

Интеграционный гид

Axiomus API

Версия: **2.65**
1 декабря 2020

Содержание

Содержание	2
Изменения.....	4
Общее описание системы.....	9
Тестирование запросов, тестовый кабинет.....	10
Статусы заявок	11
Заявка на доставку в регионе Москва и Санкт-Петербург(<i>методы new и update</i>).....	12
Ответ сервера на запрос о создании или изменении заявки	16
Коды ответов	16
Запрос географии пунктов самовывоза (<i>get_boxberry_pickup</i>)	18
Пример ответа сервера на запрос <i>get_boxberry_pickup</i>	18
Заявка на самовывоз (<i>методы new_boxberry_pickup, update_boxberry_pickup</i>).....	20
Заявка на забор товара из интернет-магазина в регионе Москва (<i>методы new_export и update_export</i>).....	22
Заявка на привоз товара на наш склад в регионе Москва (<i>методы new_self_export и update_self_export</i>)	24
Заявка на отправку Почтой России (<i>методы new_post и update_post</i>)	26
Выборочное изменение параметров заявки (<i>метод order_correction</i>)	29
Изменение выборочных параметров заявок курьерской доставки:.....	29
Изменение выборочных параметров заявок Почты России:	30
Контроль дублирования заявок по внутренним номерам	32
Запрос текущего статуса заявки (<i>метод status</i>)	34
Пример ответа сервера.....	34
Запрос статуса для группы заявок (<i>метод status_list</i>)	36
Пример ответа сервера.....	36
Запрос списка заявок за период (<i>метод get_orders_list</i>)	38
Пример ответа сервера.....	38
Запрос ключей заявок (<i>метод get_okey</i>).....	40
Пример ответа сервера.....	40
Запрос стоимости услуг (<i>метод get_price</i>)	41
Упрощенный запрос стоимости услуг доставки (<i>метод rough_estimate</i>).....	42
Пример ответа сервера.....	43
Запрос списка заявок для передачи на склад (<i>метод get_export_orders</i>)	44
Пример ответа сервера.....	44
Пример ответа сервера.....	45
Аннулирование заявки (<i>метод delete</i>).....	46
Пример ответа сервера.....	46
Запрос списка ярлыков заявок (<i>метод get_label</i>)	47
Пример ответа сервера.....	47
Запрос бланков печатных форм заявок (<i>метод get_printform</i>)	49
Пример ответа сервера.....	49
Запрос определения ближайшей даты доставки (<i>метод get_next_delivery</i>)	50
Пример ответа сервера.....	50
Запрос создания нового тикета (<i>метод new_ticket</i>)	52
Пример ответа сервера.....	52

Запрос добавления сообщения в тикет (<i>метод add_ticket_mess</i>)	53
Пример ответа сервера.....	53
Запрос переписки по тикету (<i>метод get_ticket</i>).....	54
Пример ответа сервера.....	54
Запрос списка последних тикетов (<i>метод get_ticketlist</i>).....	55
Пример ответа сервера.....	55
Запрос списка заявок, укомплектованных в возврат отказов (<i>метод</i> <i>get_returns</i>).....	56
Пример ответа сервера.....	56
Запрос текущей версии API (<i>метод get_version</i>)	57
Пример ответа сервера.....	57
Пример отправки запроса с помощью PHP.....	58
Адресный ярлык.....	59
Пример адресного ярлыка.....	59
Для разработчиков	61
Контакты	62

Изменения

Версия	Изменения
2.01	В запросе DPD в тег <order /> добавлен параметр post_type для выбора тарифа на отправку (DPD Consumer или DPD Classic Parcel)
2.02	Оформление заявок в регионы (запросы get_regions, new_region_courier, update_region_courier, new_region_pickup, update_region_pickup)
2.03	В запросе new_export/update_export в тег <order /> добавлен параметр transit для указания транзитного забора товаров
2.04	В запросе new и update new_carry и update_carry в тег <services/> добавлен параметр card для указания оплаты по пластиковой карте
2.05	В запросах new/update/new_export/update_export в тег <order /> добавлен параметр garden_ring – признак доставки в пределах зоны платной парковки
2.06	В запросе status добавлен вывод информации о пакетах (<packs>...</packs>) для заявок забора, привоза и возврата отказов
2.07	Доработан режим самовывоза при формировании заявок DPD Введены интервалы времени выполнения заявок DPD
2.08	Добавлены новые пункты выдачи заказов для заявок Самовывоза new_carry и update_carry
2.09	В запросах new/update/new_carry/update_carry в тег <order /> добавлен параметр sms_sender – наименование отправителя
2.10	В запросах new/update/new_carry/update_carry добавлена возможность задания дифференцированной стоимости доставки в зависимости от стоимости выкупаемого товара тег <delivset>
2.11	В запросах new_region_courier, update_region_courier, new_region_pickup, update_region_pickup добавлена опция указания возможности частичного отказа part_return
2.12	Добавлены: Запрос статуса для группы заявок (метод status_list) Запрос текущей версии API (метод get_version)
2.13	Обновление опций дифференцированной стоимости доставки в запросах new/update/new_carry/update_carry. Добавлена возможность автоматического расчета стоимости доставки. Изменена структура блока <delivset>
2.14	В запросах new_post/update_post в тег <services/> добавлены атрибуты: class1, post_tarif, not_avia, optimize Изменен порядок определения атрибута valuation В запросах new_dpd/update_dpd в тег <services/> добавлен атрибут: waiting Изменен порядок определения атрибута valuation

	Изменен список пунктов выдачи заказов Добавлен запрос аннулирования заявки delete.
2.15	Добавлен запрос стоимости услуг (метод get_price) В запросах new / update / new_carry / update_carry / new_post / update_post / new_dpd / update_dpd / new_region_courier / update_region_courier / new_region_pickup / update_region_pickup в тег <orders/> добавлен атрибуты: email
2.16	Добавлены новые ПВЗ в запросах new_carry / update_carry
2.17	В запросе get_price доработаны режимы получения стоимости услуг для отправок Почтой России и заявок в регионы
2.18	В запросе new_post/update_post доработаны услуги SMS-сопровождения (атрибут sms_inform в теге <services/>) и расширенного страхования (атрибут insurance в теге <services/>)
2.19	Оформление заявок самовывоза VoxBerry (запросы get_boxberry_pickup, new_boxberry_pickup, update_boxberry_pickup)
2.20	В запросах new/update/new_carry/update_carry добавлена возможность задания дифференцированной скидки в зависимости от стоимости выкупаемого товара тег <discountset> Добавлены атрибуты discount_value и discount_unit в тег <orders> В запросах new / update / new_carry / update_carry добавлен признак того, что товар следует забрать у покупателя: атрибут expmode в теге <item>
2.21	В запросах new_carry / update_carry добавлены новые ПВЗ для СПб Внесены уточнения в правила самостоятельного формирования адресных ярлыков
2.22	В запросах , new_boxberry_pickup / update_boxberry_pickup аннулирована опция part_return В запросах new_carry / update_carry добавлены новый ПВЗ Мск Шоссе Энтузиастов
2.23	Добавлен запрос географии самовывоза DPD (get_dpd_pickup) В запросах , new_dpd / update_dpd доработана возможность установки пунктов самовывоза DPD по результатам запроса get_dpd_pickup
2.24	Добавлен запрос географии самовывоза для Москвы и Санкт-Петербурга (get_carry)
2.25	Для заявок самовывоза VoxBerry (запрос get_boxberry_pickup) дополнительно добавлены ПВЗ без возможности наличной оплаты (параметр OnlyPrepaidOrders).
2.26	Добавлен упрощенный запрос для оценки стоимости услуг доставки и самовывоза (метод rough_estimate)
2.27	Добавлен атрибут datestate для отслеживания дат статусов для заявок Почты, DPD и VoxBerry Коды в ответ запроса статуса почтовых заявок (метод status)
2.28	Для тестовой площадки доступны запросы: get_price и rough_estimate Для тестовой площадки в запросы status, status_list добавлена нода:

	<code><auth ukey="XXcd208495d565ef66e7dff9f98764XX" /></code>
2.29	В запросах <code>new_boxberry_pickup / update_boxberry_pickup</code> добавлен адрес электронной почты получателя для пересылки уведомлений
2.30	В запросах <code>new_post / update_post / new_dpd / update_dpd / new_boxberry_pickup / update_boxberry_pickup</code> ограничены допустимые значения для атрибут <code>name</code> и ноды <code>< contacts></code>
2.31	В запросах <code>new_post / update_post</code> добавлен новый тип отправления «Посылка Онлайн» (параметр <code>post_type=3</code>) В ответ сервера на запросы добавлена выдача предупреждений <code>warnings</code> (см. пример в «Ответ сервера на запрос о создании или изменении заявки»)
2.32	В запросах <code>new / update / new_carry / update_carry</code> добавлен признак запрета частичного отказа покупателем/ (атрибут <code>avoid_part_return</code>)
2.33	В запросах <code>new_dpd / update_dpd</code> добавлен признак страны доставки (атрибут <code>country</code>)
2.34	В запросах <code>new / update / new_carry / update_carry / new_post / update_post / new_dpd / update_dpd / new_region_courier / update_region_courier / new_region_pickup / update_region_pickup / new_boxberry_pickup / update_boxberry_pickup</code> в тег <code><orders/></code> добавлен атрибут <code>site</code> для указания сайта Интернет магазина
2.35	В запросах <code>new / update / new_carry / update_carry</code> в тег <code><item/></code> добавлен атрибут <code>failparice</code> для указания стоимости отказа от товара
2.36	Оформление заявок доставки Стриж (запросы <code>new_strizh, update_strizh</code>)
2.37	Добавлен запрос списка заявок для передачи на склад (метод <code>get_export_orders</code>)
2.38	Добавлен запрос списка ярлыков заявок (метод <code>get_label</code>)
2.39	Добавлены методы обработки тикетов (<code>new_ticket, add_ticket_mess, get_ticket</code>)
2.40	Добавлен метод загрузки списка последних тикетов (<code>get_ticketlist</code>)
2.42	Добавлен метод определения ближайшей даты доставки по типам заявок (<code>get_next_delivery</code>)
2.43	Доработана выдача и контроль признаков допустимости Наложенного платежа и Ожидания для отдельных ПВЗ DPD (<code>get_dpd_pickup</code>)
2.44	Добавлен атрибут города отправления <code>city_from</code> для методов: <code>new / update / new_carry / update_carry / new_export / update_export / new_self_export / update_self_export / new_post / update_post / new_dpd / update_dpd / new_region_courier / update_region_courier / new_region_pickup / update_region_pickup / new_boxberry_pickup / update_boxberry_pickup / new_strizh / update_strizh</code> . Добавлен атрибут города назначения <code>city</code> для методов: <code>new_export / update_export / new_self_export / update_self_export / new_strizh / update_strizh</code>
2.45	Добавлен запрос бланков печатных форм заявок (<code>get_printform</code>)
2.46	Рализация механизма обратого вызова при изменении статуса заявок

