

Интеграционный гид

МойВыкуп

Версия: 1.1
03 февраля 2026

Содержание

Содержание.....	2
История изменений.....	4
Общее описание API.....	5
Подача тестовых запросов, тестовый кабинет.....	6
Справочник статусов заказов.....	7
Заказ на доставку в регионе Москва и Санкт-Петербург (методы new и update).....	8
Ответ сервера на запрос о создании или изменении заказа.....	14
Коды ответов.....	14
Запрос списка ПВЗ маркетплейсов для самовыкупа (метод get_selfpay_pickup).....	16
Пример ответа сервера.....	16
Заказ на самовыкуп из маркетплейсов (методы new_selfpay_marketplace и update_selfpay_marketplace).....	18
Запрос списка загруженных заказов на самовыкуп из маркетплейсов (метод get_selfpay).....	20
Пример ответа сервера.....	20
Запрос статусов заказов на самовыкуп из маркетплейсов (метод get_selfpay_status).....	23
Пример ответа сервера.....	23
Запрос подготовленных возвратов отказов по заказам самовыкупа из маркетплейсов (метод get_selfpay_returns).....	25
Пример ответа сервера.....	25
Запрос на исполнение подготовленных возвратов отказов (метод selfpay_exec_returns).....	27
Пример ответа сервера.....	27
Контроль дублирования заказов по внутренним номерам.....	29
Запрос текущего статуса заказа (метод status).....	31
Пример ответа сервера.....	32
Запрос статуса для группы заказов (метод status_list).....	34
Пример ответа сервера.....	34
Запрос списка заказов за период (метод get_orders_list).....	36
Пример ответа сервера.....	36
Запрос ключей заказов (метод get_okey).....	38
Пример ответа сервера.....	38
Аннулирование заказа (метод delete).....	39
Пример ответа сервера.....	39
Запрос определения ближайшей даты доставки (метод get_next_delivery).....	40
Пример ответа сервера.....	40
Запрос создания нового тикета (метод new_ticket).....	41
Пример ответа сервера.....	41
Запрос добавления сообщения в тикет (метод add_ticket_mess).....	43
Пример ответа сервера.....	43
Запрос переписки по тикету (метод get_ticket).....	44
Пример ответа сервера.....	44
Запрос списка последних тикетов (метод get_ticketlist).....	46

Пример ответа сервера	46
Запрос списка заказов, укомплектованных в возврат отказов (метод get_returns)	47
Пример ответа сервера	47
Запрос информации о детализации тарифа по заказам (метод detail_tarif)	48
Пример ответа сервера	48
Запрос списка финансовых детализаций (метод get_finlist)	51
Пример ответа сервера	51
Запрос параметров финансовой детализации (метод show_finans)	53
Пример ответа сервера	53
Запрос текущей версии API (метод get_version)	56
Пример ответа сервера	56
Пример отправки запроса с помощью PHP	57
Адресный ярлык	58
Пример адресного ярлыка	58
Контакты	60

История изменений

№ Версии	Изменения

Общее описание API

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 заказа 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://axilog.ru/atlas/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@axilog.ru).

Подача тестовых запросов, тестовый кабинет

Вы можете ознакомиться с демо-версией Личного кабинета Axilog по адресу <https://axilog.ru/demo/> или по адресу https://axilog.ru/alog/lk/client_demo.htm

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

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

Url для тестовых запросов: https://axilog.ru/atlas/api_xml.php

Для создания запросов используйте:

ukey: Qwersd56786786sdfy787232245xx774

uid: 12

Справочник статусов заказов

Ниже указаны коды используемых статусов заказов и их описание:

- 10: Отклонена (Заказ отклонен, причина указана в поле Примечания. Измените или отмените заказ)
- 0: В обработке (Заказ оформлен 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 данному заказу 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, либо доставляется на склад)
- 81: Предотмена (Заявлена предварительная отмена заказа)
- 90: Отмена (Заказ отменена)
- 100: Выполнен (Заказ исполнен)
- 105: Отправлен (Заказ отправлен в внешнюю службу доставки)
- 106: Поступил в ПВЗ (Заказ поступил для исполнения на ПВЗ)
- 107: Вручен (Заказ вручен получателю – при использовании внешней службы доставки)
- 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)

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

Метод позволяет создавать и модифицировать заказы на курьерскую доставку в регионах Москва и Санкт-Петербург.

