

ТЕХНИЧЕСКОЕ ОПИСАНИЕ API МОДУЛЯ «ИНТЕРНЕТ-ЗАКАЗ»

Версия 2.0

Редакция от 15.01.2015

СОДЕРЖАНИЕ

1	ОСНОВНЫЕ СВЕДЕНИЯ	3
2	ПОДКЛЮЧЕНИЕ К API.....	4
3	МЕТОДЫ API	5
3.1	ПРАЙС-ЛИСТ.....	6
3.1.1	СПРАВОЧНИК ТОВАРНЫХ ГРУПП.....	7
3.1.2	СПРАВОЧНИК ТОВАРОВ	8
3.1.3	ДОСТУПНОЕ КОЛИЧЕСТВО ТОВАРОВ И ЦЕНЫ.....	9
3.2	ХАРАКТЕРИСТИКИ И ИЗОБРАЖЕНИЕ ТОВАРОВ	10
3.2.1	ХАРАКТЕРИСТИКА ТОВАРОВ	10
3.2.2	ИЗОБРАЖЕНИЕ ТОВАРОВ.....	11
3.3	РАБОТА С ЗАКАЗАМИ.....	12
3.3.1	СОЗДАНИЕ/РЕДАКТИРОВАНИЕ ЗАГОЛОВКА ЗАКАЗА	13
3.3.2	СОЗДАНИЕ/РЕДАКТИРОВАНИЕ СТРОКИ ЗАКАЗА.....	14
3.3.3	УВЕЛИЧЕНИЕ/УМЕНЬШЕНИЕ ЖЕЛАЕМОГО КОЛИЧЕСТВА ТОВАРА В ЗАКАЗЕ	16
3.3.4	ПЕРЕНОС РЕЗЕРВОВ В НОВЫЙ ЗАКАЗ.....	17
3.3.5	КОМАНДА ДЛЯ ПОДПИСИ ЗАКАЗА	19
3.3.6	УДАЛЕНИЕ ЗАКАЗА	20
3.3.7	РЕЗУЛЬТАТ ОБРАБОТКИ КОМАНД.....	21
3.3.8	СПИСОК ЗАГОЛОВКОВ ЗАКАЗОВ	23
3.3.9	СПИСОК СТРОК ЗАКАЗА	25
3.3.10	СПИСОК СТРОК ЗАКАЗА С ГТД.....	26
3.4	СПРАВОЧНИКИ	27
3.4.1	СПРАВОЧНИК ДОСТУПНЫХ ДАТ ОТГРУЗКИ.....	28
3.4.2	СПРАВОЧНИК ДОСТУПНЫХ МЕТОДОВ ОТГРУЗКИ	28
3.4.3	СПРАВОЧНИК ДОСТУПНЫХ АГЕНТОВ ПО ДОСТАВКЕ.....	29
3.4.4	СПРАВОЧНИК КОНТРАГЕНТОВ ПО КЛИЕНТУ ПАРТНЕРА	29
3.4.5	СПРАВОЧНИК ПРЕДСТАВИТЕЛЕЙ ПО ОСНОВНОМУ ДОГОВОРУ КОНТРАГЕНТА	30
3.4.6	СПРАВОЧНИК КОНЕЧНЫХ ТОЧЕК ДОСТАВКИ	30
3.4.7	СПРАВОЧНИК ТИПОВ УПАКОВКИ	31
3.4.8	СПРАВОЧНИК СТРАН	31
3.4.9	СПРАВОЧНИК КУРС ВАЛЮТЫ.....	31
4	ОГРАНИЧЕНИЯ НА ЧАСТОТУ ЗАПРОСОВ.....	32
5	ПРИМЕРЫ	33
5.1	СТИЛЬ RPC/ENCODED	33
5.2	СТИЛЬ DOCUMENT/LITERAL	35
6	ВЕРСИИ API	38
6.1	API 1.0	38
6.2	API 2.0	38
6.3	РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ В API 2.0.....	39

1 ОСНОВНЫЕ СВЕДЕНИЯ

Web-сервис (API) B2B-модуля «Интернет-заказ» работает по протоколу SOAP (Simple Object Access Protocol) версии 1.1. поверх HTTPS. Обмен данными ведется посредством XML-сообщений, что позволяет реализовать удаленный вызов процедур (RPC - Remote Procedure Call).