	(см. метод status)
2.47	Добавлен метод запроса ключей заявок по их номерам (get_okey)
2.48	Добавлен признак запрета вскрытия упаковки при доставке заказа покупателю в запросах new / update
2.49	Завершен прием заявок самовывоза (методы get_carry / new_carry / update_carry) Для создания заявок самовывоза рекомендуем использовать методы new_boxberry_pickup/update_boxberry_pickup.
2.50	Изменения в разделе «Адресный ярлык». Добавлено описание аббревиатуры ВВМск для заявок самовывоза в московском регионе.
2.51	Изменения название тарифа у почты России: «Бандероль 1 класс» на «Посылка 1 класс».
2.52	Добавлен атрибут from_city для указания расстояния от МКАД до ПВЗ для метода get_boxberry_pickup.
2.53	Прекращено использование методов get_regions / new_region_courier / update_region_courier / new_region_pickup / update_region_pickup / new_strizh / update_strizh.
2.54	Вводится новый статус «Поступил в ПВЗ» (код=106) вместо старого аналогичного статуса (код=70) См. раздел «Запрос текущего статуса заявки (метод status)» Изменение вступает в силу 10.11.2017г.
2.55	Добавлена возможность передачи запросов по протоколу HTTPS (см. Пример отправки запроса с помощью РНР) Отменен механизм верификации контрольных сумм при формировании и изменении заявок. Параметр checksum в методах new / update / new_export / update_export и пр. более не является обязательным.
2.56	Добавлена новые типы отправления (Посылка Онлайн и Курьер Онлайн) в запросах new_post / update_post
2.57	В результаты запроса географии самовывоза DPD (get_dpd_pickup) добавлены параметры ПВЗ: maxcash, maxWeight, maxLength, maxWidth, maxHeight, dimensionSum.
2.58	Добавлен новый метод для выборки списков заявок (get_orders_list)
2.59	Добавлены атрибуты указания типа упаковки (wrap_type и exact_wrap) для отправок Почтой России в запросах new_post / update_post
2.60	Добавлена функция контроля дублирования заявок по внутренним номерам (см.раздел “Контроль дублирования заявок по внутренним номерам”). Введена возможность оплаты по пластиковой карте для заявок курьерской доставки в Санкт-Петербурге.
2.61	Добавлен новый метод внесения изменений в выборочные параметры заявок (order_correction)
2.62	Добавлен новый метод для выборки списка заявок, укомплектованных в возврат отказов (get_returns)
2.63	Анулирован атрибут datestate для отслеживания дат статусов для заявок Почты, DPD и BoxBerry
2.64	Анулированы методы new_dpd, update_dpd и get_dpd_pickup
2.65	Доработан режим выдачи списка заявок для указанного

забора/привоза на склад (метод `get_export_orders`)

Общее описание системы

Размещение заявок, отслеживание их статуса, получение иной информации происходит путем отправки **xml-документов** с помощью **POST-запроса**.

В ответе нашего сервера будет содержаться необходимая информация в виде xml-документа.

Логика работы системы подразумевает один запрос для каждого действия (например, для оформления заявки нужен только один запрос), т.е. нет необходимости делать несколько запросов для получения одного результата (нет цепочки).

В случае ошибки возвращается статус ошибки и её описание.

Все xml-документы должны быть в кодировке **UTF-8**.

Xml-документ должен быть передан в параметре **data** (также см. пример в разделе «Пример отправки запроса с помощью PHP»).

Все рабочие запросы отправляются на url:

https://axiomus.ru/hydra/api_xml.php

Для отправки запроса на рабочий url необходимо знать свои **uid** и **ukey** (ваш личный ukey и uid вышлет наш специалист после вашего запроса системе тикетов ЛК или на info@axiomus.ru).

Рекомендуем также воспользоваться нашим API для детального расчета тарифа - <https://axiomus.ru/wiki/index.php>

Тестирование запросов, тестовый кабинет

Вы можете ознакомиться с демо-версией Личного кабинета Аксиомус по адресу <http://axiomus.ru/testnew/private.htm>

Для входа используйте логин ***axiomus*** и пароль ***demo***

Там же вы можете протестировать и отладить систему Axiomus API. Все тестовые заявки, оформленные через API доступны в нем.

Url для тестовых запросов: https://axiomus.ru/hydra/api_xml.php

Для создания запросов используйте:

ukey: *XXcd208495d565ef66e7dff9f98764XX*

uid: *92*

Статусы заявок

Ниже указаны коды используемых статусов заявок и их описание:

- 10: Отклонена (Заказ отклонен, причина указана в поле Примечания. Измените или отмените заказ)
- 0: В обработке (Заявка оформлена в нашей системе, ожидает дальнейшего исполнения)
- 1: Укомплектован (Заказ укомплектован на складе)
- 2: Товар на складе (Заказ принят на складе)
- 3: Нет товара (Товар по данной заявке отсутствует)
- 4: Исполнение (Заказ выполняется нашим сотрудником)
- 5: В процессе (Сложности с заказом, наш сотрудник свяжется с Вами)
- 6: Перенос доставки (Заказ перенесен на другую дату исполнения)
- 20: Комплектация (Заказ комплектуется)
- 30: Сортировка (Заказ отсортирован на складе)
- 80: *Исполнен (Заказ исполнен нашим сотрудником, доставляется на склад)
- 81: Предотмена (Заявлена предварительная отмена заявки)
- 90: Отмена (Заявка отменена)
- 100: Выполнен (Заявка исполнена)
- 105: Отправлен (Заказ отправлен в внешнюю службу доставки)
- 107: Вручен (Заказ вручен получателю)
- 110: Частичный отказ (Покупатель отказался от части товаров в заказе. Товар поступил на склад)
- 120: Полный отказ (Покупатель/Заказчик отказался от заказа. Товар поступил на склад)
- 106: Поступил в ПВЗ (Заказ поступил для исполнения на ПВЗ)**
Статус вводится вместо используемого ранее статуса
«Поступил в ПВЗ» (код=70)

