Параметр	Описание
showMode	режим вывода: отсутствие параметра/значения - все, deleted - только удалённые, actual - только актуальные

#### 4.5.3. Удаление информации о плательщиках

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

## 4.6. API адреса доставки клиентов

`/api/v1/customers/{customerID}/deliveryAddresses/{deliveryAddressID/}`

### 4.6.1. Создание/редактирование адреса доставки

Для создания используется метод **POST** без указания поля **deliveryAddressID** в данных запроса.  
Для обновления данных используется метод **PUT/POST** с указанием поля **deliveryAddressID** в данных запроса.  
Список принимаемых параметров (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
customerID	да	int(11)	id клиента, для которого создаётся адрес доставки
customerExternalID	нет	varchar(255)	внешний id клиента, для которого создаётся адрес доставки
deliveryAddressID	нет	int(11)	id адреса доставки (только для обновления информации)
deliveryAddressExternalID	нет	varchar(255)	внешний id адреса доставки
destinationName	да	varchar(255)	ФИО / название организации получателя
country	нет	varchar(50)	страна
region	нет	varchar(50)	регион/область
city	нет	varchar(50)	город/район
district	нет	varchar(50)	район/нас. пункт
deliveryAddress	нет	varchar(100)	адрес доставки
postCode	нет	varchar(10)	почтовый индекс
information	нет	text	доп. информация

### 4.6.2. Получение информации об адресах доставки

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.

Дополнительные фильтры:

Параметр	Описание
showMode	режим вывода: отсутствие параметра/значения - все, deleted - только удалённые, actual - только актуальные

### 4.6.3. Удаление информации об адресах доставки

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

## 4.7. API балансы клиентов

`/api/v1/customers/{customerID}/balances/`

### 4.7.1. Обновление баланса

Используется метод **POST/PUT**.  
Список принимаемых параметров (если требуется изменить балансы нескольких клиентов, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
customerID	да	int(11)	id клиента
customerExternalID	нет	varchar(255)	внешний id клиента
sum	да/нет	decimal(20,2)	сумма поступления (положительная) / списания (отрицательная)
newBalance	да/нет	decimal(15,2)	новый баланс клиента
paymentTypeID	нет	int(4)	ID вида расчёта (справочник ВАР)
paymentTypeExternalID	нет	varchar(255)	внешний ID вида расчёта (справочник ВАР)
paymentTypeName	нет	varchar(50)	Название вида расчёта (справочник ВАР)
comment	нет	varchar(255)	комментарий

dateTime	нет	datetime: ГГГГ-ММ-ДД ЧЧ:ММ:СС	дата создания документа
currencyID	да	int(11)	ID валюты
currencyISO	да	varchar(10)	ISO валюты

Примечание Обязательно наличие полей **sum** или **newBalance**. При одновременной передаче обоих параметров возвращается ошибка.

При передаче **newBalance** формируется документ списания/поступления на сумму равную разнице текущего баланса клиента в системе «Веб-АвтоРесурс» и передаваемого значения **newBalance**.

Параметры **currencyID** и **currencyISO** доступны только для мультивалютного режима. При передаче этих параметров не в мультивалютном режиме возвращается ошибка. При мультивалютном режиме и отсутствующих параметрах **currencyID** и **currencyISO** операция производится в основной валюте сайта.

#### 4.7.2. Получение информации о финансовых документах

Используется метод **GET**.

Результатом будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Тип	Описание
documentID	bigint(20)	id документа (внутренний) (справочник ВАР)
documentExternalNumber	varchar(255)	id документа (внешний)
customerID	int(11)	id клиента (справочник ВАР)
dateTime	datetime: ГГГГ-ММ-ДД ЧЧ:ММ:СС	дата создания документа
documentTypeID	tinyint(4)	ID типа документа (справочник ВАР)
documentTypeName	varchar(255)	Название типа документа (справочник ВАР)
paymentTypeID	int(4)	ID вида расчёта (способа оплаты) (справочник ВАР)
paymentTypeName	varchar(50)	Название вида расчёта (способа оплаты) (справочник ВАР)
sum	decimal(20,2)	Сумма документа
comment	varchar(255)	Комментарий

Дополнительные фильтры:

Параметр	Описание
dateTimeStart	Начальная дата финансовых документов ГГГГ-ММ-ДД ЧЧ:ММ:СС
dateTimeEnd	Конечная дата финансовых документов ГГГГ-ММ-ДД ЧЧ:ММ:СС

### 4.8. API заказы клиентов

/api/v1/customers/[customerID]/orders/[orderID/]

#### 4.8.1. Создание/редактирование заказа

Для создания клиента используется метод **POST** без указания поля **orderID** в данных запроса.

Для обновления данных заказа используется метод **PUT/POST** с указанием поля **orderID** в данных запроса.