Для удобства работы с API размещается WSDL-описание (Web Services Description Language) версии 1.1. — это язык описания Web-сервисов, основанный на XML. На WSDL описываются методы, входные и результирующие структуры данных, типы данных, сетевые адреса для обращения к сервису и т.д.

WSDL-описание сервиса (с различными стилями WSDL) доступно по ссылкам:

- <https://api.merlion.com/dl/mlservice2?wsdl> (стиль document/literal)
- <https://api.merlion.com/re/mlservice2?wsdl> (стиль rpc/encoded)
- <https://api.merlion.com/rl/mlservice2?wsdl> (стиль rpc/literal)

Для тестирования API WSDL-описание тестового сервиса доступно по ссылкам:

- <https://apitest.merlion.com/dl/mlservice2?wsdl> (стиль document/literal)
- <https://apitest.merlion.com/re/mlservice2?wsdl> (стиль rpc/encoded)
- <https://apitest.merlion.com/rl/mlservice2?wsdl> (стиль rpc/literal)

По WSDL-описанию можно построить клиентский класс API, используя любую SOAP-библиотеку. Библиотека скачивает WSDL-описание, анализирует и формирует структуры данных, необходимые для вызова методов. Приложению остается заполнить структуры данными и выполнить запрос. Информация из ответного XML-сообщения выдается в виде структуры, свойственной языку программирования. Все это избавляет приложение от необходимости обрабатывать XML. Существует возможность самостоятельно формировать SOAP-запросы в формате XML и разбирать ответы, однако это трудоемкая задача, и работу с WSDL вручную выполнять нецелесообразно.

Взаимодействие между серверами B2B-модуля «Интернет-заказ» и партнера осуществляется с помощью API.

Сервис позволяет выполнять удаленные запросы к базе данных компании MERLION и получать актуальные данные, выполнять обновление в режиме он-лайн только необходимой части данных, в отличие от скачивания полного прайс-листа в формате XML, формируемого несколько раз в день.

API позволяет:

- Автоматически резервировать в системе товар на складе компании MERLION, а также получать информацию по созданным и ранее отгруженным заказам;
- Подписывать заказ на отгрузку без участия менеджера компании MERLION (при соблюдении общих и финансовых условий);
- Импортировать в учетную систему актуальные цены и доступное для резерва количество товаров на складе по интересующим товарным группам;
- Использовать функциональность сервиса в интернет-магазинах и других торговых площадках для расширения товарного ассортимента и наполнения товарных карточек характеристиками и изображениями товаров.

Работа с API осуществляется по HTTPS (Hypertext Transfer Protocol Secure) протоколу с использованием авторизационных данных B2B-модуля «Интернет-заказ». Параметры подключения к сервису зависят от используемого протокола. Подробную информацию и примеры подключения к API смотрите в разделе «Подключение к API».

2 ПОДКЛЮЧЕНИЕ К API

Подключение к сервису осуществляется по HTTPS протоколу с использованием Basic-аутентификации.

В качестве логина необходимо передавать код клиента и логин, разделяя их символом вертикальной черты "|". В качестве пароля выступает Ваш пароль сервиса.

Пример: ABC0123|MYLOGIN.

ВАЖНО! Для работы с API необходимо создать отдельные логины для каждого пользователя в личном кабинете торговой системы MERLION B2B (<https://b2b.merlion.com/>).

Для самостоятельного создания логина необходимо зайти в B2B с основного логина в Личный кабинет -> Настройки -> Сотрудники -> Добавить нового сотрудника (достаточно завести логин, галки доступа можно не выставлять). Либо обратиться к Вашему менеджеру в компании MERLION для создания нового логина для Вашего Кода партнера.