Заявка на доставку в регионе Москва и Санкт-Петербург(методы *new* и *update*)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<order inner_id="16454" name="Клотильда" address="Москва,
Живописная, д4 корп1, кв 16" from_mkad="0" d_date="2009-06-25"
b_time="12:00" e_time="18:00" incl_deliv_sum="200.15" places="1"
city="0" sms="79031234567" discount_value="auto" discount_unit="0"
avoid_part_return="0" site="www.mysite.ru">
  <contacts>тел. (499) 222-33-22</contacts>
  <description>проход в арку и налево, 3 подъезд</description>
  <services cash="yes" cheque="no" />
  <items>
    <item name="Крем для лица" weight="0.400" quantity="1"
price="155.00" />
    <item name="Крем для тела" weight="0.340" quantity="3"
price="235.00" failprice="50" />
    <item name="Крем для рук" weight="1.000" quantity="2"
price="340.55" />
  </items>
  <delivset return_price="300.00" above_price="50.00">
    <below below_sum="1000.00" price="250.00" />
    <below below_sum="5000.00" price="150.00" />
    <below below_sum="7000.00" price="100.00" />
  </delivset>
  <discountset above_discount="15.00">
    <below below_sum="1000.00" discount="0.00" />
    <below below_sum="5000.00" discount="5.00" />
    <below below_sum="7000.00" discount="10.00" />
  </discountset></order>
</singleorder>
```

<mode>

new – новая заявка

update – изменение заявки

<auth>

ukey - ключ

<order>

* **okey** – уникальный ключ заказа, обязателен только для запроса **update**

* **inner_id** - внутренний номер заказа в Интернет -магазине (varchar(255))

name – ФИО получателя заказа (varchar(255))

address – почтовый адрес доставки с указанием города (если возможно) и с разделителем ',' между городом, улицей, домом, квартирой (varchar(255))

* **from_mkad** – расстояние от МКАД / КАД (для заявок в СПб) (float(9.2))

d_date - в формате ISO (date Y-m-d), дата не ранее сегодня

b_time, e_time - время доставки в часах с .. по .., (date H:00).

Разница минимум 3 часа, **b_time** с 10 до 19 часов, **e_time** с 15 до 22 часов.

* **incl_deliv_sum** - включить в сумму заказа стоимость доставки (в т.ч. отдельной строкой в бланке) (float(9.2))
Допускаются следующие варианты задания значения атрибута **incl_deliv_sum**:
- incl_deliv_sum отсутствует или incl_deliv_sum="0". В этом случае заявка будет считаться без оплаты за доставку клиентом. В этом режиме также может определяться цена доставки для покупателя для случая Полного отказа (см. ниже описание атрибута **return_price** в блоке <delivset>)
- incl_deliv_sum="цена" (напр., incl_deliv_sum="250"). В этом случае стоимость оплаты доставки клиентом будет считаться равной указанной цене. В этом режиме также может определяться цена доставки для покупателя для случая Полного отказа (см. ниже описание атрибута **return_price** в блоке <delivset>)
- incl_deliv_sum="auto". В этом случае стоимость оплаты доставки клиентом будет рассчитана автоматически на основе данных из блока <delivset> (см. описание ниже) или на основе настроек профиля доставок из ЛК.

places - кол-во транспортных мест (не товаров!) в данном заказе (int)
city - код города доставки (int)
для Москвы **city**"=0
для Санкт-Петербурга **"city"**=1

* **sms** - телефон для рассылки sms-уведомлений о статусе заказа (varchar(11)), пример корректного номера: 79031234567. Номер будет считаться допустимым, если он начинается с 79, исключая номера, начинающиеся с 7940

* **sms_sender** - наименование отправителя sms-уведомлений (varchar(11))
буквенно-цифровое значение минимум 3, максимум 11 символов.
Поддерживаются латинские буквы a-zA-Z, цифры 0-9, дефис и точка.

* **garden_ring** - признак доставки в пределах зоны платной парковки ('yes'/'no')

* **email** - адрес электронной почты получателя для пересылки уведомлений (varchar(250))

* **discount_value** - позволяет включить в заказ скидку для клиента (float(9.2))
Допускаются следующие варианты задания значения атрибута **discount_value**:
- discount_value отсутствует или discount_value="0". В этом случае заявка будет считаться без скидки для клиента
- discount_value="цена" (напр., discount_value="150"). В этом случае скидка будет считаться равной указанной цене. Скидка задается с учетом единицы измерения, см. ниже атрибут discount_unit.
- discount_value="auto". В этом случае размер скидки для клиента будет рассчитываться автоматически на основе данных из блока <discountset> (см. описание ниже) или на основе настроек профиля доставок из ЛК.

* **discount_unit** - единица измерения скидки. Может принимать одно из двух значений:
- discount_unit="0" (используется по умолчанию), означает, что скидка задана в процентах от стоимости товара по заявке. В этом режиме скидка не должна превышать 100.
- discount_unit="1", означает, что скидка задана в рублях. В этом режиме скидка не должна превышать стоимость товара.

* **avoid_part_return** - признак запрета частичного отказа покупателем (1/0)

avoid_part_return="1" – частичный отказ покупателем по заявке запрещен;
avoid_part_return="0" (по умолчанию) – частичный отказ покупателем допускается (используется по умолчанию)
* **without_opening** – признак запрета вскрытия упаковки при доставке заказа покупателю (1/0)
without_opening="1" – вскрытие упаковки при доставке заказа покупателю запрещено; without_opening="0" (по умолчанию) – разрешено вскрытие упаковки при доставке заказа покупателю.
* **city_from** - код города, из которого производится отправка (int)
для Москвы **city_from="0"**
для Санкт-Петербурга **city_from="1"**
По умолчанию используется город, в котором зарегистрирован ваш аккаунт в нашей системе.

ПРИМЕЧАНИЕ:

Стоимость заявки для клиента=Стоимость товаров-Скидка+Стоимость доставки.

При автоматическом расчете стоимости доставки она рассчитывается от разности: Стоимость товаров-Скидка

* **site** – сайт интернет магазина (varchar(255)). Если в запросе атрибут site пропущен и при этом в ЛК в карточке интернет магазина в разделе *Для покупателей* задано значение поля *Сайт*, то это значение будет подставлено в заявку автоматически

<**contacts**> - контактная информация (varchar(255))

* <**description**> - пояснения, комментарии к заказу (varchar(1024))

<**services**> - дополнительные услуги:

* **cash**="yes" - наложенный платеж ('yes'/'no')

* **cheque**="yes" - чек по агентскому договору ('yes'/'no')

* **card**="yes" – оплата по пластиковой карте ('yes'/'no').

* **big** – признак большегруза ('yes'/'no')

Одновременно можно использовать либо **cash**, либо **cheque**, либо **card**, либо **cheque u card**, в соответствии с типом заключенного договора

<**item**> - товар

name - название товара (varchar(255))

weight - вес, положительное число (float(9,3))

quantity - кол-во, минимум 1 (int)

price - цена, м.б. равна 0, если не требуется прием денег от клиентов (float(9,2)). Если указана отрицательная стоимость позиции, то автоматически будет установлен признак exprmode=1.

* **exprmode** – признак того, что товар нужно забрать у покупателя (exprmode=1 означает, что признак включен). Если exprmode=1, то автоматически цена позиции (price) будет считаться отрицательной.

* **failprice** – стоимости отказа от товара клиентом (float(9,2)).

Примечание: Для тарифа «Легкий» число товарных позиций в заявке не должно превышать 7.

Блок дифференцированной стоимости доставки в зависимости от стоимости выкупаемого товара

* **<delivset>** - Блок дифференцированной стоимости доставки

* **return_price** – Цена доставки для покупателя для случая Полного отказа (float(9,3)), см п. 1 на рисунке ниже.

above_price – Цена доставки при превышении верхней границы стоимости выкупаемого товара (float(9,2)), см. п. 5 на рисунке ниже.

Верхняя граница стоимости выкупаемого товара вычисляется автоматически и равна максимальной стоимости выкупаемого товара **below_sum** (см. описание ниже)

* **<below>** - промежуточные границы стоимости выкупаемого товара (должно быть не более трех), см. п.2-4 на рисунке ниже.

below_sum – граница стоимости выкупаемого товара (float(9,3))

price – Цена доставки при не превышении границы стоимости выкупаемого товара (float(9,3))

Рис: Блок дифференцированной стоимости доставки в ЛК клиента

Сумма заказа, руб.:	< 1000	< 5000	< 7000	≥ 7000	В случае ПОЛНОГО ОТКАЗА
Цена доставки, руб.:	2 250	3 150	4 100	5 50	1 300

Блок дифференцированной скидки в зависимости от стоимости выкупаемого товара

* **<discountset>** - Блок дифференцированной скидки

above_discount– Скидка при превышении верхней границы стоимости выкупаемого товара (float(9,2)), см. п. 4 на рисунке ниже. .Скидка задается с учетом единицы измерения, см. атрибут discount_unit в разделе <orders>

Верхняя граница стоимости выкупаемого товара вычисляется автоматически и равна максимальной стоимости выкупаемого товара **below_sum** (см. описание ниже)

* **<below>** - промежуточные границы стоимости выкупаемого товара (должно быть не более трех), см. п.1-3 на рисунке ниже.

below_sum – граница стоимости выкупаемого товара (float(9,3))

discount – Скидка при не превышении границы стоимости выкупаемого товара (float(9,3)). Скидка задается с учетом единицы измерения, см. атрибут discount_unit в разделе <orders>

Рис: Блок дифференцированной скидки в ЛК клиента

Скидка для покупателя:	<input type="radio"/> отсутствует	<input checked="" type="radio"/> изменяемая	<input type="radio"/> фиксированная	Формат: <input checked="" type="radio"/> % <input type="radio"/> руб.	
Сумма заказа, руб.:	< 1000	< 5000	< 7000	≥ 7000	В случае ПОЛНОГО ОТКАЗА
Скидка, %:	1 0	2 5	3 10	4 15	0

* - необязательные атрибуты и элементы

Ответ сервера на запрос о создании или изменении заявки

В ответ на полученный запрос на создание (**new_**) или изменение (**update_**) заявки наш сервер возвращает ответ вида:

```
<?xml version='1.0' standalone='yes'?>
<response>
<request>new</request>
<auth objectid="1234567">642d208495dtgu8906e7dff9f98764da</auth>
<status price="154.44" code="0">Запрос выполнен успешно</status>
<warnings>
    <warning>Предупреждение 1</warning>
    <warning>Предупреждение 2</warning>
</warnings>
</response>
```

<request> - содержит тип обработанного запроса ('new_', 'update_')

<auth> - в случае успешного выполнения запроса содержит уникальный код заказа (okey), используется для **update** и получения доп.инфо по заказу (напр., запрос **status**)

objectid - возвращает id обработанной заявки (или id новой заявки для учета в вашей системе)

<status> - содержит описание кода ответа (text)

price - стоимость наших услуг (если есть)

code - статус-код операции

<warnings> - Блок предупреждений. Предупреждения не блокируют выполнение запроса, но сообщают о значимых нарушениях, которые могут повлиять на последующую обработку информации по заявке и др.

<warning> - содержит текст предупреждения (text)

Коды ответов

- 0 - запрос обработан и выполнен успешно
- 1 - ошибка идентификации ukey
- 2 - ошибка контрольной суммы
- 3 - ошибка в ФИО, адресе, расстоянии от МКАД
- 4 - ошибка в дате и времени доставки
- 5 - ошибка в контактах
- 6 - ошибка в доп.услугах
- 7 - ошибка в товарах
- 8 - неверный формат xml
- 9 - не задан auth
- 10 - неверный auth
- 11 - неверный fid
- 12 - заявок по детализации не найдено
- 13 - не соответствие профилю клиента

- 14 - ошибка получателя
- 15 - ошибка sms-номера
- 16 - ошибка номера телефона
- 17 - ошибка оформления договора
- 18 - некорректный индекс
- 19 - отсутствует Агентский договор
- 20 - заявка не найдена
- 21 - превышена квота заявок
- 22 - server too busy
- 23 - недопустимый запрос
- 24 - не допускается аннулирование заявки
- 25 - недопустимый ПВЗ
- 26 - ошибка процесса выполнения запроса
- 27 - ошибка параметров дифференцированной стоимости доставки

Примечание:

Ошибка "Не соответствие профилю клиента" возникает в случае, когда Ваш аккаунт в нашей системе числится с наличной оплатой, а Вы пытаетесь загрузить в систему заявку с указанием cheque="yes" (флажок "Чек" в форме заявки) или наоборот, когда Ваш аккаунт в нашей системе числится как безналичный, а Вы пытаетесь загрузить в систему заявку с указанием cash="yes" (флажок "Наложенный платеж" в форме заявки)

Запрос географии пунктов самовывоза (*get_boxberry_pickup*)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_boxberry_pickup</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
</singleorder>
```

<auth>

ukey - ключ

Пример ответа сервера на запрос *get_boxberry_pickup*

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
<request>boxberry_pickup</request>
<pickup_list>
.....
<office office_code="222" office_name="Альметьевск Маяковского_1603"
office_address="423440, Альметьевск г, Маяковского ул, д.47, оф. 2"
city_code="92" city_name="Альметьевск" GPS="54.9053737,52.3108395"
Area="Татарстан Респ" WorkSchedule="пн-пт: 10.00-19.00, сб: 09.00-
15.00" OnlyPrepaidOrders="0" Phone="8-800-700-54-30" TariffZone="5"
DeliveryPeriod ="5" Acquiring="1" TripDescription="Проезд: автобус №№
- 5,8, троллейбус №№ - 1,2. Остановка: &quot;Нефтяной
институт&quot;;.От остановки пройти 100 м и повернуть на ул.
Маяковского в сторону &quot;Таттелекома&quot;;, спуститься вниз по ул.
Маяковского, вдоль забора Нефтяного института, первый дом - отделение
Боксберри, дверь с левого торца здания." cash="1" card="1" />
<office office_code="23431" office_name="Анапа Крымская_2343_С"
office_address="353440, Анапа г, Крымская ул, д.272" city_code="189"
city_name="Анапа" GPS="44.901101,37.333615" Area="Краснодарский край"
WorkSchedule="пн-вс: 10.00-19.00" OnlyPrepaidOrders="0" Phone="8-800-
700-54-30" TariffZone="5" DeliveryPeriod ="4" Acquiring="0"
TripDescription="Проезд: автобус №№ -
2,9,17,25,100,111,121,126,129,130,135. Остановка: &quot;5-я
школа&quot;;. Отделение находится около остановки. Вход в отделение со
стороны дороги, с ул. Крымской, рядом с фигурами орлов. От арки
первая дверь." cash="1" card="0" />
.....
</pickup_list>
</response>
```

<pickup_list>... </pickup_list> - - список пунктов выдачи заказов (ПВЗ)

office_code – код ПВЗ

office_name – наименование ПВЗ

office_address – адрес ПВЗ

city_code – код города ПВЗ

city_name – наименование города ПВЗ

from_city – указывает расстояние от МКАД до ПВЗ. Атрибут имеет место только для ПВЗ, у которых city_code=68 (Москва). Если значение атрибута from_city равно нулю, то это означает, что для соответствующего ПВЗ действует специальный тариф доставки.

GPS – геокоординаты

Area – регион ПВЗ

WorkSchedule – режим работы ПВЗ

OnlyPrepaidOrders – запрет на наличный расчет [0/1]

Phone - контактный телефон

TariffZone - тарифная зона

DeliveryPeriod – период доставки

Acquiring – то же, что card

TripDescription – описание проезда

cash - признак допустимости оплаты наложенным платежом (1=допустима; 0=не допустима)

card - признак допустимости оплаты банковской картой (1=допустима; 0=не допустима)

Заявка на самовывоз (методы *new_boxberry_pickup*, *update_boxberry_pickup*)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new_boxberry_pickup</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<order inner_id="самовывоз 77" name="Петр Петров" d_date="2014-10-16"
incl_deliv_sum="200.00" site="www.mysite.ru">
  <address office_code="183" />
  <contacts>79031234567</contacts>
  <services cod="yes" checkup="yes" part_return="no" />
  <items>
    <item name="Крем для лица" weight="0.400" quantity="1"
price="100.00" bundling="1" />
    <item name="Крем для тела" weight="0.340" quantity="3"
price="200.00" bundling="1" />
    <item name="Крем для рук" weight="1.000" quantity="2"
price="300.00" />
  </items>
</order>
</singleorder>
```