Пример xml-запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<order inner_id="16454" name="Клотильда" address="Москва, Живописная,
д4 корп1, кв 16" address_zone="2" d_date="2009-06-25" b_time="12:00"
e_time="18:00" incl_deliv_sum="200.15" places="3" sms="79031234567"
discount_value="auto" discount_unit="0" avoid_part_return="0"
city="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" card="no" fastpayment="yes" />
  <items>
    <item name="Крем для лица" weight="0.400" quantity="1"
price="155.00" article="asc456"
mark="(01)02900000338743(21)P0RrgPdl8MujX" />
    <item name="Крем для тела" weight="0.340" quantity="3"
price="235.00" failprice="50" article="cfr127" />
    <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>
  <barcodes>
    <barcode place="1">qwe*34523абк-1</barcode>
    <barcode place="2">qwe*34523ббр-2</barcode>
    <barcode place="3">qwe*34523пвс-3</barcode>
  </barcodes>
</singleorder>
```

<mode>

new – **НОВЫЙ** заказ

update – изменение заказа

<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))

* *address_zone* – зона доставки (int). Допускаются следующие значения:

- доставка в Москве

address_zone="1" (Внутри ТТК)

address_zone="2" (Москва)

address_zone="3" (за пределами МКАД-1)

address_zone="4" (за пределами МКАД-2)

- доставка в Санкт-Петербурге

address_zone="2" (Санкт-Петербург)

address_zone="3" (За пределами СПб 1)

address_zone="4" (За пределами СПб 2)

подробнее по зонам доставки см. информацию в договоре.

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

b_time, *e_time* – интервал доставки в часах с .. по .., (date H:00).

Допускаются следующие интервалы доставки с учетом адресной зоны (см. атрибут *address_zone*):

- доставка в Москве

address_zone="1" (10-14; 14-18; 10-18; 19-22; 15-22; 10-22)

address_zone="2" (10-14; 14-18; 10-18; 19-22; 15-22; 10-22)

address_zone="3" (10-18; 14-22; 10-22)

address_zone="4" (10-22)

- доставка в Санкт-Петербурге

address_zone="2" (10-14; 14-18; 10-18; 19-22; 15-22; 10-22)

address_zone="3" (10-22)

address_zone="4" (10-22)

ПРИМЕЧАНИЕ: Если в атрибутах *b_time*, *e_time* будет указан недопустимый интервал, он автоматически будет расширен до максимального интервала с 10 до 22 часов. В ответе нашего сервера будет выдано предупреждение:

<warning>Изменен временной интервал доставки!</warning>

city – код города доставки (int). Для Москвы *city*"=0; для Санкт-Петербурга *city*"=1

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

* *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 заказ 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.

* `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)`)

* `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 заказ 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 заказу. 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 заказу запрещен;

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

ПРИМЕЧАНИЕ:

Стоимость заказа 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 заказ автоматически

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

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

<services> - дополнительные услуги:

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

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

* *card* - оплата по пластиковой карте ('yes'/'no'). Допускается использование только совместно с указанием *cheque="yes"*. Если в запросе в разделе **<services>** указана единственная опция *card="yes"* система автоматически добавит к ней опцию *cheque="yes"*.

* *fastpayment* - признак допустимости оплаты заказа по системе быстрых платежей ('yes'/'no'). Допускается использование только для доставок по Москве (*city"=0*). Допускается использование совместно с указанием *cheque="yes"* или *cash="yes"*.

ПРИМЕЧАНИЕ: Одновременно можно использовать либо *cash*, либо *cheque*, либо *cheque* и *card*, в соответствии с типом заключенного договора.

<item> - товар

name - наименование 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 *expmode=1*.

* *article* - артикул товара (varchar(50))

* *mark* - маркер товара (для маркируемых товаров) (varchar(50))

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

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

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: Схематичное изображение дифференцированной стоимости доставки

Сумма заказа, руб.:	Цена доставки, руб.:
< 1000	450
< 5000	300
<	
≥ 5000	0
При полном отказе:	450

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: Схематичное изображение дифференцированной скидки

считать в руб. считать в %

Сумма заказа, руб.:	Скидка, руб.:
< 1000	300
< 7000	500
< 15000	600
≥ 15000	800
При полном отказе:	0

Блок штих-кодов мест заказа.

Мы можем принимать и обрабатывать ваши заказы без наших адресных ярлыков. Для этого на каждом отдельном месте заказа должен присутствовать ваш адресный ярлык с уникальным штрихкодом. Использование одинаковых штрихкодов для разных мест заказа недопускается.

Мы настоятельно рекомендуем указывать на ваших адресных ярлыках не только штрихкод, но и информацию о магазине, получателе, адресе, дате и времени доставки – для исключения ошибок в работе курьера.

* **<barcodes>** - Блок ШК мест заказа

<barcode> - штрих-код места заказа

place - номер места (int)

* - необязательные атрибуты и элементы

Ответ сервера на запрос о создании или изменении заказа

В ответ на полученный запрос на создание (**new_**) или изменение (**update_**) заказа 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 обработанного заказа (или id нового заказа 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 изменениях в заказе, которые могут повлиять на последующую обработку информации по заказу и др.

<warning> - содержит текст предупреждения (text)

Коды ответов

0 - запрос обработан и выполнен успешно

1 - ошибка идентификации ikey

2 - ошибка контрольной суммы

3 - ошибка в ФИО, адресе, зоне доставки

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 - заказ не найдена
- 21 - превышена квота заказов
- 22 - превышено допустимое количество запросов в секунду
- 23 - недопустимый запрос
- 24 - не допускается аннулирование заказа
- 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 заказ с указанием cheque="yes" (флажок "Чек" в форме заказа) или наоборот, когда Ваш аккаунт в нашей системе числится как безналичный, а Вы пытаетесь загрузить в систему заказ с указанием cash="yes" (флажок "Наложный платеж" в форме заказа)

Запрос списка ПВЗ маркетплейсов для самовыкупа (метод `get_selfpay_pickup`)

Метод позволяет получить список ПВЗ маркетплейсов для самовыкупов.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_selfpay_pickup</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<points city="0" marketplace="2" />
</singleorder>
```

<auth>

ukey – ключ

***<points>**

* *city* – код города (int). Параметр может принимать следующие значения:

0 – Москва

1 – Санкт-Петербург

По умолчанию *city*="0".

* *marketplace* – код маркетплейса (int). Параметр может принимать

следующие значения:

1 – OZON

2 – Wildberries

По умолчанию *marketplace*="2".

ПРИМЕЧАНИЕ: В настоящее время доступны ПВЗ только для маркетплейса Wildberries (*marketplace*="2")

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_selfpay_pickup</request>
<points>
  <point id="38643" address="Россия, Москва, Бронницкий переулок, 2"
marketplace="2" price_rate="0" />
  <point id="38714" address="Москва, Проезд Стройкомбината 4с1"
marketplace="2" price_rate="0" />
  <point id="38729" address="Москва, Хорошёвское Шоссе 1"
marketplace="2" price_rate="0" />
  .....
  <point id="38535" address="Москва, Коломенская Набережная 22"
marketplace="2" price_rate="0" />
  <point id="38551" address="Москва, Загорьевский Проезд 5а"
marketplace="2" price_rate="0" />
</points>
</response>
```

где

<points> - список доступных ПВЗ

<point> - параметры ПВЗ

id – идентификатор ПВЗ

address – адрес ПВЗ

marketplace – признак маркетплейса. Атрибут может принимать следующие значения:

1 – OZON

2 – Wildberries

price_rate – уровень цен. Атрибут может принимать следующие значения:

0 – Нейтральные ПВЗ (средняя цена)

1 – Хорошие ПВЗ (дешевые)

2 – Плохие ПВЗ (самые дорогие)

3 – Рекомендуемые ПВЗ (спецтариф, дешевле Хороших ПВЗ)

*** - необязательные атрибуты и элементы**

Заказ на самовыкуп из маркетплейсов (методы `new_selfpay_marketplace` и `update_selfpay_marketplace`)

Метод позволяет создавать и модифицировать заказы на самовыкуп из маркетплейсов.

Пример xml-запроса:

```
<?xml version="1.0" standalone="yes"?>
<singleorder>
<mode>new_selfpay_marketplace</mode>
<auth ukey=" Qwersd56786786sdfy787232245xx774" />
<order okey="bfd146001ea475b2139ed8a37c48c6e7" marketplace="2"
d_date="2024-2-26" address="г. Москва, Комсомольская площадь, 1А, стр. 1"
>
    <items>
        <item srcid="126" recipient_name="Татьяна Бояринова"
recipient_phone="+79261234567" name=" Крем для лица; Крем для рук"
weight="0.100" price="1100" barcode="asc456567dg"
recipient_code="345gb2"
barcode_img="
MAAAGVv0qAAAAABlBMVEX///8AAABVwtN+AAAACXBIXMMAA7EAAAOxAGVKw4bAAAAbU1
EQVRIiWNgkz3RwEAswe+0lIFogqFWaNmTiQ1UZIQJL22czTCYQGzW25h5z1i+UDzs3jDK
Z2Bg1LN5V3E+mZlbTvbAKH+UT4gPFOG+118REif66MNxufYVrRzuC+MUqy8wMIzKjsqOy
o5UWQCcTZIUCy0l/wAAAABJRU5ErkJggg==" />
        <item srcid="127" recipient_name="Сергей Васильев"
recipient_phone="+79261234568" recipient_code="235ДК"
barcode="hjkh34593jkg" refuse="1" />
        <item srcid="128" recipient_name="Анна Сергеева"
recipient_phone="+79261234567"
barcode_img="
MAAAGVv0qAAAAABlBMVEX///8AAABVwtN+AAAACXBIXMMAA7EAAAOxAGVKw4bAAAAb01
EQVRIiWPgf1zMwEAsIVT2kHiCcd1euTpuBuoxGCp0csTkEgYzg2nui75PYj+09zDeE2EY
5TPoX4h90fch7BfXVda6Uf4onxCfgSX28qKdF6+8eBUTrrF7Suy5rieBwot2ipcwjMqOy
o7Kj1RZADDdWfMTd2tXAAAAAE1FTkSuQmCC" />
    </items>
</order>
</singleorder>
```

<mode>

`new_selfpay_marketplace` – новый заказ на самовыкуп из маркетплейса

`update_selfpay_marketplace` – изменение заказа

<auth>

`ukey` – ключ

<order>

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

update_selfpay_marketplace

`marketplace` – признак маркетплейса. Допускаются следующие значения:

`marketplace="1"` (ОЗОН)

marketplace="2" (Вайлдберрис)

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

address – почтовый адрес пункта маркетплейса с разделителем ',' между городом, улицей, домом (varchar(255)). Атрибут может быть пропущен в случае, если указан атрибут *point_id*.

* *point_id* – идентификатор ПВЗ (int), см. результаты метода *get_selfpay_pickup* (атрибут *id* ноды *point*).

Если указано значение атрибута *point_id*, то адрес ПВЗ в заказе будет определен автоматически.

<item> - позиции самовыкупа

* *srcid* – идентификатор заказа в вашей системе (varchar(50)). По этому идентификатору вы сможете идентифицировать выкупленные заказы в результатах выдачи метода *status*.

recipient_name – имя получателя (varchar(100))

recipient_phone – телефон получателя (varchar(20))

* *recipient_code* – код получения заказа в маркетплейсе (varchar(30)).

* *barcode* – штрих-код получения заказа в маркетплейсе (varchar(50)).

* *barcode_img* – файл изображения штрих-кода получения заказа в маркетплейсе в кодировке base64. Закодированные данные должны предваряться информацией о файле, например:

data:image/png;base64,...

data:image/jpeg;base64,...

data:image/gif;base64,...

см. также пример html-запроса выше

ПРИМЕЧАНИЕ: Можно передавать какой-то один параметр *barcode* или *barcode_img* – в зависимости от заполнения мы либо используем вашу картинку, либо формируем её самостоятельно.

* *sku_barcode* – штрих-код внутренней складской маркировки (varchar(50)).

* *name* – артикул или наименования выкупаемых товаров, разделенных точкой с запятой (varchar(255)).

* *weight* – вес, положительное число (float(9,3)).

* *price* – цена выкупаемых товаров (float(9,2)).

* *refuse* – признак того, что курьер должен отказаться от выкупаемого товара (*refuse=1* означает, что признак включен).

* - необязательные атрибуты и элементы

Запрос списка загруженных заказов на самовыкуп из маркетплейсов (метод get_selfpay)

Метод позволяет получить список загруженных заказов на самовыкуп из маркетплейсов по их ключам

Пример запроса:

```
<?xml version="1.0" standalone="yes"?>
<singleorder>
<mode>get_selfpay</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<orders>
    <hash>c4879fb6505e0d66152676386b3185fe</hash>
    <hash>caed692354239dbda05eadd2134a7f85</hash>
    <hash>9b34589035e67125b4e9188b804e652a</hash>
</orders>
</singleorder>
```

<auth>

ukey – ключ

<orders>

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

Прим: ключ заказа возвращается в ответ на запрос вставки нового заказа.

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_selfpay</request>
<order_list>
<order id="1905914" key="c4879fb6505e0d66152676386b3185fe"
address="Москва, Улица Борисовские Пруды 24/2" long="37.755518"
lat="55.639532" point_id="48894"
d_date="2025-02-24 00:00:00" b_datetime="2025-02-24 11:00:00"
e_datetime="2025-02-24 22:00:00" places="1"
price="209.00">
<status code="80">*исполнен</status>
<description>автозагрузка из Excel</description>
<items>
    <item srcid="" recipient_name="Зинаида" recipient_phone="+7 987
748-25-70" article="326856241" weight="0.100" price="0.00" refuse="0"
return="0" />
    <item srcid="" recipient_name="Зинаида" recipient_phone="+7 988
713-54-90" article="326860296" weight="0.100" price="0.00" refuse="0"
return="0" />
</items>
</order>
<order id="1906564" key="caed692354239dbda05eadd2134a7f85"
address="Москва, Улица Маршала Голованова 5" long="37.72997"
lat="55.649622" point_id="21051"
d_date="2025-02-25 00:00:00" b_datetime="2025-02-25 11:00:00"
```

```

e_datetime="2025-02-25 22:00:00" places="1"
price="226.00" fact_weight="1.550">
<status code="100">выполнен</status>
<description>автозагрузка из Excel
Число мест возвратов=1</description>
<items>
  <item srcid="" recipient_name="Анжелика" recipient_phone="+7 983
325-37-17" article="320418971" weight="0.100" price="0.00" refuse="0"
return="1" />
  <item srcid="" recipient_name="Анжелика" recipient_phone="+7 981
359-70-76" article="326860296" weight="0.100" price="0.00" refuse="0"
return="1" />
</items>
</order>
<order id="1906570" key="9b34589035e67125b4e9188b804e652a"
address="Москва, Челябинская улица 4к2" long="37.822559" lat="55.775251"
point_id="43208"
d_date="2025-02-25 00:00:00" b_datetime="2025-02-25 11:00:00"
e_datetime="2025-02-25 22:00:00" places="1"
price="130.00">
<status code="80">* исполнен</status>
<description>автозагрузка из Excel</description>
<items>
  <item srcid="205846" recipient_name="Лилия" recipient_phone="+7
981 329-57-16" article="320418971" weight="0.100" price="0.00"
refuse="0" return="0" />
</items>
</order>
</order_list>
</response>