Список принимаемых параметров (если требуется создать несколько заказов, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
customerID	да (при создании)	int(11)	id клиента
customerExternalID	нет	varchar(255)	внешний id клиента
orderID	да (при обновлении)	int(11)	id заказа (для обновления информации)
orderExternalID	нет	varchar(255)	внешний id заказа
orderExternalNumber	нет	varchar(255)	внешний номер заказа
paymentTypeID	нет	int(4)	id типа оплаты (справочник)
paymentTypeExternalID	нет	varchar(255)	внешний id типа оплаты
paymentTypeName	нет	varchar(50)	название типа оплаты (справочник)

deliveryTypeID	нет	int(4)	id типа доставки (справочник)
deliveryTypeExternalID	нет	varchar(255)	внешний id типа доставки (справочник)
deliveryTypeName	нет	varchar(50)	название типа доставки (справочник)
deliveryAddressID	нет	int(11)	id адреса доставки (справочник)
deliveryAddressExternalID	нет	varchar(255)	внешний id адреса доставки
deliveryOrderAddress	нет	varchar(255)	произвольный текстовый адрес доставки
phone	нет	varchar(100)	телефон (если отсутствует - будет использован контактный телефон из карточки клиента)
email	нет	varchar(50)	e-mail (если отсутствует - будет использован e-mail из карточки клиента)
comment	нет	text	комментарий к заказу клиента
dateTime	нет	datetime	дата оформления заказа в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
currencyID	нет	int(11)	ID валюты
currencyExternalID	нет	varchar(255)	внешний ID валюты
currencyISO	нет	varchar(10)	ISO валюты

#### 4.8.2. Получение информации о заказах

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям, а также дополнительно по следующим критериям:

Параметр	Описание
dateOrderedStart	Начальная дата оформления заказа в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
dateOrderedEnd	Конечная дата оформления заказа в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
dateUpdatedStart	Начальная дата последнего обновления в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
dateUpdatedEnd	Конечная дата последнего обновления в формате ГГГГ-ММ-ДД ЧЧ:мм:СС

Результатом будет выборка по заказам в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности, а также дополнительно:

Параметр	Тип	Описание
orderNumber	varchar(255)	Внутренний номер заказа в Веб-АвтоРесурсе
positionsQuantity	int(11)	Кол-во позиций в заказе
positionsAmount	int(11)	Кол-во деталей в заказе
sum	decimal(20,2)	Сумма по заказу
payedSum	decimal(20,2)	Оплаченная сумма по заказу
debtSum	decimal(20,2)	Сумма долга по заказу
contragentType	tinyint(1)	Тип контрагента заказа (1 - клиент, 2 - ТТ)
officeMode	tinyint(1)	Тип торговой точки, к которой привязан клиент (0 - филиал ГО, 1 - дилер)
positions	array	Массив позиций заказа (описание полей см. в соответствующем разделе). Возвращается при передаче GET параметра withPositions

#### 4.8.3. Удаление информации о заказе (заказах)

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

### 4.9. API позиций заказов клиентов

**/api/v1/customers/[customerID]/orders/[orderID]/positions/[positionID/]**

#### 4.9.1. Создание/редактирование позиции заказа клиента

Для создания позиции заказа клиента используется метод **POST** без указания поля **positionID** в данных запроса. Для обновления данных используется метод **PUT/POST** с указанием поля **positionID** в данных запроса. Список принимаемых параметров для создания/редактирования сущности (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
orderID	да (при создании)	int(11)	id заказа, для которого создается позиция
orderExternalID	нет	varchar(255)	внешний id заказа, для которого создается позиция
positionID	да (при обновлении)	int(11)	id позиции (для режима обновления)
positionExternalID	нет	varchar(255)	внешний id позиции
orderHash	нет	varchar	хеш для заказа, полученный через сущность проценки
brand	да (при создании)	varchar(30)	производитель детали
article	да (при создании)	varchar(30)	артикул детали
description	нет	varchar(255)	описание детали (наименование)
supplierID	нет	int(11)	id поставщика (справочник)
supplierExternalID	нет	varchar(255)	внешний id поставщика (справочник)
supplierName	нет	varchar(255)	название поставщика (справочник)
amount	нет	int(11)	кол-во деталей
price	нет	decimal(20,2)	цена за штуку
supplierPrice	нет	decimal(20,2)	закупочная цена за штуку
comment	нет	varchar(255)	комментарий клиента к позиции
managerComment	нет	varchar(255)	комментарий менеджера к позиции
statusID	нет	tinyint(4)	id состояния позиции (справочник)
statusExternalID	нет	varchar(255)	внешний id состояния позиции (справочник)
statusName	нет	varchar(50)	название состояния позиции (справочник)
term	нет	tinyint(3)	срок поставки (средний)
maxTerm	нет	tinyint(3)	гарантированный срок поставки
arrivalDate	нет	datetime	дата поставки
destination	нет	varchar(255)	направление
dateTime	нет	datetime	дата добавления позиции в формате ГГГГ-ММ-ДД ЧЧ:мм:СС
articleFromSupplier	нет	varchar(60)	артикул от поставщика
brandFromSupplier	нет	varchar(60)	производитель от поставщика

Для обновления статусов позиций (в частности деления-объединения позиций) допустимо посылать информацию об этом в одной сущности позиции с подмассивом состояний в указанном формате в поле **'subStates'**, например:

```
'subStates' => [[
    'amount' => 3,
    'statusID' => 2
  ], [
    'amount' => 2,
    'statusID' => 3
  ]
]

'subStates' => [[
    'amount' => 3,
    'statusExternalID' => 2
  ], [
    'amount' => 2,
    'statusExternalID' => 3
  ]
]
```

При этом, требуется указание поля **positionID** в данных запроса (в случае первого деления - исходный **positionID** позиции, в случае повторного деления reference позиции, т.к. для неразделенной позиции ref = id, а для разделённой ref'ом является ref исходной позиции, при этом, id будет отличаться, т.к. данный параметр уникален для каждой отдельной позиции. При работе с разделёнными позициями в качестве id необходимо использовать ref номер исходной позиции).