<mode>

new_boxberry_pickup – новая заявка

update_boxberry_pickup – изменение заявки

<auth>

ukey - ключ

<order>

* **okey** – уникальный ключ заказа, обязателен только для запроса

update_boxberry_pickup

* **inner_id** - внутренний номер заказа в Интернет -магазине (varchar(255))

name – обязательно Фамилия и Имя получателя заказа (varchar(255))

Допускается только кириллица

d_date - в формате ISO (date Y-m-d), дата не ранее завтра для нового заказа

* **incl_deliv_sum** - включить в сумму заказа стоимость доставки (в т.ч.

отдельной строкой в бланке) (float(9.2))

* **places** - кол-во транспортных мест (не товаров!) в данном заказе (int). По умолчанию значение считается равным единице, т.е. заказ состоит из одного места.

* **email** – адрес электронной почты получателя для пересылки уведомлений (varchar(250))

* **site** – сайт интернет магазина (varchar(255)). Если в запросе атрибут site пропущен и при этом в ЛК в карточке интернет магазина в разделе *Для покупателей* задано значение поля *Сайт*, то это значение будет подставлено в заявку автоматически

* **city_from** - код города, из которого производится отправка (int)

для Москвы **city_from**="0"

для Санкт-Петербурга **city_from**="1"

По умолчанию используется город, в котором зарегистрирован ваш аккаунт в нашей системе.

<address>

office_code – код ПВЗ BoxBerry (см. office_code в результате выдачи запроса get_boxberry_pickup).

<**contacts**> - контактный телефон а (varchar(11)), пример корректного номера: 79031234567. Номер будет считаться допустимым, если он содержит 11 знаков, начинается с 79 или 77, исключая номера, начинающиеся с 7940

<**services**> - дополнительные услуги:

* **cod** - наложенный платеж ('yes'/'no')

* **checkup** – признак допустимости проведения проверки заказа при вручении клиенту ('yes'/'no').

* **part_return** – признак допустимости частичного отказа клиентом ('yes'/'no'). При установке опции checkup="no" для опции part_return автоматически будет установлено значение "no". Иными словами, при запрете проведения проверки заказа при вручении клиенту частичный отказ запрещен.

<**item**> - товар

name - название товара (varchar(255))

weight - вес, положительное число (float(9,3))

quantity - кол-во, минимум 1 (int)

price - цена, м.б. равна 0 (float(9,2)). Если деньги с клиента принимать не следует установите атрибут cod="no" (см. выше)

* - необязательные атрибуты и элементы

Заявка на забор товара из интернет-магазина в регионе Москва (методы *new_export* и *update_export*)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
  <mode>new_export</mode>
  <auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
  <order name="Иванов Иван" address="Москва, Васильковская, д4 корп1, оф
16" from_mkad="0" d_date="2010-11-25" b_time="12:00" e_time="16:00"
export_quantity="3" transit="no" >
    <contacts>Иван Иванов, тел. (499) 222-33-22</contacts>
    <description>заказать пропуск за 1 час до прибытия</description>
    <services warrant="no" />
    <items>
      <item name="Крем для лица" weight="0.400" quantity="1"
price="155.00" oid="12345" />
      <item name="Крем для тела" weight="0.340" quantity="3"
price="235.00" oid="12345" />
      <item name="Крем для рук" weight="1.000" quantity="2"
price="340.55" oid="12345" />
    </items>
  </order>
</singleorder>
```

<mode>

new_export – новая заявка

update_export – изменение заявки

<auth>

ukey - ключ

<order>

* **okey** – уникальный ключ заказа, обязателен только для запроса

update_export

name – ФИО ответственного за отгрузку заказов со стороны Интернет-магазина

address – почтовый адрес доставки (varchar(255))

* **from_mkad** - необязательный атрибут, (float(9.2))

d_date - в формате ISO (date Y-m-d), дата не ранее завтра для нового заказа

b_time, e_time - время доставки в часах с .. по .., (date H:00).

Разница минимум 3 часа, **b_time** с 10 до 19 часов, **e_time** с 15 до 22 часов.

export_quantity - кол-во заказов, планируемых к передаче

* **garden_ring**– признак забора в пределах зоны платной парковки ('yes'/'no')

* **city** - код города забора (int)

для Москвы **city='0'**

для Санкт-Петербурга **city="1"**

По умолчанию используется город, в котором зарегистрирован ваш аккаунт в нашей системе.

<contacts> - контактная информация (varchar(255))

* **<description>** - пояснения, комментарии к заказу (varchar(1024))

<services> - дополнительные услуги:

* **warrant** – требуется ли доверенность от имени ООО “Нова” ('yes'/'no')

* **transit** – флаг указания транзитного забора ('yes'/'no')

<item> - позиции (товары)

name – наименование позиции (varchar(255))

weight - вес, положительное число (float(9.3))

quantity - кол-во, минимум 1 (int)

price – цена. Указывайте равной нулю, если по забору не требуется оплата товаров (float(9,2)). Возможность оплаты по забору оговаривается отдельно с нашими менеджерами.

***oid** - номер ранее оформленной в нашей системе заявки **order_id** (int)

Примечание.

Если в заборе товаров в качестве позиций вы указываете номера заявок в нашей системе (параметр oid), то в этом случае остальные параметры (name, weight, quatity, price) могут быть опущены. Список позиций в этом случае может выглядеть следующим образом:

<items>

<item oid="1111111" />

<item oid="1111112" />

<item oid="1111113" />

</items>

* - необязательные атрибуты и элементы

Заявка на привоз товара на наш склад в регионе Москва (методы *new_self_export* и *update_self_export*)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new_self_export</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<order name="Иванов Иван" car="A444AA199" d_date="2010-11-25"
b_time="12:00" e_time="16:00" places="1" quantity="2">
  <items>
    <item name="Крем для лица" weight="0.400" quantity="1"
price="155.00" oid="12344" />
    <item name="Крем для тела" weight="0.340" quantity="3"
price="235.00" oid="12345" />
    <item name="Крем для рук" weight="1.000" quantity="2"
price="340.55" oid="12345" />
  </items>
</order>
</singleorder>
```

<mode>

new_self_export – новая заявка

update_self_export – изменение заявки

<auth>

ukey - ключ

<order>

* ***okey*** – уникальный ключ заказа, обязателен только для запроса

update_self_export

name – ФИО ответственного за отгрузку заказов со стороны Интернет-магазина

* ***car*** – номер автомобиля в формате A000AA (varchar(16)), используется для заказа пропуска

d_date - в формате ISO (date Y-m-d), дата не ранее завтра для нового заказа

b_time, e_time - время привоза в часах с .. по .., (date H:00).

Разница минимум 1 час, ***b_time*** с 12 до 19 часов, ***e_time*** с 15 до 22 часов.

quantity - кол-во заказов, планируемых к отгрузке

places - кол-во транспортных мест (не товаров, не заказов!), планируемых к отгрузке

* ***transit*** – флаг указания транзитного забора ('yes'/'no')

* ***city*** - код города привоза (int)

для Москвы ***city***="0"

для Санкт-Петербурга ***city***="1"

По умолчанию используется город, в котором зарегистрирован ваш аккаунт в нашей системе.

<contacts> - контактная информация (varchar(255))

* **<description>** - пояснения, комментарии к заказу (varchar(1024))

<item> - позиции (товары)

name – наименование позиции (varchar(255))

weight - вес, положительное число (float(9.3))

quantity - кол-во, минимум 1 (int)

price – цена. Указывайте равной нулю, если по привозу не требуется оплата товаров (float(9,2)). Возможность оплаты по привозу оговаривается отдельно с нашими менеджерами.

***oid** - номер ранее оформленной в нашей системе заявки **order_id** (int)

Примечание.

Если в привозе товаров в качестве позиций вы указываете номера заявок в нашей системе (параметр oid), то в этом случае остальные параметры (name, weight, quatity, price) могут быть опущены. Список позиций в этом случае может выглядеть следующим образом:

<items>

<item oid="1111111" />

<item oid="1111112" />

<item oid="1111113" />

</items>

*** - необязательные атрибуты и элементы**

Заявка на отправку Почтой России (методы *new_post* и *update_post*)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new_post</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<order inner_id="почта 333" name="Петр Петров" b_date="2011-03-10"
incl_deliv_sum="200.15" post_type="1" site="www.mysite.ru">
  <address index="127322" region="Камчатский край"
area="Петропавловск Камчатский" p_address="ул. Солнечная д.70, кв.
30" />
  <contacts>79031234567</contacts>
  <services valuation="yes" fragile="no" cod="no" />
  <items>
    <item name="Крем для лица" weight="0.400" quantity="1"
price="155.00" />
    <item name="Крем для тела" weight="0.340" quantity="3"
price="235.00" />
    <item name="Крем для рук" weight="1.000" quantity="2"
price="340.55" />
  </items>
</order>
</singleorder>
```

<mode>

new_post – новая заявка

update_post – изменение заявки

<auth>

ukey - ключ

<order>

* **okey** – уникальный ключ заказа, обязателен только для запроса

update_post

* **inner_id** - внутренний номер заказа в Интернет -магазине (varchar(255))

address - с указанием города (если возможно) и с разделителем ',' между городом, улицей, домом, квартирой (varchar(255))

name – обязательно Фамилия и Имя получателя заказа (varchar(255))

Допускается только кириллица

b_date – предполагаемая дата отправки заказа в формате ISO (date Y-m-d), дата больше текущей

* **incl_deliv_sum** - включить в сумму заказа стоимость доставки (в т.ч. отдельной строкой в бланке) (float(9.2))

post_type – предпочтительный тип отправления (1 – Псылка, 2 – Псылка 1 класс, 3 – Псылка Онлайн, 4 – Курьер Онлайн)

* **email** – адрес электронной почты получателя для пересылки уведомлений (varchar(250))

* **sms** - телефон для sms-сопровождения заказа (varchar(11)), см. также атрибут **sms_inform** ниже. Пример корректного номера: 79031234567.

Номер будет считаться допустимым, если он начинается с 79 или 77, исключая номера, начинающиеся с 7940

* **site** – сайт интернет магазина (varchar(255)). Если в запросе атрибут site пропущен и при этом в ЛК в карточке интернет магазина в разделе *Для покупателей* задано значение поля *Сайт*, то это значение будет подставлено в заявку автоматически

* **city_from** - код города, из которого производится отправка (int)
для Москвы **city_from="0"**

для Санкт-Петербурга **city_from="1"**

По умолчанию используется город, в котором зарегистрирован ваш аккаунт в нашей системе.

* **wrap_type** – тип упаковки отправления (int). Допускаются следующие значения атрибута: 0 - без доупаковки, 1 - без доупаковки (негабарит), 2 - упаковка, 3 - упаковка (негабарит).

По умолчанию используется значение wrap_type="2".

* **exact_wrap** – признак строгого соответствия упаковки, указанной в атрибуте wrap_type (int). Допускаются следующие значения атрибута: 1 - строгое соответствие упаковки; 0 – допускается изменение указанного в атрибуте wrap_type типа упаковки.

<address>

index - индекс получателя (varchar (6))

region - регион получателя (varchar (255))

area - город / район получателя (varchar (255))

p_address - адрес получателя (varchar (255))

<contacts> - контактный телефон а (varchar(11)), пример корректного номера: 79031234567. Номер будет считаться допустимым, если он содержит 11 знаков, начинается с 7 или 8

<services> - дополнительные услуги:

* **valuation** - объявленная стоимость ('yes'/'no'). При установке cod="yes" производится автоматическая установка valuation="yes"

* **cod** - наложенный платеж ('yes'/'no')

* **post_tarif** - Наложный платеж + почтовый тариф ('yes'/'no'). Установка допускается только при условии установки cod="yes"

* **fragile** – отметка "Осторожно" ('yes'/'no'). Используйте только для типа отправления «Посылка» (post_type="1")

* **optimize** – отметка включения Оптимизатора тарифа ('yes'/'no').

* **class1** – отметка Строгого типа отправления ('yes'/'no'). Отметка имеет приоритет над отметкой optimize.

* **not_avia** – отметка Авиазапрет ('yes'/'no'). В настоящее время не используется (оставлен для совместимости)

* **big** – признак большегруза ('yes'/'no'). В настоящее время не используется (оставлен для совместимости)

* **sms_inform** – отметка SMS-сопровождения заявки ('yes'/'no'). Для включения услуги необходимо определить атрибут sms в теге <orders>

* **insurance** – отметка расширенного страхования ('yes'/'no'). Установка допускается только при условии установки cod="yes" или valuation="yes"

<item> - товар

name - название товара (varchar(255))

weight - вес, положительное число (float(9,3))

quantity - кол-во, минимум 1 (int)

price - цена, м.б. равна 0, если не требуется прием денег от клиентов (float(9,2))

***** - необязательные атрибуты и элементы

Выборочное изменение параметров заявки (*метод order_correction*)

Внесение изменений в ранее созданные заявки осуществляется методами update, update_post, update_export, update_self_export, и т.п. для заявок курьерской доставки, самовывоза, заборов и привозов товара и др., соответственно (см. выше настоящее руководство).

Все эти методы предполагают, передачу в запросе полной информации о заявке, включая ее товарные позиции, порядок оплаты, скидки и т.п. Это не всегда удобно, если требуется изменить один или два параметра заявки, например, дату или интервал ее исполнения, уточнить адрес, контактную информацию и др. Настоящий метод является вспомогательным и позволяет реализовать изменение отдельных параметров заявки без указания полной информации по заявке. Число параметров, которые могут быть изменены настоящим методом ограничено.

Если требуемые вам параметры не указаны ниже, то следует воспользоваться методами полного изменения update, update_boxberry_pickup и т.п.

В настоящее время метод реализован для заявок курьерской доставки и Поты России.

Изменение выборочных параметров заявок курьерской доставки:

Пример запроса

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>order_correction</mode>
<auth ukey="XXcd208495d565ef66e7dfff9f98764XX" />
<order okey="919986120a924b00c2bc6d4e65edc51a" type="delivery"
d_date="2018-05-18" b_time="12" e_time="17" inner_id="164541"
city="0" name="Клотильда" address="Москва, ул. Живописная, д 3/3 корп
2, кв 15" contacts="тел. (499) 222-33-221" places="2"
garden_ring="yes" from_mkad="6.88" avoid_part_return="1"
without_opening="1" sms="79037657305" mail="YAANIKA34fqn@GMAIL.COM"
/>
</singleorder>
```

<auth>

ukey - ключ

<order>

okey – уникальный ключ заказа

type – тип изменяемой доставки ("delivery" – означает курьерскую доставку)

d_date - в формате ISO (date Y-m-d), дата исполнения доставки не ранее сегодня

b_time, e_time – интервал доставки в часах с .. по .., (date H:00).
Разница минимум 3 часа, **b_time** с 10 до 19 часов, **e_time** с 15 до 22 часов.

* **inner_id** - внутренний номер заказа в Интернет -магазине (varchar(255))

* **city** - код города доставки (int)

для Москвы **city**"=0

для Санкт-Петербурга **"city"**=1

* **name** – ФИО получателя заказа (varchar(255))

* **address** – почтовый адрес доставки с указанием города (если возможно) и с разделителем ',' между городом, улицей, домом, квартирой (varchar(255))

* **contacts** - контактная информация (varchar(255))

* **places** - кол-во транспортных мест (не товаров!) в данном заказе (int)

* **garden_ring**– признак доставки в пределах зоны платной парковки ('yes'/'no')

* **avoid_part_return** – признак запрета частичного отказа покупателем (1/0)

avoid_part_return="1" – частичный отказ покупателем по заявке запрещен;

avoid_part_return="0" (по умолчанию) – частичный отказ покупателем допускается (используется по умолчанию)

* **without_opening** – признак запрета вскрытия упаковки при доставке заказа покупателю (1/0)

without_opening="1" – вскрытие упаковки при доставке заказа покупателю запрещено; without_opening = "0" (по умолчанию) – разрешено вскрытие упаковки при доставке заказа покупателю.

* **from_mkad** – расстояние от МКАД / КАД (для заявок в СПб) (float(9.2))

* **sms** - телефон для рассылки sms-уведомлений о статусе заказа (varchar(11)), пример корректного номера: 79031234567. Номер будет считаться допустимым, если он начинается с 79, исключая номера, начинающиеся с 7940

* **email** – адрес электронной почты получателя для пересылки уведомлений (varchar(250))

* - необязательные атрибуты и элементы

Изменение выборочных параметров заявок Почты России:

Пример запроса

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>order_correction</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<order okey="61505e7f03a2ec3fba815122a6ec8c27" type="post"
inner_id="почта 333" b_date="2018-12-19"
name="Петр Петров" index="127321" region="Камчатский край"
area="Петропавловск Камчатский2" p_address="ул. Солнечная д.70, кв.
302" contacts="79031234567" sms="79031234567"
email="YAANIKA34fgn1@GMAIL.COM" wrap_type="2" exact_wrap="1"
/>
```

</singleorder>

<auth>

ukey - ключ

<order>

okey – уникальный ключ заказа

type – тип изменяемой доставки ("post" – означает отправку Почтой России)

b_date - предполагаемая дата отправки заказа в формате ISO (date Y-m-d), дата больше текущей

* **inner_id** - внутренний номер заказа в Интернет -магазине (varchar(255))

* **name** – ФИО получателя заказа (varchar(255))

* **index** - индекс получателя (varchar (6))

* **region** - регион получателя (varchar (255))

* **area** - город / район получателя (varchar (255))

* **p_address** - адрес получателя (varchar (255))

* **contacts** - контактный телефон а (varchar(11)), пример корректного номера: 79031234567. Номер будет считаться допустимым, если он содержит 11 знаков, начинается с 79 или 77, исключая номера, начинающиеся с 7940

* **email** – адрес электронной почты получателя для пересылки уведомлений (varchar(250))

* **sms** - телефон для sms-сопровождения заказа (varchar(11)). Пример корректного номера: 79031234567. Номер будет считаться допустимым, если он начинается с 79 или 77, исключая номера, начинающиеся с 7940

* **wrap_type** – тип упаковки отправления (int). Допускаются следующие значения атрибута: 0 - без доупаковки, 1 - без доупаковки (негабарит), 2 - упаковка, 3 - упаковка (негабарит).

* **exact_wrap** – признак строгого соответствия упаковки, указанной в атрибуте wrap_type (int). Допускаются следующие значения атрибута: 1 - строгое соответствие упаковки; 0 – допускается изменение указанного в атрибуте wrap_type типа упаковки.

* - необязательные атрибуты и элементы

Контроль дублирования заявок по внутренним номерам

В периоды повышенной нагрузки ответы от нашего API-сервиса на запросы создания заявок могут поступать с увеличенным интервалом. В зависимости от того, как настроена ваша система, может возникать рассогласование систем, которое может приводить к дублированию заявок в нашей системе.

Например, по истечении определенного в вашей системе таймаута, при условии, что ответ от нашего сервера не поступил, на вашей стороне формируется повторный запрос на создание заявки. В результате в нашу систему поступают два одинаковых запроса, которые обрабатываются независимо друг от друга и, соответственно, в нашей системе будут созданы 2 заявки. При этом заявки в нашей системе будут иметь различные номера, но по сути они будут дубликатами.

Для предотвращения подобных проблем вам следует увеличить таймаут ожидания ответа нашего сервера в вашей системе или воспользоваться функцией контроля дублирования заявок по внутренним номерам.

Принцип работы функции контроля дублирования заявок по внутренним номерам следующий:

В заявках Курьерской доставки, Самовывоза, Почты России и DPD предусмотрено задание внутреннего номера заявки из вашей системы. Этот номер задается в атрибуте `inner_id` ноды `order` в запросе на создание заявки.

При включении функции контроля дублирования заявок по внутренним номерам наш API-сервис будет автоматически отслеживать уникальность ваших внутренних номеров при создании заявок за последний час.

Например, от вас поступает запроса на создание заявки с внутренним номером `inner_id="qwe123"`. Наш сервис выявляет наличие в нашей системе заявки с идентичным внутренним номером `"qwe123"`, которая была создана не ранее часа тому назад. В результате новая заявка создана не будет, а в ответе на ваш запрос будут возвращены параметры выявленной заявки.

Таким образом, даже если от вас в течение часа поступит несколько запросов на создание заявки с внутренним номером `inner_id="qwe123"`, то в результате будет создана только одна заявка, остальные дублирующие запросы по сути исполняться не будут.

По умолчанию функция контроля дублирования заявок по внутренним номерам отключена. Включить функцию вы можете двумя способами:

Вариант 1. В интерфейсе ЛК

В настройках карточки клиента в разделе "Другие настройки" установите флажок в поле "Контроль дублирования заявок на основе внутренних номеров при создании через API"

Другие настройки	
Быстрые деньги (для заявок DPD):	выкл.
Контроль дублирования заявок на основе внутренних номеров при создании через API: <i>Контроль номеров заявок, созданных за последний час, от момента поступления текущего API-запроса</i>	<input checked="" type="checkbox"/>

В этом варианте контроль будут включен для всех заявок Курьерской доставки, Самовывоза, Почты России и DPD, формируемых вами через наш API-сервис

Вариант 1. В контенте запроса на создание заявки

Укажите атрибут `avoid_duplication="1"` в ноде `order` при создании заявок Курьерской доставки, Самовывоза, Почты России и DPD. Например:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<order inner_id="16454" name="Клотильда" address="Москва,
Живописная, д4 корп1, кв 16" from_mkad="0" d_date="2009-06-25"
b_time="12:00" e_time="18:00" incl_deliv_sum="200.15" places="1"
city="0" sms="79031234567" discount_value="auto" discount_unit="0"
avoid_part_return="0" site="www.mysite.ru" avoid_duplication="1">
.....
.....
.....
</order>
</singleorder>
```