```

<order_list>... </order_list> - результирующий список выгруженных заказов

<order> - параметры выгруженного заказа:

id – номер заказа в системе

key – ключ заказа

address – адрес ПВЗ

long – долгота ПВЗ

lat – широта ПВЗ

point_id – идентификатор ПВЗ из справочника ПВЗ

d_date – дата исполнения заказа

b_datetime – начало интервала исполнения заказа

e_datetime – окончание интервала исполнения заказа

price – тариф по заказу

<status > - статус заказа:

code – код статуса заказа

<description> - комментарии по заказу

<items >... </items > - список самовыкупов по заказу.

<item> - параметры самовыкупа:

srcid – идентификатор заказа в вашей системе

recipient_name – имя получателя

recipient_phone – телефон получателя

article – артикул или наименование выкупаемого товара
weight – заявленный вес выкупаемого товара
price – заявленная стоимость выкупаемого товара
refuse – признак того, что курьер должен отказаться от выкупаемого товара
(refuse=1 означает, что признак включен)
return – признак того, что выкупленный товар принят на складе исполнителя
(return=1 означает, что выкупленный товар принят на складе)

Запрос статусов заказов на самовыкуп из маркетплейсов (метод `get_selfpay_status`)

Метод позволяет получить список загруженных заказов на самовыкуп из маркетплейсов и их статусам за период

Пример запроса:

```
<?xml version="1.0" standalone="yes"?>
<singleorder>
<mode>get_selfpay_status</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<orderlist date_from="2026-01-01" date_to="2026-01-25" />
</singleorder>
```

<auth>

ukey – ключ

<orderlist> - параметры запроса:

date_from – начальная дата периода выборки заказов (включительно)

date_to – конечная дата периода выборки заказов (включительно)

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

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_selfpay_status</request>
<orderlist date_from="01.01.2026" date_to="25.01.2026">
  <order id="1955619" account_id="1557" account_name="ООО Лайм
(ПВЗ)" pvz_type="WB" address="г Москва, Проспект Будённого 17к1"
date="05.01.2026" service_price="249.00"
apikey="4e1290a2ba0f0eeb063f7c0000de1380" status="выполнен"
status_code="100" />
  <order id="1955620" account_id="1557" account_name="ООО Лайм
(ПВЗ)" pvz_type="WB" address="г Москва, Ясногорская Улица 17к1"
date="05.01.2026" service_price="120.00"
apikey="8e55fe28564355c7569669990d3947e2" status="выполнен"
status_code="100" />
  ....
</orderlist>
</response>
```

<orderlist>... </orderlist> - результирующий список выгруженных заказов

<order> - параметры выгруженного заказа:

id – номер заказа в системе

account_id – идентификатор аккунта клиента в системе

account_name – наименование аккунта клиента в системе

pvz_type – тип ПВЗ самовыкупа ('WB' – Wildberries; 'OZON' – ОЗОН)

address – адрес ПВЗ

date – дата исполнения заказа

service_price – тариф по заказу
apikey – ключ заказа для API доступа
status – расшифровка текущего статуса заказа
status_code – код текущего статуса заказа

Запрос подготовленных возвратов отказов по заказам самовыкупа из маркетплейсов (метод get_selfpay_returns)

Метод позволяет получить список подготовленных возвратов отказов по заказам самовыкупа из маркетплейсов, а также состав подготовленных возвратов отказов.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_selfpay_returns</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
</singleorder>
```

<auth>

ukey – ключ

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_selfpay_returns</request>
<orderlist>
  <order id="1957974" account_id="557" account_name="ООО Лрайм
(ПВЗ)" ret_mode="Самовывоз" apikey="6563f3ce3f805258f31e79dbf4983eef"
>
    <items>
      <item id="1957671" srcid="275560" account_id="557"
account_name="ООО Лрайм (ПВЗ)" pvz_type="WB" address="Москва, Шоссейная
Улица 58к3" date="20.01.2026"/>
      <item id="1957691" srcid="59872" account_id="557"
account_name="ООО Лрайм (ПВЗ)" pvz_type="WB" address="Москва, Поречная
улица 21" date="20.01.2026"/>
    </items>
    ....
  </order>
  ....
</orderlist>
</response>
```

<orderlist>... </orderlist> - результирующий список подготовленных возвратов отказов

<order> - параметры возвращаемого заказа самовыкупа:

id – номер заказа в системе

account_id – идентификатор аккунта клиента в системе

account_name – наименование аккунта клиента в системе

ret_mode – режим исполнения возврата отказов

apikey – ключ заказа для API доступа

<items> - список возвращаемых заказов самовыкупа:

id – номер заказа в системе

srcid – номер заказа в вашей системе

account_id – идентификатор аккунта клиента в системе

account_name – наименование аккунта клиента в системе

pvz_type – тип ПВЗ самовыкупа ('WB' – Wildberries; 'OZON' – ОЗОН)

address – адрес ПВЗ

date – дата исполнения заказа

Запрос на исполнение подготовленных возвратов отказов (метод selfpay_exec_returns)

Метод позволяет выдать распоряжение на отгрузку подготовленных возвратов отказов по заказам самовыкупа из маркетплейсов

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>selfpay_exec_returns</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<orders>
    <hash>6563f3ce3f805258f31e79dbf4983eef</hash>
</orders>
</singleorder>
```

<auth>

ukey – ключ

<orders>

Указывается список ключей заказов возвратов отказов <hash>, для которых требуется выдать распоряжение на отгрузку.