**Внимание**

Каждый раз необходимо передавать полный набор **'subStates'** по разделенным позициям, чтобы общее кол-во позиций в БД, определенных по Ref-номеру позиции, совпадала с общим кол-вом, пришедшим в запросе. Например, если в БД находится 5 позиций, то в **'subStates'** может быть 5, или 3+2, или 1+2+2, но не 2+2.

#### 4.9.2. Получение информации о позициях заказов

Для получения информации о позиции заказа клиента используется метод **GET**. В get-параметрах можно дополнительно задать фильтрацию по описанным в таблице ниже полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности, а также дополнительно:

Параметр	Тип	Описание
reference	int(11)	референс-номер позиции
sum	decimal(20,2)	сумма по позиции
payedSum	decimal(20,2)	оплаченная сумма по позиции
debtSum	decimal(20,2)	сумма долга по позиции
currencyID	int(11)	ID валюты
currencyExternalID	varchar(255)	внешний ID валюты
currencyISO	varchar(10)	ISO валюты
supplierCustomerID	int(11)	id контрагента поставщика
supplierCustomerExternalID	varchar(255)	внешний id контрагента поставщика
supplierOfficeID	smallint(5)	id торговой точки (в случае если позиция от поставщика, контрагент которого является контрагентом торговой точки)
supplierOfficeExternalID	varchar(255)	внешний id торговой точки (в случае если позиция от поставщика, контрагент которого является контрагентом торговой точки)

#### 4.9.3. Удаление информации о позиции заказа клиента

Для удаления информации о позиции заказа клиента используется метод **DELETE**. В get-параметрах можно дополнительно задать фильтрацию по описанным в таблице ниже полям.

### 4.10. API позиции прайс-листов (остатков)

/api/v1/pricelists/[pricelistsID]/wares/

#### 4.10.1. Загрузка остатков

Добавляет/обновляет в системе данные о товарных остатках из присланного массива позиций в прайс-листе, ID которого передан в url обращения (pricelistsID).

**Внимание**

За один запрос может быть загружено ограниченное количество позиций (максимальное количество зависит от характеристик и настроек сервера).

Управляющие параметры запроса (GET-параметры):

Параметр	Описание	Значения	Примечание
updateType	Режим выгрузки	"1" - Режим полной перезаписи, "2" - Режим изменения\дополнения.	Если режим выгрузки не задан - возвращается ошибка загрузки прайс-листа
finalPart	Признак конца прайс-листа (логическое поле)	"0" – неконечный пакет данных прайс-листа, "1" - конечный пакет данных прайс-листа.	Поле используется для режима полной перезаписи - остатки на сайте фактически заменяются только при получении запроса с параметром "1". При отсутствии поля – считается, что передан "1", т.е. весь прайс передан в одном запросе.



Примечание	<p>При обновлении прайс листа в режиме изменения (GET параметр <b>updateType=2</b>) - временные таблицы на сайте не создаются. Все найденные позиции по связке <b>артикул-бренд</b> обновляются по количеству и цене, если они были изменены. Не найденные в прайс-листах позиции дописываются в остатки. Позиции, которых в выгрузке не было - остаются без каких-либо изменений.</p> <p>Необходимо обратить внимание на следующее замечание: т.к. обновление идёт по связке <b>артикул-бренд</b>, то возможны следующие ситуации:</p> <p>1) в существующем прайс-листе есть несколько предложений по связке <b>артикул-бренд</b> (например, с разными ценами и/или сроками). При обновлении данных будет обновлена только первая позиция прайс-листа для этой связки (загруженная первой).</p> <p>2) в передаваемых данных для обновления присутствуют несколько строк данных по связке <b>артикул-бренд</b>. В этом случае они будут применяться последовательно, т.е. актуальной информацией в прайс-листе будут данные от последней переданной в запросе позиции по данной связке.</p>
------------	---

Примечание	<p>При обновлении прайс-листа в режиме перезаписи (GET параметр <b>updateType=1</b>) - создается временная таблица, в которую загружается постранично информация из загружаемых прайс-листов. Загрузка происходит до того момента, пока параметр конца прайс-листа (GET параметр <b>finalPart</b>) не примет положительное значение (1), после этого - текущие остатки на сайте стираются и заменяются на значения из временной таблицы.</p> <p>В случае, если в течение 3-х дней не был получен пакет, содержащий признак конца - временная таблица удаляется. Текущие остатки на сайте при этом не изменяются.</p>
------------	--

Для передачи данных используется метод **POST/PUT**.