В этом варианте контроль будут произведен только для внутреннего номера указанного в конкретном запросе.

Запрос текущего статуса заявки (*метод status*)

Обратный вызов при изменении статуса заявок (*status webhook*)

Вы можете получать автоматическое уведомление об изменении статуса заявок. Для этого вам потребуется подготовить на своей стороне web-скрипт, который будет вызываться POST-запросом при изменении статуса заявок в нашей системе. Такой подход позволит оптимизировать работу вашей системы и уйти от многочисленных опросов статуса заявок через наш API.

Для запуска уведомления через web-скрипт вам потребуется произвести настройку в карточке клиента - указать адрес web-скрипта в поле «Адрес web-вызова на изменение статуса заявки».

Также следует разрешить вашему серверу принимать POST-запросы с нашего домена axiomus.ru

При изменении статуса заявки на указанный вами web-скрипт будут приходить следующие POST параметры:

```
$ _POST['oid'] => 606606
//int OID (номер заказа в системе Axiomus)
$ _POST['status'] => 100
//int status (числовое обозначение статуса, также см. ниже)
$ _POST['info'] => Выполнен
//string (текстовое название статуса заказа)
```

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status</mode>
<okey>2e3023c3e78f4f0c8cbb81257743c2d7</okey>
</singleorder>
```

<okey> - значение **okey**, возвращаемое при создании новой заявки или её изменении

ПРИМЕЧАНИЕ:

Для запроса текущего статуса заявки на тестовой площадке требуется добавлять в запрос ноду

```
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
```

Напр:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<okey>2e3023c3e78f4f0c8cbb81257743c2d7</okey>
</singleorder>
```

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<response>
<request>status</request>
<order id="1013" inner_id="16777" price="156.83"
customer_price="1430.00" incl_deliv_sum="430.00" group="317"
export_order="1179925" fid="61257" />
<status code="211">расчет за доставку</status>
<d_date>2011-03-17</d_date>
</response>

```

<order>

id - номер заявки Аксиомус

inner_id - внутренний номер заявки ИМ

price - цена доставки/забора/самовывоза

customer_price - сумма, которую требуется взять с покупателя

incl_deliv_sum - стоимость доставки для покупателя

export_order - забор или привоз, в который включена заявка

return_order - возврат отказов, в который включена заявка

group - группа обработки заявки

payment_mode - код актуального типа оплаты по заказу для заявок доставки (0=без оплаты;1=наложенный платеж;2=чек;3=карта;4=чек или карта).

fid - если сумма по заказу вошла в финансовую детализацию, то этот атрибут будет содержать номер финансовой детализации.

<status> - текстовое описание статуса зааявки

code - числовой код статуса заявки (подробнее см описание в разделе «Статусы заявок»)

<d_date> - дата заявки

В случае частичного или полного отказа

<refused_items>

<item>

name - название товара

quantity - кол-во, от которого отказались

price - цена товара

В случае отправки через субагента (Почта России, DPD, EMS)

<poststatus>

tracking - трекинг-номер

postprice - стоимость услуг субагента

Для заявок забора, привоза и возврата отказов дополнительно выводится состав пакетов заявки:

<packs >

<pack>

number – внутренний номер заявки ИМ

places- число мест

status- для заборов и привозов [0=не принят, 1=принят]; для возвратов отказов [0=не укомплектован, 1=укомплектован];

Запрос статуса для группы заявок (метод *status_list*)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status_list</mode>
<okeylist>
    <okey>7bbf66349060b723bba20ef687ee4ebf</okey>
    <okey>f2c2635a00310d39cf26b7cc1db6fab2</okey>
    <okey>7bbf66349060b723bba20ef687ee4ebf</okey>
</okeylist>
</singleorder>
```

<okey> - значение **okey**, возвращаемое при создании новой заявки или её изменении.

На тестовой площадке запрос не поддерживается.

ПРИМЕЧАНИЕ 1:

Выдача по запросу ограничена 300 заявками

ПРИМЕЧАНИЕ 2:

Для запроса статуса заявок на тестовой площадке требуется добавлять в запрос ноду

```
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
```

Напр:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status_list</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<okeylist>
    <okey>7bbf66349060b723bba20ef687ee4ebf</okey>
    <okey>f2c2635a00310d39cf26b7cc1db6fab2</okey>
    <okey>7bbf66349060b723bba20ef687ee4ebf</okey>
</okeylist>
</singleorder>
```

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
<request>status_list</request>
<okeylist>
<okey id="1180901" status_code="100" status_name="выполнен"
inner_id="Заказ #878" price="406.02"
customer_price="430.00">7bbf66349060b723bba20ef687ee4ebf</okey>
<okey id="1991064" status_code="100" status_name="выполнен"
inner_id="" price="0.00"
customer_price="">f2c2635a00310d39cf26b7cc1db6fab2</okey>
</okeylist>
</response>
```

<okey>

id - номер заявки Аксиомус

okey – уникальный ключ заказа
status_code - числовой код статуса расчета за доставку
status_name - текстовое описание статуса
inner_id - внутренний номер заявки ИМ
price - цена доставки/забора/самовывоза
exe_date – предполагаемая дата исполнения заявки (или отправки заявки для региональных заявок)
customer_price - сумма, которую требуется взять с покупателя
payment_mode – код актуального типа оплаты по заказу для заявок доставки (0=без оплаты; 1=наложенный платеж; 2=чек; 3=карта; 4=чек или карта).
tracking – трек-номер для региональных заявок (Самовывоз БоксБерри, Почта России, DPD)

ПРИМЕЧАНИЕ:

Если к-л океу переданный в запросе является некорректным, то в ответе сервера ошибка не выдается и некорректный океу игнорируется.

Обратный вызов при изменении статуса заявок (status webhook)

Вы можете получать автоматическое уведомление об изменении статуса заявок. Для этого вам потребуется подготовить на своей стороне web-скрипт, который будет вызываться POST-запросом при изменении статуса заявок в нашей системе. Такой подход позволит оптимизировать работу вашей системы и уйти от многочисленных опросов статуса заявок через наш API.

Для запуска уведомления через web-скрипт вам потребуется произвести настройку в карточке клиента - указать адрес web-скрипта в поле «Адрес web-вызова на изменение статуса заявки».

Также следует разрешить вашему серверу принимать POST-запросы с нашего домена axiomus.ru

При изменении статуса заявки на указанный вами web-скрипт будут приходить следующие POST параметры:

```
$_POST['oid'] => 606606
    //int OID (номер заказа в системе Axiomus)
$_POST['status'] => 100
    //int status (числовое обозначение статуса, также см. ниже)
$_POST['info'] => Выполнен
    //string (текстовое название статуса заказа)
```

Запрос списка заявок за период (метод *get_orders_list*)

Метод позволяет получить список заявок за период с выдачей ограниченного числа параметров этих заявок

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_orders_list</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<orderlist date_from="2018-03-03" date_to="2018-03-07"
status_mode="0" />
</singleorder>
```

<auth>

ukey - ключ

<orderlist> - параметры запроса:

date_from – начальная дата периода выборки заявок (включительно)

date_to – конечная дата периода выборки заявок (включительно)

Прим: интервал (от **date_from** до **date_to**) не должен превышать 31 дня

status_mode – режим выборки заявок. Допустим следующие режимы:

status_mode="0" - выбор всех заявок

status_mode="1" - выбор финальных заявок

status_mode="2" - выбор не финальных заявок

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_orders_list</request>
<orderlist date_from="03.03.2018" date_to="07.03.2018"
status_mode="0">
<order id="7635727" inner_id="16454" date="03.03.2018"
interval="10:00-15:00" type="доставка" status="90"
region_source="Москва" region_destination="Москва"
service_price="0" client_price="0"
apikey="ed0fc1dcdb53637ce9bbcd08cae61342"
/>
<order id="7647455" inner_id="почта 333" date="03.03.2018"
type="Почта России" status="105"
region_source="Москва" region_destination="Регионы"
service_price="100.5" client_price="0"
apikey="3e2294696ff8aba0071d8f7a55bc7876"
/>
.....
</orderlist>
</response>
```

<order> - параметры заявки

id – номер заявки

inner_id – внутренний номер заявки, который вы указали при создании заявки
date – предполагаемая дата исполнения (отправки) заявки
type – тип заявки
status – числовой код статуса заявки (подробнее см описание в разделе «Статусы заявок»)
region_source – регион отправки
region_destination – регион назначения
service_price – стоимость услуг по заявке
client_price – стоимость для получателя
apikey – api-ключ заявки

Запрос ключей заявок (метод *get_okey*)

Метод позволяет получить ключи (*okey*) заявок на основе номеров заявок.