ПРИМ:

Ключ заказа на возврат отказов возвращается в ответ на запрос `get_selfpay_returns` в атрибуте `apikey` ноды `order`.

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>selfpay_exec_returns</request>
<orderlist>
    <order id="1957974" apikey="6563f3ce3f805258f31e79dbf4983eef"
    result="accepted">
</orderlist>
</response>
```

<orderlist>... </orderlist> - результирующий список обработанных запросов на отгрузку возвратов отказов

<order> - параметры отгружаемого возврата отказов:

id – номер возврата отказов в системе

apikey – ключ заказа для API доступа

result – результат операции:

`result="accepted"` – запрос на отгрузку принят

`result="rejected"` – запрос на отгрузку отклонен, поскольку ранее уже был зарегистрирован запрос на отгрузку указанного возврата отказов

ПРИМ:

Если в запросе были переданы значения <hash>, для которых не найдены соответствия в системе, то они будут проигнорированы.

Контроль дублирования заказов 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 заказа. В результате в нашу систему поступают два одинаковых запроса, которые обрабатываются независимо друг от друга и, соответственно, в нашей системе будут созданы 2 заказа. При этом заказы 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 заказах Курьерской доставки, Почты России и СДЭК предусмотрено задание внутреннего номера заказа из вашей системы. Этот номер задается в атрибуте `inner_id` ноды `order` в запросе на создание заказа.

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 заказа с внутренним номером `inner_id="test777"`. Наш сервис выявляет наличие в нашей системе заказа с идентичным внутренним номером "test777", которая была создана не ране часа тому назад. В результате новый заказ создан не будет, а в ответе на ваш запрос будут возвращены параметры выявленного заказа.

Таким образом, даже если от вас в течение часа поступит несколько запросов на создание заказа с внутренним номером `inner_id="test777"`, то в результате будет создана только один заказ, 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 (уточните возможность самостоятельной установки этого параметра у менеджера)

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, Почты России и СДЭК, формируемых вами через наш API-сервис

Вариант 2. В контенте запроса на создание заказа

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

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>new</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<order inner_id="16454" name="Клотильда" address="Москва, Живописная,
д4 корп1, кв 16" address_zone="2" d_date="2009-06-25" b_time="12:00"
e_time="18:00" incl_deliv_sum="200.15" places="1" 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 заказа (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 заказа» (уточните возможность самостоятельной установки этого параметра у менеджера).

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

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

```
$_POST['oid'] => 1111112
    //int OID (номер заказа в системе Axilog)
$_POST['status'] => 4
    //int status (числовое обозначение статуса, также см. ниже)
$_POST['info'] => Исполнение
    //string (текстовое наименование статуса заказа)
```

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

<okey> - значение **okey**, возвращаемое при создании нового заказа или его изменении

ПРИМЕЧАНИЕ:

Для запроса текущего статуса заказана тестовой площадке требуется добавлять в запрос ноду

```
<auth ukey="Qwersd56786786sdfy787232245xx774" />
```

Напр :

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<okey>7b30d02a64a37f1f28a19186d80643ef</okey>
</singleorder>
```

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>status</request>
<order id="235" inner_id="im458-02" price="326.35"
customer_price="2510.00" incl_deliv_sum="150.00" group="28"
export_order="253" type="доставка" fid="112" />
<status code="211">расчет за доставку</status>
<d_date>2019-11-17</d_date>
</response>
```

<order>

id - номер заказа Axilog

inner_id - внутренний номер заказа ИМ

type – тип заказа

price - цена доставки/забора/самовывоза

customer_price - сумма, которую требуется взять с покупателя

incl_deliv_sum – стоимость доставки для покупателя

export_order – вывоз или завоз товаров, в который включен заказ

return_order – возврат отказов, в который включен заказ

group – группа обработки заказа

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 заявки

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

<d_date> - дата отправки заказа

В случае частичного или полного отказа

<refused_items>

<item>

name – наименование товара

article – артикул товара

mark – маркировка товара

quantity – кол-во, от которого отказались

price – цена товара

Для заказов самовыкупа также возвращаются следующие параметры:

srcid – идентификатор позиции в вашей системе

recipient_name – имя получателя

recipient_phone – телефон получателя

recipient_code – код получения заказа в маркетплейсе

barcode – штрих-код получения заказа в маркетплейсе

mp_label – QR-лейбл

quantity_on_warehouse – кол-во выкупленного товара

В случае отправки через субагента (Почта России, СДЭК)

<poststatus>

tracking - трекинг-номер

postprice - стоимость услуг субагента

Для заказов типа вывоз товаров, завоз товаров и возврат отказов
дополнительно выводится состав пакетов заказа:

<packs >

<pack>

number – внутренний номер заказа ИМ

places – число мест

status – для заказов типа вывоз товаров и завоз товаров [0=не принят,
1=принят]; для возвратов отказов [0=не укомплектован,
1=укомплектован, 2=возвращен];

Запрос статуса для группы заказов (метод status_list)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status_list</mode>
<okeylist>
  <okey>1dfd1ec1e35e1e2859e812fb695505db</okey>
  <okey>7039a532087b479437dff3f1182464ef</okey>
</okeylist>
</singleorder>
```

<okey> - значение **okey**, возвращаемое при создании нового заказа или его изменения.

ПРИМЕЧАНИЕ 1:

Выдача по запросу ограничена 300 заказами

ПРИМЕЧАНИЕ 2:

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

```
<auth ukey="Qwersd56786786sdfy787232245xx774" />
```

Напр :

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status_list</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<okeylist>
  <okey>1dfd1ec1e35e1e2859e812fb695505db</okey>
  <okey>7039a532087b479437dff3f1182464ef</okey>
</okeylist>
</singleorder>
```

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
<request>status_list</request>
<okeylist>
<okey id="265" status_code="100" type="Доставка" status_name="выполнен"
inner_id="8345-234" price="372.00"
customer_price="1520.00">1dfd1ec1e35e1e2859e812fb695505db</okey>
<okey id="341" status_code="100" type="Почта России"
status_name="выполнен" inner_id="564-3532" price="80.00"
customer_price="2078.00">7039a532087b479437dff3f1182464ef</okey>
</okeylist>
</response>
```

<okey>

id - номер заказа Axilog

inner_id - внутренний номер заказа ИМ
type – тип заказа
okey – уникальный ключ заказа
status_code - числовой код статуса
status_name - текстовое описание статуса
price - тариф по заказу
exe_date – предполагаемая дата исполнения заказа (или отправки заказа для региональных заказов)
customer_price - сумма, которую требуется взять с покупателя
payment_mode – код актуального типа оплаты по заказу для заказов доставки (0=без оплаты;1=наложенный платеж;2=чек;3=карта;4=чек или карта).
tracking – трек-номер для региональных отправок (Почта России, СДЭК)

Примечание

Если к-л okey переданный в запросе является некорректным, то в ответе сервера ошибка не выдается и некорректный okey 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 заказа» (уточните возможность установки параметра у менеджера).