Список принимаемых параметров позиций (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
brand	да	varchar(40)	Производитель
article	да	varchar(50)	Артикул (код детали)
quantity	нет	varchar(20)	Количество товара в наличии
phand	нет	decimal(15,2)	Залог
term	нет	int(11)	Срок (средний)
minPack	нет	int(11)	Минимальная партия
price	да (при добавлении)	decimal(15,2)	Цена поставщика за единицу товара
comment	нет	varchar(80)	Комментарий к товару (наименование товара)
managerComment	нет	varchar(255)	Комментарий менеджера
articleFromProvider	нет	varchar(100)	Артикул поставщика
destination	нет	varchar(255)	Направление
remainsRating	нет	tinyint(3)	Рейтинг наличия

## 4.11. API поиск по коду

**/api/v1/search/**

### 4.11.1. Получение списков брендов

Используется метод **GET** с параметрами:

Параметр	Обязательный	Тип	Описание
code	да	varchar	Искомый код детали. Полученный код системой приводится к «очищенному» виду, содержащему только алфавитные и цифровые символы
useWebSuppliers	нет	enum[0,1]	Флаг «использовать web-поставщиков». Может принимать значения 0 – не использовать или 1 - использовать (по умолчанию - 0). Если выключено, то в выдачу не будут попадать детали от web-поставщиков, что позволит существенно увеличить скорость ответа

Результат запроса:

Параметр	Описание
brand	Наименование производителя
code	Код детали
description	Наименование детали

#### 4.11.2. Получение списка доступных к заказу деталей по артикулу и бренду

Используется метод **GET** с параметрами:

Параметр	Обязательный	Тип	Описание
code	да	varchar	Искомый код детали. Полученный код системой приводится к «очищенному» виду, содержащему только алфавитные и цифровые символы
brand	да	varchar	Производитель детали
useWebSuppliers	нет	enum[0,1]	Флаг «использовать web-поставщиков». Может принимать значения 0 – не использовать или 1 - использовать (по умолчанию - 0). Если выключено, то в выдачу не будут попадать детали от web-поставщиков, что позволит существенно увеличить скорость ответа
useAnalog	нет	enum[0,1]	Флаг «использовать аналоги». Может принимать значения 0 – не использовать или 1 - использовать (по умолчанию - 0)
customerID	нет	int	ID клиента сайта, под которым осуществляется оценка (для получения актуального ценообразования), если параметр отсутствует – то оценка осуществляется под «незарегистрированным пользователем»

Результат запроса:

Параметр	Описание
orderHash	хеш позиции для заказа через сущность позиций
brand	производитель
code	код детали (артикул)
articleFromSupplier	артикул поставщика
brandFromSupplier	производитель поставщика
description	наименование товара
availability	наличие
availabilityRate	оценка наличия
packing	кратность (мин. партия)
deliveryDaysMid	срок доставки средний
deliveryDaysMax	срок доставки максимальный
supplierID	ID поставщика
supplierName	название поставщика
providersPricesID	ID маркетинга
direction	направление поставки
currencyID	ISO код валюты
price	цена в валюте currencyID
weight	вес детали
volume	объем детали
deliveryRate	оценка поставки
deliveryCondition	условие поставки
deliveryWeightTax	стоимость поставки за килограмм
supplierPrice	цена поставщика в валюте currencyID
date	дата обновления прайса

#### 4.12. API производители

/api/v1/brands/[brandID/]

#### 4.12.1. Создание/редактирование производителя

Для создания используется метод **POST** без указания поля **brandID** в данных запроса.

Для обновления данных используется метод **PUT/POST** с указанием поля **brandID** в данных запроса.

Список принимаемых параметров (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
brandID	да (при редактировании)	int(11)	ID производителя
brandExternalID	нет	varchar(255)	внешний ID производителя
brandAIID	нет	varchar(255)	ID производителя в AI (только для синхронизатора с AI)
brandName	да (при создании)	varchar(255)	название производителя
brandFullName	нет	varchar(512)	полное название производителя
oem	нет	tinyint(1)	оригинальный производитель [0-нет, 1-да], по умолчанию 0
info	нет	text	информация о производителе (описание)
country	нет	varchar(50)	страна
concernBrandID	нет	int(11)	ID производителя концерна (если при создании производителя указать в этом поле "-1", то будет считаться что создаётся производитель-концерн, если 0 (или отсутствует), то обычный производитель)
concernBrandExternalID	нет	varchar(255)	внешний ID производителя концерна (если при создании производителя указать в этом поле "-1", то будет считаться что создаётся производитель-концерн, если 0 (или отсутствует), то обычный производитель)
brandAllNames	нет	array of string	другие названия производителя (синонимы)

#### Примечание

Если в запросе присутствует **brandAllNames**, то считается что это всегда полный перечень нужных синонимов - соответственно при обновлении этот массив заменяет текущий (**brandName** и **brandFullName** в нём можно не передавать - эти названия добавляются автоматически к списку синонимов).

Если передать GET параметр **forcedMergeMode=1** при создании/обновлении производителя, то проверка на уникальность имени осуществляться не будет, при этом система сольёт все указанные в параметрах запроса наименования (**brandName** + **brandFullName** + **brandAllNames**) в один бренд (если ранее они относились к нескольким). Связанные с удаляемыми производителями сущности (например, прайс-листы или справочник деталей) будут указывать уже на нового объединённого производителя. Использовать этот режим необходимо с крайней осторожностью.