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_okey</mode>
<auth ukey="XXcd208495d565ef66e7dff9f98764XX" />
<orders>
  <order>6158506</order>
  <order>6157524</order>
</orders>
</singleorder>
```

<auth>

ukey - ключ

<order> - номера заявок, по которым предполагается получить ключи (*okey*)

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
<?xml version="1.0" encoding="utf-8" ?>
<response>
<request>get_okey</request>
<orders>
<order objectid="6157524">fbe9c5b7b98ec639df6a5ad5e8db1111</order>
<order objectid="6158506">bbe04443fc2e8ec165a5a7409e231111</order>
</orders>
</response>
```

где

fbe9c5b7b98ec639df6a5ad5e8db1111 – ключ (*okey*) для заявки с номером 6157524
bbe04443fc2e8ec165a5a7409e231111 – ключ (*okey*) для заявки с номером 6158506

Запрос стоимости услуг (*метод `get_price`*)

В настоящее время метод считается устаревшим

Для получения точных значений по стоимости услуг рекомендуем воспользоваться нашим API для детального расчета тарифа - <https://axiomus.ru/wiki/index.php>

Упрощенный запрос стоимости услуг доставки (метод *rough_estimate*)

Рекомендуем также воспользоваться нашим API для детального расчета тарифа - <https://axiomus.ru/wiki/index.php>

Метод позволяет получить оценку стоимости услуг для доставки.

Пример запроса стоимости услуг доставки:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>rough_estimate</mode>
<order order_type="delivery" day_of_weak="3" city="0" from_mkad="5"
garden_ring="no" total_weight="5.35" total_price="5750.56"
sms_inform="yes" cash="yes" cheque="no" card="no">
</order>
</singleorder>
```

<order>

* **order_type** – тип заявки. Допустимые значения: order_type="delivery" (доставка);.

По умолчанию order_type="delivery"

* **day_of_weak** – день недели, на который намечено исполнение заявки (допустимые значения: 1=Пн, 2=Вт, 3=Ср, 4=Чт, 5=Пт, 6=Сб,).

По умолчанию day_of_weak="1"

* **city** - код города доставки (int)

для Москвы city="0"

для Санкт-Петербурга city="1"

По умолчанию city="0"

* **from_mkad** – расстояние от МКАД / КАД (для заявок в СПб) (float(9.2))

По умолчанию from_mkad="0"

* **garden_ring**– признак доставки в пределах зоны платной парковки ('yes'/'no')

По умолчанию garden_ring="0"

* **total_weight**- общий вес доставки (float(9.2))

По умолчанию total_weight="3"

* **total_price**- общая стоимость доставляемых товаров (float(9.2))

По умолчанию total_price="1000"

* **sms_inform**="yes" – информационное сопровождение доставки ('yes'/'no')

По умолчанию sms_inform="no"

* **cash**="yes" - наложенный платеж ('yes'/'no')

По умолчанию cash="no"

* **cheque**="yes" - чек по агентскому договору ('yes'/'no')

По умолчанию cheque="no"

* **card**="yes" – оплата по пластиковой карте ('yes'/'no'). Не допускается для заявок в Санкт-Петербург.

По умолчанию card="no"

Одновременно можно использовать либо **cash**, либо **cheque**, либо **card**, либо **cheque u card**

*** - необязательные атрибуты и элементы**

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>rough_estimate</request>
<order total_price="305,00" />
</response>
```

Запрос списка заявок для передачи на склад (метод *get_export_orders*)

Метод позволяет получить список заявок доставки, самовывоза, отправлений Почтой России, DPD, VoxBerry и др., подлежащих передаче на склад по заборам/привозам.

Примечание: Запрос работает только на основной площадке.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_export_orders</mode>
<auth ukey="fg6ff4f669ff8e346df43ca16bb201a0"/>
</singleorder>
```

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <request>get_export_orders</request>
  <orders>
    <order id=" 4626068" inner_id="43564" places="1"
date="12.01.2016" />
    <order id=" 4626112" inner_id="43565" places="1"
date="13.01.2016" />
    <order id=" 4626156" inner_id="43578" places="1"
date="13.01.2016" />
    <order id=" 4626234" inner_id="43582" places="1"
date="16.01.2016" />
    <order id=" 4626338" inner_id="43591" places="1"
date="17.01.2016" />
  </orders>
</response>
```

где

id - номер заявки в системе Аксиомус

inner_id - внутренний номер заявки в ИМ

places - число мест заявки

date - дата исполнения заявки

Метод также позволяет получить список заявок, которые подлежат передаче на склад по указанному забору/привозу.

Примечание: Запрос работает только на основной площадке.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_export_orders</mode>
<auth ukey="fg6ff4f669ff8e346df43ca16bb201a0"/>
<order id="4627011" />
</singleorder>
```

<order>

id- номер забора/привза в системе Аксиомус

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <request>get_export_orders</request>
  <orders export_id="4627011">
    <order id=" 4626128" inner_id="43564" places="1"
date="12.01.2016" />
    <order id=" 4626340" inner_id="43565" places="1"
date="13.01.2016" />
    <order id=" 4626224" inner_id="43578" places="1"
date="13.01.2016" />
  </orders>
</response>
```

где

id - номер заявки в системе Аксиомус по указанному забору/привозу

inner_id - внутренний номер заявки в ИМ

places - число мест заявки

date - дата исполнения заявки

Аннулирование заявки (метод delete)

Метод позволяет аннулировать созданные ранее заявки.

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>delete</mode>
<auth ukey="8gdn93229223f23b5485d5613e1dbebb4"/>
<okey>dee09cba5b12a8b7c287999a7732e837</okey>
</singleorder>
```

Для удаления через api доступны заявки только в статусе «В обработке»

<auth>

ukey - ключ

<okey >

Значение **okey**, возвращаемое при создании новой заявки или её изменении

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>delete</request>
<order id="2379191" />
</response>
```

ПРИМЕЧАНИЕ:

Если заявка не может быть аннулирована, например, по причине того, что она уже находится в обработке, то будет выдана ошибка

Запрос списка ярлыков заявок (метод *get_label*)

Метод позволяет получить список ярлыков для последующей печати в формате html-документа

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_label</mode>
<auth ukey="b7823229223f23b533329f13e1dbebb4" />
<orders>
    <hash>dgj8620c40be8b333394bcfbe370528a</hash>
    <hash>fd57889cf574526be95f3e647a199d62</hash>
</orders>
</singleorder>
```

<auth>

ukey - ключ

<orders>

указывается список ключей заявок <hash>, для которых требуется получить ярлыки.

Прим: ключ заявки возвращается в ответ на запрос вставки новой заявки.

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_label</request>
<html>
<html><head><title>Ярлык</title><meta
http-equiv=quot;content-typequot; content=quot;text/html;
charset=utf-8quot; /><style type=quot;text/cssquot;
.....
div class=quot;d10quot;><img
src=quot;
AGVv0qAAAAABlBMVEX///8AAABVwtN+AAAAcElEQVRIiWPge72ZgYFYQux5IvEE454/8+
dxMlCPwRChkyMmlzCYGUyzH6z7M01X8x7GeyIMo3wG/UPRL3o/hO/kuspan8of5RpiM7D
EXurb6XrxavA6xq75x717noSKLxop3gJw6jsqOyo7EiVBQD9LV+GtZCgCgAAAABJRU5E
rkJggg==quot;></div></div>
class=quot;d11quot;>Москва, Огородный пр., 20 стр.
38</div></div>
class=quot;d9quot;><b><i>3.4</i></b></div></div>
span class=quot;cuquot;><kr></span></div></div>
class=quot;d5quot;><PB3-1</div></div>
class=quot;d6quot;><Мск</div></div>
class=quot;d7quot;><1/1</div></div>
</div></body></html>
</html>
</response>
```

Контент html-документа списка ярлыков содержится в ноде <html>

Пример вывода html-документа списка ярлыков в браузере:

<div> <div> <div>475`4726</div> <div><Чек></div> <div>(16454)</div> </div> <div> <div>20.10.16</div> <div>10-15</div> </div> </div> <div> <div>ООО "ДПМГ" (BAL)</div> <div>  </div> </div> <div> <div>Москва,</div> <div>Живописная, д 3/3</div> <div>корп1, кв 16</div> </div>			
2.7 _{кг}	Д	Мск	1/1

<div> <div> <div>475`4732</div> <div><Чек></div> <div>(самовыв. 111)</div> </div> <div> <div>20.10.16</div> <div>00-00</div> </div> </div> <div> <div>ООО "ДПМГ" (BAL)</div> <div>  </div> </div> <div> <div>Москва,</div> <div>Огородный пр., 20</div> <div>стр. 38</div> </div>			
3.4 _{кг}	ПВЗ-1	Мск	1/1

Запрос бланков печатных форм заявок (метод *get_printform*)

Метод позволяет получить бланки печатных форм для списка заявок с целью их последующей печати в формате html-документа.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_printform</mode>
<auth ukey="b1193229223f23b548529f13e1dbebb4" />
<orders>
    <hash>dgj8620c40be8b333394bcfbe370528a</hash>
    <hash>fd57889cf574526be95f3e647a199d62</hash>
</orders>
</singleorder>
```

<auth>

ukey - ключ

<orders>

указывается список ключей заявок <hash>, для которых требуется получить печатные формы.

Прим: ключ заявки возвращается в ответ на запрос вставки новой заявки.

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_printform</request>
<html>
    <html><head><title>Печать бланка
заявки</title><meta http-equiv="content-type"
content="text/html; charset=utf-8" /><style
type="text/css"
.....
gt;</tr><tr><td colspan="2"
align="center"></td></tr></table></div><hr
style=""></body></html>
</html>
</response>
```

Контент html-документа для последовательной печати бланков заявок содержится в ноде <html>

Запрос определения ближайшей даты доставки (метод *get_next_delivery*)

Метод позволяет получить ближайшую дату доставки/отправки с учетом типов заявок.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode type="delivery">get_next_delivery</mode>
<auth ukey="b7823229223f23b533329f13e1dbebb4" />
</singleorder>
```

<mode>

type – тип заявки, для которой требуется определить ближайшую дату доставки/отправки:

В случае заявки доставки:

```
<mode type="delivery">get_next_delivery</mode>
```

В случае заявки самовывоза BoxBerry:

```
<mode type="boxberry_pickup">get_next_delivery</mode>
```

В случае заявки забора:

```
<mode type="export">get_next_delivery</mode>
```

В случае заявки привоза:

```
<mode type="self_export">get_next_delivery</mode>
```

В случае заявки отправки Почтой России:

```
<mode type="post">get_next_delivery</mode>
```

В случае заявки отправки DPD:

```
<mode type="dpd">get_next_delivery</mode>
```

В случае заявки отправки в регионы:

```
<mode type="region_pickup">get_next_delivery</mode>
```

```
<mode type="region_courier">get_next_delivery</mode>
```

В случае заявки доставки «Стриж»:

```
<mode type="strizh">get_next_delivery</mode>
```

<auth>

ukey - ключ

Примечание:

Для заявок доставки (type="delivery") в ноде <mode> могут быть указаны дополнительные атрибуты, для указания направления магистральных доставок между городами Москва и Санкт-Петербург

city_from – город отправки [0=Мск; 1=СПб]

city – город назначения [0=Мск; 1=СПб]

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request type="dpd">get_next_delivery</request>
```

```
<date>04.07.2016</date>  
</response>
```

где нода `date` содержит ближайшую доступную дату доставки/отправки, которую вы можете использовать при создании заявки указанного типа.