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

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

```
$_POST['oid'] => 1111113
    //int OID (номер заказа в системе Axilog)
$_POST['status'] => 90
    //int status (числовое обозначение статуса, также см. ниже)
$_POST['info'] => Отмена
    //string (текстовое наименование статуса заказа)
```

Запрос списка заказов 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="Qwersd56786786sdfy787232245xx774" />
<orderlist date_from="2018-03-03" date_to="2018-03-07"
status_mode="0" with_positions="1" />
</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

with_positions - режим выборки позиций заявок. Допустим следующие режимы:

with_positions="0" - выбор позиций не производится

with_positions="1" - производится выбор позиций заявок

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_orders_list</request>
<orderlist date_from="26.02.2025" date_to="26.02.2025"
status_mode="0" with_positions="1">
<order id="1807101" inner_id="" address="г Москва, Улица Героев
Панфиловцев 16к1" date="26.02.2025" interval="11:00-22:00"
type="самовыкуп" status="100"
service_price="0" client_price="0"
apikey="f73gh44f5cd5002e2c4fd2d8f1fac0fd">
<items>
```

```
<item srcid="" recipient_name="Глеб Никитич" recipient_phone="+7 (***)
*** 8432" article="318406872" weight="0.100" price="130.00"
refuse="0" return="0" />
</items>
</order>.....
</orderlist>
</response>
```

<orderlist>... </orderlist> - результирующий список выгруженных заказов

<order> - параметры выгруженного заказа:

id – номер заказа

type – тип заказа

address – адрес заказа

date – предполагаемая дата исполнения (отправки) заказа

type – тип заказа

status – числовой код статуса заказа (подробнее см описание в разделе «Статусы заявок»)

service_price – стоимость услуг по заказу

client_price – стоимость для получателя

apikey – api-ключ заказа

<items >... </items > - список позиций по заказу.

<item> - параметры позиции:

Для заказов самовыкупа и возвратов отказов:

srcid – идентификатор заказа в вашей системе

recipient_name – имя получателя

recipient_phone – телефон получателя

article – артикул или наименование выкупаемого товара

weight – заявленный вес выкупаемого товара

price – заявленная стоимость выкупаемого товара

refuse – признак того, что курьер должен отказаться от выкупаемого товара (refuse=1 означает, что признак включен)

return – признак того, что выкупленный товар принят на складе исполнителя (return=1 означает, что выкупленный товар принят на складе)

ordid – идентификатор заказа самовыкупа в системе Аксилог (для заказов возврата отказов)

Для других типов заказов:

name – наименование товара

article – артикул товара

weight – заявленный вес товара

quantity – кол-во товара

price – заявленная стоимость товара

exprmode – признак того, что товар нужно забрать у покупателя (exprmode=1 означает, что признак включен)

return – признак того, что товар (отказной) принят на складе исполнителя (return=1 означает, что товар принят на складе)

Запрос ключей заказов (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="Qwersd56786786sdfy787232245xx774" />
<orders>
  <order>342</order>
  <order>251</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=" 342">55c9ce0c25b15d816319202a33b0e45a</order>
<order objectid=" 251">dec8620c40be8bf2b394bcfbe370528a</order>
</orders>
</response>
```

где

55c9ce0c25b15d816319202a33b0e45a – ключ (okey) для заказа с номером 342

dec8620c40be8bf2b394bcfbe370528a – ключ (okey) для заказа с номером 251

Аннулирование заказа (метод delete)

Метод позволяет аннулировать созданные ранее заказы.

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>delete</mode>
<auth ukey="8gdn93229223f23b5485d5613e1dbebb4" />
<okey>dec8620c40be8bf2b394bcfbe370528a</okey>
</singleorder>
```

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

<auth>

ukey – ключ

<okey >

Значение **okey**, возвращаемое при создании нового заказа или его изменении

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>delete</request>
<order id="342" />
</response>
```

ПРИМЕЧАНИЕ:

Если заказ не может быть аннулирован по причине того, что статус отличен от статуса «В обработке» - будет выдана ошибка

Запрос определения ближайшей даты доставки (метод get_next_delivery)

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

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode type="delivery">get_next_delivery</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
</singleorder>
```

<mode>

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

В случае заказа на доставку:

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

В случае отправки СДЕК:

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

В случае заказа на вывоз товаров:

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

В случае заказа на завоз товаров:

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

В случае отправки Почтной России:

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

<auth>

ukey – ключ

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request type="sdek">get_next_delivery</request>
<date>04.12.2019</date>
</response>
```