#### 4.12.2. Получение информации о производителях

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.

Дополнительные фильтры:

Параметр	Описание
brandType	тип бренда. Возможные значения: concern (только производители-концерны), brand (только производители-не концерны), all (все, параметр по-умолчанию, может отсутствовать)

#### 4.12.3. Удаление информации о производителях

Используется метод **DELETE**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям.

### 4.13. API справочник деталей

/api/v1/details/

#### 4.13.1. Создание/редактирование детали

Для создания используется метод **POST** без указания поля **detailID** в данных запроса.

Для обновления данных используется метод **PUT/POST** с указанием поля **detailID** в данных запроса.

Список принимаемых параметров (обязательными данными при создании детали являются артикул (**article**) и производитель (**brandID/brandName**)):

Параметр	Обязательный	Тип	Описание
detailID	да (при обновлении)	int(11)	внутренний id детали в системе
detailExternalID	нет	varchar(255)	внешний id детали
article	нет	varchar(50)	артикул
articleDisplay	нет	varchar(50)	отображаемый артикул
brandID	нет	int(11)	внутренний id производителя в системе
brandExternalID	нет	varchar(255)	внешний id производителя
brandName	нет	varchar(255)	наименование производителя
description	нет	varchar(80)	наименование детали
weight	нет	float	вес
volume	нет	float	объем

#### 4.14. API справочника аналогов и ошибочных аналогов

[/api/v1/details-analogs/](#)

[/api/v1/details-analogs-wrong/](#)

##### 4.14.1. Создание/редактирование аналога

Для создания используется метод **POST** без указания поля **analogID** в данных запроса.

Для обновления данных используется метод **PUT/POST** с указанием поля **analogID** в данных запроса.

Список принимаемых параметров (обязательными данными при создании записи об аналоге являются артикул (**article**) и производитель (**brandID/brandName**) обеих деталей, либо id этих деталей).

Параметр	Обязательный	Тип	Описание
analogID	да (при обновлении)	int(11)	внутренний id аналога в системе
firstDetailID	нет	int(11)	внутренний id детали в системе (первой)
firstDetailExternalID	нет	varchar(255)	внешний id детали в системе (первой)
firstDetailArticle	нет	varchar(50)	артикул детали (первой)
firstDetailBrandID	нет	int(11)	внутренний id производителя детали в системе (первой)
firstDetailBrandExternalID	нет	varchar(255)	внешний id производителя детали (первой)
firstDetailBrandName	нет	varchar(255)	название производителя детали (первой)
secondDetailID	нет	int(11)	внутренний id детали в системе (второй)
secondDetailExternalID	нет	varchar(255)	внешний id детали в системе (второй)
secondDetailArticle	нет	varchar(50)	артикул детали (второй)
secondDetailBrandID	нет	int(11)	внутренний id производителя детали в системе (второй)
secondDetailBrandExternalID	нет	varchar(255)	внешний id производителя детали (второй)
secondDetailBrandName	нет	varchar(255)	название производителя детали (второй)

Примечание

При запросе на создание записи об аналоге по связке артикул+производитель и отсутствии такой записи, она создается автоматически в справочнике деталей.

##### 4.14.2. Получение информации об аналогах

Используется метод **GET**. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.

## 4.15. API валюты

/api/v1/currencies/[currencyID/]

### 4.15.1. Создание/редактирование валюты

Для создания используется метод **POST** без указания поля **currencyID/currencyISO** в данных запроса. Для обновления данных используется метод **PUT/POST** с указанием поля **currencyID/currencyISO** в данных запроса.

Список принимаемых параметров (если требуется создать несколько сущностей, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
currencyID	да (при редактировании)	tinyint(3)	ID валюты
currencyExternalID	нет	varchar(255)	внешний ID валюты
currencyName	да (при создании)	varchar(20)	название валюты
currencyISO	да (при создании)	varchar(10)	ISO код валюты
htmlSign	нет	varchar(10)	подпись валюты в ценах

### 4.15.2. Получение информации о валютах

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.

## 4.16. API курсы валют

/api/v1/currencyRates/

### 4.16.1. Создание курса валют

Для создания клиента используется метод **POST**.

Список принимаемых параметров (если требуется создать несколько курсов, то принимается массив, содержащий в себе несколько сущностей с данными полями):

Параметр	Обязательный	Тип	Описание
targetCurrencyID	да	int(11)	ID валюты, для которой задаётся курс
targetCurrencyExternalID	да	varchar(255)	внешний ID валюты, для которой задаётся курс
targetCurrencyISO	да	varchar(10)	ISO валюты, для которой задаётся курс
measureCurrencyID	да	int(11)	ID валюты, в которой указан курс
measureCurrencyExternalID	да	varchar(255)	внешний ID валюты, в которой указан курс
measureCurrencyISO	да	varchar(10)	ISO валюты, в которой указан курс
rate	да	float	курс

### 4.16.2. Получение информации о курсах валют

Используется метод **GET**, в get-параметрах можно дополнительно задать фильтрацию по описанным выше полям. Результатом будет выборка в виде массива, каждый элемент которого состоит из полей, описанных в разделе создания сущности.