Запрос создания нового тикета (метод *new_ticket*)

Метод создает новый тикет (запрос в службу поддержки).

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new_ticket</mode>
<auth ukey="b7823229223f23b533329f13e1dbebb4" />
<ticket order="1237756" type="18" >
  <message>test mess<p>my mess<p></message>
</ticket>
</singleorder>
```

<auth>

ukey – ключ

<ticket>

* **order** – номер заявки, с которой связывается тикет (int)

type – тип тикета (int)

Допустимые типы тикетов:

- 18 - Изменение/перенос заказа
- 19 - Отмена заказа
- 20 - Связаться с клиентом!
- 21 - Подготовка детализаций
- 22 - Статус заказа?
- 23 - Стоимость услуг?
- 24 - Документы
- 25 - Возврат отказов
- 26 - Разное
- 27 - Кадры

< message > - контент сообщения

*** - необязательные атрибуты и элементы**

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>new_ticket</request>
<ticket id="150509" />
</response>
```

где

id - номер созданного тикета (по этому номеру следует в последствии добавлять сообщения в тикет методом *add_ticket_mess* и получать переписку по тикету методом *get_ticket*)

Запрос добавления сообщения в тикет (метод *add_ticket_mess*)

Метод добавляет сообщение в тикет.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>add_ticket_mess</mode>
<auth ukey="b7823229223f23b533329f13e1dbebb4" />
<ticket id="150509">
  <message>add mess 2222 &lt;p>my mess 2222 3333
  &lt;p/></message>
</ticket>
</singleorder>
```

<auth>

ukey – ключ

<ticket>

Id – номер тикета, полученный в ответе сервера на запрос new_ticket (int)

Пример ответа сервера

```
<response>
<request>add_ticket_mess</request><query>9fde219c518dd2d1f9aa1dbddf4e
3a20</query>
<ticket id="150509" />
</response>
```

где

id - номер родительского тикета, в которое добавлено сообщение

Запрос переписки по тикету (метод *get_ticket*)

Метод позволяет получить переписку по тикету.

Пример запроса:

```
<singleorder>
<mode>get_ticket</mode>
<auth ukey="b7823229223f23b533329f13e1dbebb4" />
<ticket id="150509" />
</singleorder>
```

<auth>

ukey – ключ

<ticket>

Id – номер тикета, полученный в ответе сервера на запрос *new_ticket* (int)

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_ticket</request>
<ticket>
  <message type="request" time="2016-03-21 00:12:41" company="ИП
Помашко" user="xml-bot" order="1237756" >
    test mess<lt;p>my mess<lt;p>
  </message>
  <message type="response" time="2016-03-21 00:50:06"
company="Аксиомус" user="Колосков" order="1237756" >
    <lt;b>Здравствуйте!<lt;b><lt;br /><lt;p>Ответ
1<lt;p>
  </message>
  <message type="request" time="2016-03-21 00:50:12" company="ИП
Помашко" user="xml-bot" order="1237756" >
    add mess 2222 <lt;p>my mess 2222 3333 <lt;p>
  </message>
  <message type="response" time="2016-03-21 00:50:24"
company="Аксиомус" user="Колосков" order="1237756" >
    <lt;p>Ответ 2<lt;p>
  </message>
</ticket>
</response>
```

где

<message>

type – тип сообщения (request – запрос в службу поддержки, response - ответ службы поддержки)

time – дата и время создания сообщения тикета

company – наименование компании

use – имя пользователя, создавшего сообщение

order – номер заявки, с которой связан тикет (может отсутствовать)

Запрос списка последних тикетов (метод *get_ticketlist*)

Метод позволяет получить список тикетов за последние несколько дней.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_ticketlist</mode>
<auth ukey="b7823229223f23b533329f13e1dbebb4" />
<ticketlist days="2" />
</singleorder>
```

<auth>

ukey – ключ

<ticketlist>

days – число предшествующих дней, за которые просматривается список тикетов (int). Значение должно быть не менее 1 и не более 15

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_ticketlist</request>
<ticketlist>
  <ticket id="50509" time="2016-04-05 00:10:06" status="0"
order="3947001">Изменение/перенос заказа</ticket>
  <ticket id="50510" time="2016-04-05 00:10:32" status="3"
>запрос копии документов </ticket>
</ticketlist>
</response>
```

где

<ticket> - тема тикета

id – номер тикета

time – дата и время создания тикета

status – статус тикета. Атрибут может принимать следующие значения:

0 – ожидается ответ менеджера

1 – получен ответ менеджера

2 – тикет закрыт (не используется)

3 – ожидается ответ от интернет магазина

order – номер заявки, с которой связан тикет (может отсутствовать)

Запрос списка заявок, укомплектованных в возврат отказов (метод *get_returns*)

Метод позволяет получить список заявок, укомплектованных в возврат отказов.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_returns</mode>
<auth ukey="b7823229223f23b533329f13e1dbebb4" />
</singleorder>
```

<auth>

ukey – ключ

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_returns</request><query>9fde219c518dd2d1f9aa1dbddf4e3a20
</query>
<returns>
    <item order="9116890" ret_order="9186382"
inner_id="PLS00007311" ret_date="09.01.2019" ret_mode="Самовывоз" />
    <item order="9139312" ret_order="9186382"
inner_id="PLS00007403" ret_date="09.01.2019" ret_mode="Самовывоз" />
    <item order="9125699" ret_order="9164887"
inner_id="751977" ret_date="10.01.2019" ret_mode="Возврат совместно с
забором №9186947" />
    <item order="9126638" ret_order="9164887"
inner_id="366472" ret_date="10.01.2019" ret_mode="Возврат совместно с
забором №9186947" />
    <item order="9130757" ret_order="9164887"
inner_id="464760" ret_date="10.01.2019" ret_mode="Возврат совместно с
забором №9186947" />
    <item order="9105271" ret_order="9186376"
inner_id="548101" ret_date="10.01.2019" ret_mode="Доставка на адрес"
/>
    <item order="9118797" ret_order="9186376"
inner_id="321028" ret_date="10.01.2019" ret_mode="Доставка на адрес"
/>
</returns>
</response>
```

где

<item> - свойства возврата

order – номер заявки, подлежащей возврату

inner_id – внутренний номер заявки, подлежащей возврату

ret_order – номер возврата отказов, в который укомплектована заявка, подлежащая возврату

ret_date – дата исполнения возврата отказов:

ret_mode – вариант исполнения возврата отказов

Запрос текущей версии API (метод *get_version*)

```
<?xml version='1.0' standalone='yes'?>  
<singleorder>  
<mode>get_version</mode>  
</singleorder>
```

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8" ?>  
<response>  
<request>get_version</request>  
<version>2.11</version>  
</response>
```

Пример отправки запроса с помощью PHP

Ниже представлен пример отправки запроса по протоколу HTTPS, сам запрос должен быть в файле *update_order.xml*

```
<?
//load xml-data from file
$filename = "update_order.xml";
$handle = fopen($filename, "r");
$xml = fread($handle, filesize($filename));
fclose($handle);

$ch = curl_init();

curl_setopt($ch, CURLOPT_URL,
"https://axiomus.ru/hydra/api_xml.php"); // set url to post to
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1); // return into a variable
curl_setopt($ch, CURLOPT_POST, 1); // set POST method
curl_setopt ($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt ($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, "data=".urlencode($xml)); // add
POST fields
$result = curl_exec($ch); // run the whole process

echo $result; //show result on screen

curl_close($ch);
?>
```

Адресный ярлык

На каждом заказе (в том числе, на каждом отдельном транспортном месте заказа), передаваемом в Аксиомус, обязательно должен размещаться **Адресный ярлык**.

Адресный ярлык содержит контрастный черно-белый штрих-код в формате CODE39 с закодированным номером Заявки и номером транспортного места, читаемый штрих-сканерами.

Размер штрихкода должен быть не менее, чем 45x19мм. Размер ярлыка не менее 56x98мм. Также Адресный ярлык должен содержать наименование Заказчика, идентификационный номер Заказа, количество транспортных мест в составном Заказе, общее количество мест в составном Заказе, дату и рекомендуемое время исполнения Заказа, регион доставки Заказа, адрес.

Пример адресного ярлыка



1. Номер заявки (**order_id**), последние 4 символа отделяются тильдой
2. Юридическое название клиента (ООО ..., ИП ..., и т.п.)
3. Штрихкод в формате CODE39, который содержит '**order_id**.'+'транспортное место'.
*Например, если заявка № 1234567 содержит 3 места, то нужно распечатать 3 ярлыка, в которых будет закодированно **1234567+1**, **1234567+2** и **1234567+3***
4. Общий вес заказа
5. Тип заявки
Д – доставка
ВВМск – для заявок самовывоза VoxBerry в регионе Москва. Следует указывать на ярлыке ВВМск для тех заявок, которые оформлены на Московские ПВЗ. Для этих ПВЗ в выгрузке географии (get_boxberry_pickup) установлен атрибут city_code="68"
ВoxВ – для заявок VoxBerry
Р – для заявок на Почту
DPD – для заявок DPD

- R** – для заявок в регионы
ST – для заявок «Стриж»
6. Регион исполнения (**Мск** – Москва, **СПб** – Санкт-Петербург для заявок на доставку. **ПР** – Почта России, **DPD** – для заявок DPD)
 7. Дата исполнения заявки
 8. Время исполнения заявки (для заявок Доставка)
 9. Адрес исполнения заявки (в случае заявок на самовывоз VohBerry – город ПВЗ и получатель)
 10. Номер транспортного места / общее кол-во мест в заказе.
Например, если заявка № 1234567 содержит 3 места, то нужно распечатать 3 ярлыка, в которых будет закодированно 1/3, 2/3 и 3/3
 11. Надпись **<Чек>**. Указывается только в случае, если для заявки требуется чек по агентскому договору (установлен атрибут **cheque**="yes" в ноде **<services>** в запросе создания/изменения заявки).

Для разработчиков

Для работы с нашим API Вы можете использовать библиотеку для языка Ruby, предоставляемую нашим Партнером HttpLab.

Библиотека распространяется бесплатно с открытым исходным кодом и доступна по адресу https://github.com/kinderly/axiomus_api

Рекомендуем также воспользоваться нашим API для детального расчета тарифа - <https://axiomus.ru/wiki/index.php>

Контакты

Мы будем рады ответить на все ваши вопросы по интеграции Axiomus API в вашу систему.

Пожалуйста, напишите нам на info@axiomus.ru или оформите тикет в Личном кабинете.