где нода *date* содержит ближайшую доступную дату доставки/отправки, которую вы можете использовать при создании заказа 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="Qwersd56786786sdfy787232245xx774" />
<ticket order="178" type="20" >
  <message>просьба срочно связаться с покупателем</message>
</ticket>
</singleorder>
```

<auth>

ukey – ключ

<ticket>

* *order* – номер заказа, с которым связывается тикет (int)

type – тип тикета (int)

Допустимые типы тикетов:

- 18 - Изменение/перенос заказа
- 19 - Отмена заказа
- 20 - Связаться с покупателем!
- 21 - Подготовка финансовой 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="64" />
</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="Qwersd56786786sdfy787232245xx774" />
<ticket id="64">
  <message>просьба отменить заказ</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="64" />
</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="Qwersd56786786sdfy787232245xx774" />
<ticket id="64" />
</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="2019-11-21 13:12:41" company="ИП
Безруков К.С." user="xml-bot" order="178" >
    просьба срочно связаться с покупателем
  </message>
  <message type="response" time="2019-11-21 13:15:06"
company="мойВыкуп" user="Выс" order="178" >
    <b><br /><p>Передали
курьеру</p></b>
  </message>
  <message type="request" time="2019-11-21 13:50:12" company="ИП
Безруков К.С." user="xml-bot" order="178" >
    просьба отменить заказ
  </message>
  <message type="response" time="2019-11-21 13:55:24"
company="мойВыкуп" user="Колосков" order="178" >
    <p>Выполнено</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 заказа, с которым 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="Qwersd56786786sdfy787232245xx774" />
<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="64" time="2019-11-21 13:50:06" status="0"
order="3947001"> Связаться с покупателем!</ticket>
    <ticket id="73" time="2019-11-23 18:10:32" status="3"
>Изменение/перенос заказа</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 заказа, с которым связан тикет (может отсутствовать)

Запрос списка заказов, укомплектованных в возврат отказов (метод get_returns)

Метод позволяет получить список заказов, укомплектованных в возврат отказов.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_returns</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
</singleorder>
```

<auth>

ukey – ключ

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>get_returns</request>
<returns>
  <item order="324" ret_order="412" inner_id="3456-6433"
ret_date="24.11.2019" ret_mode="Самовывоз" />
  <item order="229" ret_order="397" inner_id="8345-99"
ret_date="20.11.2019" ret_mode="Возврат совместно с вывозом №376" />
  <item order="314" ret_order="412" inner_id="3-65765867"
ret_date="24.11.2019" ret_mode="Доставка на адрес " />
  <item order="183" ret_order="412" inner_id="fg789789"
ret_date="24.11.2019" ret_mode="Самовывоз" />
</returns>
</response>
```

где

<item> - параметры возврата

order – номер заказа, подлежащего возврату

inner_id – внутренний номер заказа, подлежащего возврату

ret_order – номер возврата отказов, в который укомплектован заказ, подлежащий возврату

ret_date – дата исполнения возврата отказов:

ret_mode – вариант исполнения возврата отказов

1=укомплектован, 2=возвращен];

Запрос информации о детализации тарифа по заказам (метод detail_tarif)

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>detail_tarif</mode>
<okeylist>
  <okey>6042fa6fc9e097996381588feac0bada</okey>
  <okey>4926de471f7713d413be3ad51447b3e5</okey>
  <okey>f81af43c3cff5cb420343525c052572b</okey>
  <okey>79fb27ca572589608158241e84931ab5</okey>
  <okey>630891010de9113ef7b633782471bb0d</okey>
  <okey>79fb27ca572589608158241e84931ddd</okey>
</okeylist>
</singleorder>
```

<okey> - значение **okey**, возвращаемое при создании нового заказа или его изменения.

ПРИМЕЧАНИЕ 1:

Выдача по запросу ограничена 300 заказами

ПРИМЕЧАНИЕ 2:

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

```
<auth ukey="Qwersd56786786sdfy787232245xx774" />
```

Напр:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>status_list</mode>
<auth ukey="Qwersd56786786sdfy787232245xx774" />
<okeylist>
  <okey>6042fa6fc9e097996381588feac0bada</okey>
  <okey>4926de471f7713d413be3ad51447b3e5</okey>
  <okey>f81af43c3cff5cb420343525c052572b</okey>
  <okey>79fb27ca572589608158241e84931ab5</okey>
  <okey>630891010de9113ef7b633782471bb0d</okey>
  <okey>79fb27ca572589608158241e84931ddd</okey>
</okeylist>
</singleorder>
```

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>detail_tarif</request>
<okeylist>
<okey okey="4926de471f7713d413be3ad51447b3e5" id="130596"
inner_id="автоВывозТоваров" >
```

```

        <tarif_detail>
            <tarif code="exp" name="Услуги вывоза товаров" price="299.00"
        />
    </tarif_detail>
</okey>
<okey okey="6042fa6fc9e097996381588feac0bada" id="130543"
inner_id="W7100009205" >
    <tarif_detail>
        <tarif code="chq" name="Услуги оплаты по чеку" price="52.53"
    />
        <tarif code="dlv" name="Услуги доставки" price="199.00" />
        <tarif code="insur" name="Услуги страхования" price="13.23"
    />
    </tarif_detail>
</okey>
<okey okey="79fb27ca572589608158241e84931ab5" id="120791"
inner_id="160267" >
    <tarif_detail>
        <tarif code="adjust" name="Корректировка" price="15.00"
findoc="3205" comments="Комиссия за обработку НП" />
        <tarif code="adjust" name="Корректировка" price="35.00"
findoc="3205" comments="Комиссия оператора за НП" />
        <tarif code="info" name="Услуги информирования" price="5.00"
findoc="3010" />
        <tarif code="post" name="Услуги предпочтовой подготовки"
price="86.00" findoc="3010" />
        <tarif code="subag" name="Услуги субагента" price="222.38"
findoc="3010" comments="Тариф отправки от 2020-08-19 20:50:40" />
    </tarif_detail>
</okey>
<okey okey="630891010de9113ef7b633782471bb0d" error="Детализация тарифа
недоступна. Заказ не в финальном статусе." id="132444" inner_id="99358C" />
<okey okey="79fb27ca572589608158241e84931ddd" error="Заказ не выявлен"
/>
</okeylist>
</response>

```

<okey>

okey – уникальный ключ заказа

id – номер заказа Axilog

inner_id – внутренний номер заказа ИМ

error – сообщение об ошибке, если детализация тарифа для заказа недоступна

<tarif>

code – код услуги. Список допустимых кодов услуг:

subag - Услуги субагента

dlv - Услуги доставки

chq - Услуги оплаты по чеку

cash - Услуги наличной оплаты

exp - Услуги вывоза товаров

ret - Услуги возврата отказов

doc - Услуги передачи документов

info - Услуги информирования

insur - Услуги страхования

post - Услуги предпочтовой подготовки

adjust - Корректировка

corr - Корректировка услуг агента

name – наименование услуги

price – тариф по услуге

findoc – номер финансовой детализации, в которую включена услуга

Запрос списка финансовых детализаций (метод get_finlist)