Дополнительные фильтры:

Параметр	Тип	Описание
rateID	int(11)	ID курса
dateStart		Начало действия курса валюты в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС
dateEnd		Окончание действия курса валюты в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС
active	tinyint(1)	активность курса на данный момент

## 5. Сущности, для которых реализован API V2

Актуальную информацию о методах и сущностях можно получить по url вида **ваш\_домен/api/v2/**.

### Примечание

Сущности, отмеченные  на схеме требуют обязательной авторизации с помощью токена (раздел 5.1).

- Токен для клиентских запросов (раздел 5.1);
- Клиенты (раздел 5.2);
- Финансовые документы (раздел 5.3);
- Заказы клиентов (раздел 5.4);
- Позиции заказов клиентов (раздел 5.5);
- Оформление заказа (раздел 5.6);
- Корзина клиента (раздел 5.7);
- Позиции в корзине (раздел 5.8);
- Торговые точки (раздел 5.9);
- Менеджеры торговых точек (раздел 5.10);
- Настройки системы (раздел 5.11);
- Страницы сайта (раздел 5.12).

### 5.1. API токена для клиентских запросов

**/api/v2/client/token/**

#### 5.1.1. Получение информации о токене клиентских запросов

Для создания токена клиентских запросов используется метод **GET**.

Список параметров:

Параметр	Обязательный	Тип	Описание
userlogin	да	string	логин клиента
userpassword	да	string	пароль клиента
token_name	да	string	название формируемого токена (приложения)

Ответом на запрос является значение параметра:

Параметр	Описание
token	значение токена

### 5.2. API клиенты

**/api/v2/client/personal-data/**

#### 5.2.1. Создание клиента (регистрация)

Для создания клиента используется метод **POST**.

Список параметров:

Параметр	Обязательный	Тип	Описание
officeID	да	integer	id торговой точки
name	да	string	имя клиента
secondName	нет	string	отчество клиента
surname	да	string	фамилия клиента
login	да	string	логин клиента
password	да	string	пароль клиента
phone	Да	string	номер телефона клиента
email	да	string	e-mail клиента
cityID	да	integer	id города клиента
districtID	да	integer	id района/округа клиента

Ответом на запрос является значение параметра:

Параметр	Описание
customerID	id клиента

### 5.2.2. Редактирование данных клиента

Для изменения данных клиента используется метод **PUT**.

Примечание	В <b>headers</b> запроса необходимо передавать параметр <b>AuthorizationToken</b> со значением полученного токена (раздел 5.1).
------------	---

Список параметров:

Параметр	Обязательный	Тип	Описание
name	да	string	имя клиента
secondName	нет	string	отчество клиента
surname	да	string	фамилия клиента
password	да	string	пароль клиента
phone	да	string	номер телефона клиента
email	да	string	e-mail клиента
cityID	да	integer	id города клиента
districtID	да	integer	id района/округа клиента

Ответом на запрос является значение параметра:

Параметр	Описание
customerID	id клиента

### 5.2.3. Получение информации о данных клиента

Для создания запроса на получение данных клиента используется метод **GET** без параметров.

Примечание	В <b>headers</b> запроса необходимо передавать параметр <b>AuthorizationToken</b> со значением полученного токена (раздел 5.1).
------------	---

Результат запроса:

Параметр	Описание
customerID	id клиента
managerID	id менеджера клиента
officeID	id торговой точки
officeName	название торговой точки
name	имя клиента
secondName	отчество клиента
surname	фамилия клиента
login	логин клиента
phone	номер телефона клиента
email	e-mail клиента
districtID	id района/округа клиента
districtName	название района/округа клиента
cityID	id города клиента
cityName	название города клиента
regionName	название области клиента
countryName	название страны клиента
balance	состояние счета клиента

В случае ошибки исполнения запроса будут возвращены параметры:

Параметр	Описание
error	название типа ошибки
errorMessage	сообщение об ошибке

## 5.3. API финансовые документы

*/api/v2/client/balance-documents/*

### 5.3.1. Получение информации о финансовых документах клиента

Для создания запроса на получение данных о финансовых документах клиента используется метод **GET** без параметров.

Примечание В **headers** запроса необходимо передавать параметр **AuthorizationToken** со значением полученного токена (раздел 5.1).

Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
documentID	id документа
dateTime	дата и время проведения документа
documentTypeID	id типа документа
documentTypeName	название типа документа (например, списание/поступление платежей)
paymentTypeName	название способа оплаты
sum	сумма платежа
comment	комментарий к документу (например, в счет какого заказа списаны средства)

## 5.4. API заказы

*/api/v2/client/orders/*

### 5.4.1. Получение информации о заказах клиента

Для создания запроса на получение данных о заказах клиента используется метод **GET** без параметров.

Примечание В **headers** запроса необходимо передавать параметр **AuthorizationToken** со значением полученного токена (раздел 5.1).

Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
orderID	id заказа
orderNumber	номер заказа
paymentTypeName	название способа оплаты
deliveryTypeName	способ доставки товара
deliveryOrderAddress	адрес доставки товара
phone	номер телефона клиента
email	e-mail клиента
comment	комментарий к заказу
positionsQuantity	количество товаров в заказе
positionsAmount	количество позиций в заказе
sum	сумма заказа
currencyISO	валюта, используемая в заказе
dateTime	дата оформления заказа



## 5.5. API позиции заказов клиентов

**/api/v2/client/positions/**

### 5.5.1. Получение информации о позициях заказов клиента

Для создания запроса на получение данных о позициях заказов клиента используется метод **GET** без параметров.

Примечание	В <b>headers</b> запроса необходимо передавать параметр <b>AuthorizationToken</b> со значением полученного токена (раздел 5.1).
------------	---

Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
positionID	id позиции заказа
orderID	id заказа
brand	производитель позиции
article	артикул позиции
description	название позиции
amount	количество единиц заказанной позиции
price	цена за единицу позиции
sum	общая стоимость позиции с учетом количества
payedSum	оплаченная по позиции сумма
debtSum	оставшаяся сумма долга по позиции
statusID	id состояния заказа
statusName	состояние заказа
destination	место назначения доставки позиции
term	средний срок поставки позиции
maxTerm	гарантированный срок поставки
arrivaDate	дата и время доставки позиции в место назначения

## 5.6. API оформление заказа

**/api/v2/client/basket-order/**

### 5.6.1. Создание нового заказа

Для создания заказа клиента используется метод **POST**.

Список параметров:

Параметр	Обязательный	Тип	Описание
positionIDs	нет	array of int	id позиций к оформлению
customer	да (при совмещении заказа с регистрацией)	array	данные клиента для регистрации (раздел 5.2.1)

Ответом на запрос являются значения параметров:

Параметр	Описание
success	признак успешности формирования заказа
orderID	id заказа

## 5.7. API корзина

**/api/v2/client/basket/**

### 5.7.1. Удаление корзины клиента

Для создания запроса на очистку корзины используется метод **DELETE** без параметров.

Ответом на запрос является значение параметра:

Параметр	Описание
success	признак успешности очистки корзины

## 5.8. API позиции в корзине

`/api/v2/client/basket-positions/`

### 5.8.1. Получение информации о позициях в корзине клиента

Для создания запроса на получение данных о позициях в корзине клиента используется метод **GET** без параметров. Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
positionID	id позиции, добавленной в корзину
brand	производитель позиции
article	артикул позиции
description	название позиции
amount	количество единиц позиции, добавленное в корзину
price	цена за единицу позиции
sum	общая стоимость позиции с учетом количества
destination	место назначения доставки позиции
term	средний срок поставки позиции
maxTerm	гарантированный срок поставки

### 5.8.2. Редактирование позиции в корзине клиента

Для создания запроса на изменение данных о позиции в корзине клиента используется метод **PUT**.  
Список параметров:

Параметр	Обязательный	Тип	Описание
positionID	да	integer	id позиции, добавленной в корзину
amount	нет	number	количество единиц позиции, добавленное в корзину

Ответом на запрос является значение параметра:

Параметр	Описание
positionID	id позиции в корзине

### 5.8.3. Удаление позиции из корзины клиента

Для создания запроса на удаление позиции из корзины клиента используется метод **DELETE**.  
Список параметров:

Параметр	Обязательный	Тип	Описание
positionID	да	integer	id позиции в корзине

Ответом на запрос является значение параметра:

Параметр	Описание
positionID	id позиции, удаленной из корзины

## 5.9. API торговые точки

`/api/v2/shop/stocks/`  
`/api/v2/shop/stocks/{stockId}/`

### 5.9.1. Получение информации о торговых точках

Для создания запроса на получение данных о списке торговых точек используется метод **GET** без параметров. Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
officeID	id торговой точки
mode	вид торговой точки
officeName	название торговой точки
phone	номер телефона торговой точки
address	адрес торговой точки (улица и номер дома)
country	страна торговой точки
region	регион/область торговой точки
city	город/район торговой точки
district	район/нас. пункт торговой точки
chiefEmail	email ответственного менеджера
show	признак видимости торговой точки (доступности при регистрации)
disabled	признак блокировки торговой точки
default	признак головного офиса

Примечание Для получения информации по конкретной торговой точке необходимо использовать URL вида `/api/v2/shop/stocks/{stockId}/`.

## 5.10. API менеджеры торговых точек

`/api/v2/shop/managers/`

### 5.10.1. Получение информации о менеджерах торговых точек

Для создания запроса на получение данных о менеджерах торговых точек используется метод **GET** без параметров. Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
managerID	id менеджера
managerName	имя менеджера
managerLogin	логин менеджера
officeID	id торговой точки менеджера
phone	номер телефона менеджера
email	email менеджера

Примечание Для получения информации по конкретному менеджеру торговой точки необходимо использовать URL вида `/api/v2/shop/managers/{managerId}/`.