3 МЕТОДЫ API

ПРАЙС-ЛИСТ:

- **getCatalog()** – справочник товарных групп;
- **getItems()** – справочник товаров;
- **getItemsAvail()** – доступное количество товаров и цен.

ХАРАКТЕРИСТИКИ И ИЗОБРАЖЕНИЕ ТОВАРОВ:

- **getItemsProperties()** – характеристика товаров;
- **getItemsImages()** – изображение товаров.

РАБОТА С ЗАКАЗАМИ:

- **setOrderHeaderCommand()** – создание/редактирование заголовка заказа;
- **setOrderLineCommand()** – создание/редактирование строки заказа;
- **setAddOrderLineCommand()** – увеличение/уменьшение желаемого количества товара в заказе;
- **setMoveOrderLineCommand()** – перенос резервов в новый заказ;
- **setSignOrderCommand()** – команда для подписи заказа;
- **setDeleteOrderCommand()** – удаление заказа;
- **getCommandResult()** – результат обработки команд;
- **getOrdersList()** – список заголовков заказов;
- **getOrderLines()** – список строк заказа;
- **getOrderLinesGTD()** – список строк заказа с ГТД.

СПРАВОЧНИКИ:

- **getShipmentDates()** – справочник доступных дат отгрузки;
- **getShipmentMethods()** – справочник доступных методов отгрузки;
- **getShipmentAgents()** – справочник доступных агентов по доставке;
- **getCounterAgent()** – справочник контрагентов по клиенту партнера;
- **getRepresentative()** – справочник представителей по основному договору контрагента;
- **getEndPointDelivery()** – справочник конечных точек доставки;
- **getPackingTypes()** – справочник типов упаковки;
- **getCountry** – справочник стран;
- **getCurrencyRate()** – справочник курс валюты.

ВАЖНО! Методы API необходимо использовать по правилам, описанным в документе. Это касается заполнения обязательных полей, а также использования корректных и допустимых значений полей.

3.1 ПРАЙС-ЛИСТ

Прайс-лист можно получить с помощью трех отдельных методов:

- **getCatalog()** – для получения справочника товарных групп;
- **getItems()** – для получения справочника товаров;
- **getItemsAvail()** – для получения доступного количества товаров и цен.

Такой подход позволяет получать актуальные данные по всему прайс-листу или только по интересующим товарным группам, обращаясь к каждому методу с разной периодичностью.

Классификация в товарном справочнике построена по принципу:

Направления ⇒ Группы ⇒ Подгруппы ⇒ Товары

Направления содержат группы, группы содержат подгруппы, подгруппы содержат товары. Товары могут находиться только в подгруппах.

Для направлений, групп и подгрупп, содержащих в коде букву «А», она указывается кириллицей.

На расчет доступного количества и цен влияют значения метода отгрузки и даты отгрузки. В зависимости от метода отгрузки определяется, по какому складу и по какому прайс-листу будет произведен расчет данных. От даты отгрузки зависит доступное количество товаров для резерва.

ВАЖНО! При выборе доставочного метода отгрузки (например, "ДОСТАВКА") необходимо указывать дату отгрузки, начиная с завтрашнего дня, т.к. на сегодня доставка не осуществляется (при этом доступное количество товара будет нулевое).

Все данные по количеству товара возвращаются в штуках.

Пример получения прайс-листа:

- 1) Методом **getCatalog()** делаем выгрузку справочника товарных групп;
- 2) Методом **getItems()** получаем справочник товаров;
- 3) Методом **getItemsAvail()** проходимся по нужным нам товарным подгруппам или товарам для получения доступного количества и цен.

3.1.1 СПРАВОЧНИК ТОВАРНЫХ ГРУПП

Для получения справочника товарных групп или обновления только необходимой его части используется метод **getCatalog()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
cat_id	string	код товарной группы	Обязательный параметр!