Метод позволяет получить список финансовых детализаций за указанный период. Метод работает только на основной площадке.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>get_finlist</mode>
<auth ukey="dddff4f669ff8e346df43ca16bb201a0" />
<finlist date_from="2020-09-01" date_to="2020-10-01" />
</singleorder>
```

<auth>

ukey – ключ

<finlist>

date_from – дата начала периода поиска финансовых детализаций в формате ISO (date Y-m-d).

date_to – дата окончания периода поиска финансовых детализаций в формате ISO (date Y-m-d).

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response><request>get_finlist</request>
<finlist>
  <item number="28" date="2020-09-12" price="14823.00"
status="12" />
  <item number="32" date="2020-09-26" price="2517.20" status="0"
/>
</finlist>
</response>
```

где

<finlist> - список финансовых детализаций

<item> - параметры финансовой детализации

number – номер финансовой детализации

date – дата создания финансовой детализации

price – сумма финансовой детализации

status – статус финансовой детализации. Атрибут может принимать следующие значения:

0 – в обработке

1 – деньги подготовлены

2 – счет выставлен

3 – оплачено частично

4 – принято к переводу

- 6 – заблокирована
- 10 – деньги переданы
- 11 – счет оплачен безналичным переводом
- 12 – деньги переведены безналично

Запрос параметров финансовой детализации (метод show_finans)

Метод позволяет получить параметры финансовой детализации по ее номеру.
Метод работает только на основной площадке.

Пример запроса:

```
<?xml version='1.0' standalone='yes'?>
<singleorder>
<mode>show_finans</mode>
<auth ukey="dddff4f669ff8e346df43ca16bb201a0" />
<finans number="32" />
</singleorder>
```

<auth>

ukey – ключ

<finans>

number – номер финансовой детализации (int), см. одноименный параметр в методе get_finlist.

Пример ответа сервера

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<request>show_finans</request>
<finans number="32" status="0" b_date="04.09.2020"
e_date="27.09.2020" price="2517.20" subtracting="0.00" />
<description></description>
<orders>
  <order id="220" okey="274d31a8c03950ea4a62e3733b502bb9"
d_date="04.09.2020" inner_id="164545" customer_price="2177.00">
    <services>
      <service code="adjustment_price" price="150.00"
comments="Корректировка" />
      <service code="adjustment_price" price="200.00"
comments="Тариф СПб (субагент)" />
      <service code="insurance_price" price="23.03" />
      <service code="inform_price" price="5.00" />
      <service code="cheque_price" price="37.01" />
      <service code="delivery_price" price="199.00" />
    </services>
  </order>
  <order id="314" okey="3af94680b4e011d24aaa051e5b1c9332"
d_date="27.09.2020" inner_id="678-232" customer_price="1496.00">
    <services>
      <service code="insurance_price" price="7.33" />
      <service code="cheque_price" price="25.43" />
      <service code="delivery_price" price="709.00" />
    </services>
  </order>
```

```

    <order id=" 315" okey=" 70d40fdae4a4e73889d7d1cd6c873bae"
d_date=" 26.09.2020" inner_id=" 234-983" customer_price=" 500.00">
      <services>
        <service code=" adjustment_price" price=" 300.00" />
      </services>
    </order>
  </orders>
</response>

```

где

<finans> - параметры финансовой детализации

number – номер финансовой детализации

status – статус финансовой детализации. Атрибут может принимать следующие значения:

0 – в обработке

1 – деньги подготовлены

2 – счет выставлен

3 – оплачено частично

4 – принято к переводу

6 – заблокирована

10 – деньги переданы

11 – счет оплачен безналичным переводом

12 – деньги переведены безналично

b_date – начальная дата периода вхождения исполненных заказов в финансовую детализацию

e_date – конечная дата периода вхождения исполненных заказов в финансовую детализацию

price – сумма финансовой детализации

subtracting – вычет по финансовой детализации

<orders> - список заказов, вошедших в финансовую детализацию

<order> - параметры заказа, вошедшего в финансовую детализацию

id – номер заказа в системе

okey – ключ заказа

d_date – дата исполнения заказа

inner_id – внутренний номер заказа в Интернет –магазине

customer_price – стоимость заказа для получателя, вошедшая в финансовую детализацию (может отсутствовать)

<services> - список услуг, вошедших в финансовую детализацию (может отсутствовать)

<service> - параметры услуги, вошедшей в финансовую детализацию

code – код услуги. Атрибут может принимать следующие значения:

delivery_price – услуга курьерской доставки

cheque_price – услуга оплаты с выпиской чека

cash_price – услуга оплаты наложенным платежом

export_price – услуга вывоза товаров
return_price – услуга возврата отказов
documents_price – услуга передачи документов
inform_price – услуга информирования покупателя
insurance_price – услуга страхования
post_price – услуга предпочтовой подготовки
subagent_price – услуга субагента
adjustment_price – корректировка
price – сумма услуги
comments – комментарии (может отсутствовать)

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>1.1</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($ch, CURLOPT_URL, "https://axilog.ru/atlas/api_xml.php");
// set url to post to
curl_setopt($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 Axilog, обязательно должен размещаться **Адресный ярлык**.

Адресный ярлык содержит контрастный черно-белый штрих-код в формате CODE39 с закодированным номером заказа 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 заказа (**order_id**), последние 4 символа отделяются тильдой
2. Юридическое наименование клиента (ООО ..., ИП ..., и т.п.)
3. Штрихкод в формате CODE39, который содержит '**order_id**.'+'транспортное место'.
*Например, если заказ № 1234567 содержит 3 места, то нужно распечатать 3 ярлыка, в которых будет закодировано **1234567+1**, **1234567+2** и **1234567+3***
4. Общий вес заказа
5. Тип заказа
Д – для курьерской доставки
С – для отправок СДЭК
Р – для отправок Почта России

6. Регион исполнения (**Мск** – Москва, **СПб** – Санкт-Петербург для заказов на доставку. **ПР** – Почта России, **СД** – для отправок СДЭК)
7. Дата исполнения заказа
8. Время исполнения заказа (для заказов курьерской доставки и курьерской отправки СДЭК)
9. Адрес исполнения заказа (в случае отправок на самовывоз СДЭК – город ПВЗ и получатель)
10. Номер транспортного места / общее кол-во мест в заказе.
Например, если заказ № 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 заказа требуется чек по агентскому договору (установлен атрибут **cheque="yes"** в ноде **<services>** в запросе создания/изменения заказа).

Контакты

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

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