## 5.11. API настройки системы

`/api/v2/shop/settings/`

### 5.11.1. Получение настроек системы

Для создания запроса на получение настроек системы используется метод **GET** без параметров. Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
name	техническое имя настройки
caption	название настройки
value	значение настройки
updated	дата обновления настройки

## 5.12. API страницы сайта

`/api/v2/site/content-pages/`

### 5.12.1. Получение контентных страниц сайта

Для создания запроса на получение контентных страниц для приложения используется метод **GET** без параметров. Результатом запроса будет выборка в виде массива, каждый элемент которого состоит из полей:

Параметр	Описание
contentID	id страницы
name	техническое название страницы
url	адрес страницы
caption	заголовок страницы
text	текст страницы

## 6. Пример работы с REST API на PHP

```
function httpRequest($url, $postData = [], $method = 'GET', $timeout = 10) {

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);

    if (!empty($postData) && ($method != 'GET')) {

        if (strtoupper($method) == 'PUT') {

            curl_setopt ($ch, CURLOPT_CUSTOMREQUEST, 'PUT');
            curl_setopt ($ch, CURLOPT_POSTFIELDS, json_encode($postData));

        } elseif (strtoupper($method) == 'DELETE') {

            curl_setopt ($ch, CURLOPT_CUSTOMREQUEST, 'DELETE');
            curl_setopt ($ch, CURLOPT_POSTFIELDS, json_encode($postData));

        } else {

            curl_setopt($ch, CURLOPT_POST, 1);
            curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($postData));

        }

    }

    curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 3);
    curl_setopt($ch, CURLOPT_TIMEOUT, $timeout);

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

    $result = curl_exec($ch);

    curl_close($ch);

    return json_decode($result);

}

$baseUrl = 'http://_domain.ru/_api/v1';
$token = '_token_';

// получение списка клиентов (GET)
$result = httpRequest($baseUrl . '/customers/?token=' . $token);
if (empty($result->error) && !empty($result->result)) {
    echo '<pre>'; print_r($result->result); echo '</pre>';
}

// добавление клиента (POST)
$postData = [
    'name' => 'Test',
    'surname' => 'Testov',
    'login' => 'test123',
    'phone' => '123',
];
$result = httpRequest($baseUrl . '/customers/?token=' . $token, $postData, 'POST');
if (empty($result->error) && !empty($result->result)) {

    $newCustomerID = $result->result->customerID;

    echo 'клиент создан с customerID = ' . $newCustomerID . '<br />';

    // изменение информации о клиенте (PUT/POST с передачей customerID в data-параметрах или в url сущности)
    $postData = [
        'phone' => '12345'
    ];
    $result = httpRequest($baseUrl . '/customers/' . $newCustomerID . '/?token=' . $token, $postData, 'PUT');
    if (empty($result->error) && !empty($result->result)) {
        echo 'информация о клиенте с customerID = ' . $result->result->customerID . ' изменена<br />';
    }

    // удаление клиента (DELETE - удаляется или конкретный клиент, если в url сущности передан id, или ВСЕ
    сущности подходящие под выборку (аналогичную выборке GET))
    $result = httpRequest($baseUrl . '/customers/' . $newCustomerID . '/?token=' . $token, [], 'DELETE');
    if (empty($result[0]->error) && !empty($result[0]->result)) {
        echo 'клиент с customerID = ' . $result[0]->result->customerID . ' удалён<br />';
    }
}

}
```

## 7. Изменения в документации к модулю REST API

Дата изменения	Раздел, в который внесены изменения	Изменение
02.10.2019	Список параметров для получения информации о позиции	Из таблицы параметров удалена дублирующая строка с параметром debtSum.
02.10.2019	Получение информации о заказах	Из таблицы с параметрами, содержащимися в ответе на запрос, удалена строка с параметром dateTime.
02.10.2019	Создание/редактирование/получение информации о позиции заказа клиента	В таблицу параметров, принимаемых при создании/редактировании заказа, добавлены параметры brandFromSupplier и articleFromSupplier.
02.10.2019	Создание/редактирование заказа	В таблице с параметрами для создания заказа добавлен параметр orderExternalNumber.
15.02.2021	Формат запросов REST API v2	Добавлен раздел
15.02.2021	Получение информации о плательщиках	В описании параметра showMode изменен режим d, на deleted
15.02.2021	Обновление баланса	Добавлены параметры <b>currencyID</b> и <b>currencyISO</b>
15.02.2021	Получение информации о финансовых документах	Исправлено написание параметра <b>sum</b> с <b>Sum</b>
15.02.2021	API позиций заказов клиентов	Исправлен URL
15.02.2021	Загрузка остатков	Добавлены параметры <b>destination</b> и <b>remainsRating</b>
15.02.2021	API справочник деталей API справочника аналогов и ошибочных аналогов	Добавлены разделы
15.02.2021	Сущности, для которых реализован API V2	Добавлен раздел
15.04.2021	Создание/редактирование клиента	В таблицу списка принимаемых параметров добавлен параметр district. Изменено описание параметров region, city.
15.04.2021	Создание редактирование адреса доставки	В таблицу списка принимаемых параметров добавлен параметр district. Изменено описание параметров region, city.
15.04.2021	Получение информации о торговых точках	В таблицу результата запроса добавлен параметр district. Изменено описание параметра city.
15.04.2021	Создание клиента (регистрация)	В таблицу списка параметров добавлен параметр districtID.
15.04.2021	Редактирование данных клиента	В таблицу списка параметров добавлен параметр districtID.
15.04.2021	Получение информации о данных клиента	В таблицу результата запроса добавлены параметры districtID, districtName.