Также можно использовать одно из следующих значений параметра **cat_id** (код товарной группы):

- **All** – элементы всех уровней классификации (по умолчанию);
- **Order** – список всех направлений (список групп верхнего уровня).

Выходные данные:

Параметр	Тип	Значение	Примечание
ID	string	код группы	
ID_PARENT	string	код родительской группы	
Description	string	название товарной группы	

3.1.2 СПРАВОЧНИК ТОВАРОВ

Для получения справочника товаров используется метод **getItems()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
cat_id	string	код товарной группы	Обязательный параметр , если не указан item_id
item_id	string	код товара	Обязательный параметр , если не указан cat_id
shipment_method	string	код метода отгрузки	Используются значения из справочника getShipmentMethods()
page	int	номер страницы	По умолчанию = 0 (постраничный вывод выключен)
rows_on_page	int	количество записей на странице	По умолчанию 10 000 строк

Важно! Результат запроса к методу **getItems()** может возвращать десятки тысяч строк, что может вызвать ошибку из-за превышения времени выполнения скрипта. Для исключения подобных ситуаций можно использовать постраничный вывод, явно указывая параметры **page** (номер страницы) и **rows_on_page** (количество записей на странице).

При организации цикла для запроса всех страниц, последнюю страницу можно определить двумя способами:

- 1) количество строк в ответе меньше, чем в параметре **rows_on_page**;
- 2) при запросе следующей страницы получили пустой ответ.

Параметр **shipment_method** указывается для определения регионального прайс-листа, по которому будет выводиться товар. По умолчанию информация будет выводиться по прайсу для Московского региона. Также это влияет на минимальное количество товара в упаковке – эта настройка может быть различна для складов.

Выходные данные:

Параметр	Тип	Значение	Примечание
No	string	код товара	
Name	string	наименование товара	
Brand	string	бренд	
Vendor_part	string	партиномер	
Size	string	размер	
EOL	int	признак товара, снятого с производства	<i>EOL – End of life</i>
Warranty	int	срок гарантии	<i>В месяцах</i>
Weight	float	вес (кг)	
Volume	float	объем (м ³)	
Min_Packaged	int	минимальное количество	
GroupName1	string	наименование товарного направления	
GroupName2	string	наименование товарной группы	
GroupName3	string	наименование товарной подгруппы	
GroupCode1	string	код товарного направления	
GroupCode2	string	код товарной группы	
GroupCode3	string	код товарной подгруппы	
IsBundle	int	признак бандла (является ли товар комплектом)	<i>Бандл – комплект, состоящий из нескольких товаров, поставляемых как единое целое</i>

3.1.3 ДОСТУПНОЕ КОЛИЧЕСТВО ТОВАРОВ И ЦЕНЫ

Для получения доступного количества товаров и цен используется метод **getItemsAvail()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
cat_id	string	код товарной группы	Обязательный параметр , если не указан <i>item_id</i>
shipment_method	string	код метода отгрузки	Обязательный параметр! Используются значения из справочника getShipmentMethods()
shipment_date	string	дата отгрузки	Обязательный параметр! Используются значения из справочника getShipmentDates()
only_avail	string	признак вывода по доступности товаров	"0" – все активные товары (по умолчанию); "1" – только доступные для резервирования товары
item_id	string	код товара	Обязательный параметр , если не указан <i>cat_id</i>

Выходные данные:

Параметр	Тип	Значение	Примечание
No	string	код товара	
PriceClient	float	цена клиента	По текущему методу отгрузки, принимает значение PriceClient_RG или PriceClient_MSK
PriceClient_RG	float	цена клиента для региона	
PriceClient_MSK	float	цена клиента для московского региона	
AvailableClient	int	доступное количество товара	Для склада по текущему методу отгрузки, принимает значение AvailableClient_RG или AvailableClient_MSK
AvailableClient_RG	int	доступное количество товара на региональном складе	
AvailableClient_MSK	int	доступное количество товара на московском складе	
AvailableExpected	int	ожидаемый приход товара	
AvailableExpectedNext	int	следующий приход товара	
DateExpectedNext	string	дата следующего прихода товара	

Метод **getItemsAvail()** возвращает цены в USD. Получить рублевые цены можно путем вычисления по курсу валюты из справочника **getCurrencyRate()**.

Важно! Для оптимизации работы рекомендуем использовать метод **getItemsAvail()** по товарным подгруппам.

3.2 ХАРАКТЕРИСТИКИ И ИЗОБРАЖЕНИЕ ТОВАРОВ

3.2.1 ХАРАКТЕРИСТИКА ТОВАРОВ

Для получения характеристик товаров используется метод **getItemsProperties()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
cat_id	string	код товарной группы	Обязательный параметр , если не указан <i>item_id</i>
item_id	string	код товара	Обязательный параметр , если не указан <i>cat_id</i>
page	int	номер страницы	По умолчанию = 0 (постраничный вывод выключен)
rows_on_page	int	количество записей на странице	По умолчанию 10 000 строк
last_days_changed	int	изменения за n последних дней	По умолчанию = 0 (все характеристики)

Важно! Результат запроса к методу **getItemsProperties()** может возвращать десятки тысяч строк даже при запросе к товарной подгруппе, что может вызвать ошибку из-за превышения времени выполнения скрипта. Для исключения подобных ситуаций можно использовать постраничный вывод, явно указывая параметры **page** (номер страницы) и **rows_on_page** (количество записей на странице).

При организации цикла для запроса всех страниц, последнюю страницу можно определить двумя способами:

- 1) количество строк в ответе меньше, чем в параметре **rows_on_page**;
- 2) при запросе следующей страницы получили пустой ответ.

Выходные данные:

Параметр	Тип	Значение	Примечание
No	string	код товара	
PropertyID	int	код характеристики	
PropertyName	string	название характеристики	
Sorting	int	порядок сортировки	
Value	string	значение характеристики	

API выводит только **заполненные** характеристики.

3.2.2 ИЗОБРАЖЕНИЕ ТОВАРОВ

Для получения изображений товаров используется метод **getItemsImages()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
cat_id	string	код товарной группы	Обязательный параметр , если не указан item_id
item_id	string	код товара	Обязательный параметр , если не указан cat_id
page	int	номер страницы	По умолчанию = 0 (постраничный вывод выключен)
rows_on_page	int	количество записей на странице	По умолчанию 10 000 строк

Важно! Результат запроса к методу **getItemsImages()** может возвращать десятки тысяч строк даже при запросе к товарной подгруппе, что может вызвать ошибку из-за превышения времени выполнения скрипта. Для исключения подобных ситуаций можно использовать постраничный вывод, явно указывая параметры **page** (номер страницы) и **rows_on_page** (количество записей на странице).

При организации цикла для запроса всех страниц, последнюю страницу можно определить двумя способами:

- 1) количество строк в ответе меньше, чем в параметре **rows_on_page**;
- 2) при запросе следующей страницы получили пустой ответ.

Выходные данные:

Параметр	Тип	Значение	Примечание
No	string	код товара	
ViewType	string	вид изображения	p – упаковка; v – товар; d – детали товара крупным планом
SizeType	string	тип размера изображения	b – большой; m – средний; s – маленький
FileName	string	имя файла для скачивания	по ссылке http://img.merlion.ru/items/[FileName] , где [FileName] – имя файла с изображением
Created	string	дата и время создания файла	
Size	int	размер изображения	В байтах
Width	int	ширина изображения	
Height	int	высота изображения	

3.3 РАБОТА С ЗАКАЗАМИ

При работе с заказами необходимо знать следующее:

- Заказы состоят из двух частей: заголовков и строки.
- Все модификации заказов осуществляются через очередь команд.
- Каждый метод для модификации заказов возвращает номер команды для последующего отслеживания результата обработки.
- При резервировании товара указывается желаемое количество для резерва. В резерв ставится количество товара, доступное на текущий момент, которое может быть меньше желаемого в зависимости от параметров отгрузки заказа.
- Проверить текущее состояние заказа можно при помощи методов **getOrdersList()** и **getOrderLines()**.
- Учетным заказ становится только после отгрузки товара со склада; до этого момента он считается неучтенным или учетным частично, если отгружается несколькими частями (несколькими документами).

Важно! Если выходным параметром метода является номер команды (**return**), то после выполнения команды необходимо сначала проверить результат ее выполнения с помощью метода **getCommandResult()**. Успешное завершение выполнения команды означает только статус «Сделано».

3.3.1 СОЗДАНИЕ/РЕДАКТИРОВАНИЕ ЗАГОЛОВКА ЗАКАЗА

Для создания/редактирования заголовка заказа используется метод **setOrderHeaderCommand()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер документа	
shipment_method	string	код метода отгрузки	Обязательный параметр! Используются значения из справочника getShipmentMethods()
shipment_date	string	дата отгрузки	Обязательный параметр! Используются значения из справочника getShipmentDates()
counter_agent	string	код контрагента	Используются значения из справочника getCounterAgent()
shipment_agent	string	код агента по доставке	Используются значения из справочника getShipmentAgents()
end_customer	string	конечный заказчик	
comment	string	комментарий к заказу	
representative	string	представитель	Используются значения из справочника getRepresentative()
endpoint_delivery_id	int	код конечной точки доставки	Используются значения из справочника getEndPointDelivery()
packing_type	string	код типа упаковки	Используются значения из справочника getPackingTypes()

Важно! Если не указан параметр **document_no** (номер документа), будет создан новый заказ, в ином случае произойдет изменение указанного заказа.

Выходные данные:

Параметр	Тип	Значение	Примечание
return	int	номер команды	

Для создания заголовка заказа необходимо:

- 1) Отправить команду **setOrderHeaderCommand()** на создание заголовка заказа, параметр **document_no** (номер документа) оставить пустым (сформируется автоматически). Обязательно указать параметры метода и даты отгрузки заказа!
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). В поле **DocumentNo** будет указан номер созданного документа (если команда выполнена успешно). Успешное завершение выполнения команды означает только статус «Сделано».

Для редактирования заголовка заказа необходимо:

- 1) Отправить команду **setOrderHeaderCommand()** на редактирование заголовка заказа, указав параметр **document_no** (номер документа) и значения тех параметров, которые необходимо изменить.
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). Успешное завершение выполнения команды означает только статус «Сделано».

3.3.2 СОЗДАНИЕ/РЕДАКТИРОВАНИЕ СТРОКИ ЗАКАЗА

Для создания/редактирования строки заказа используется метод **setOrderLineCommand()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер документа, в который будет происходить добавление товара	Обязательный параметр!
item_no	string	код товара	Обязательный параметр!
qty	int	желаемое количество товара для резерва	Обязательный параметр!
price	float	желаемая цена	

При **qty** (желаемое количество товара) = 0 произойдет удаление строки из заказа.

Выходные данные:

Параметр	Тип	Значение	Примечание
return	int	номер команды	

Важно! При добавлении товара с минимальным количеством в упаковке больше 1 (параметр **Min_Packaged** (минимальное количество см. в методе **getItems()**)), товар добавится в резерв с количеством, кратным минимальному доступному. В этом случае при проверке строк заказа методом **getOrderLines()** это количество будет указано и в поле **qty** (зарезервированное количество), а желаемое количество в поле **desire_qty** будет изменено к ближайшему минимальному кратному.

Например, в упаковке товара 10 шт. (**Min_Packaged** = 10). Если хотим зарезервировать 23 шт. при доступном количестве для резервирования 10, то в заказе количество (**qty**) будет равно 10 шт., а желаемое количество (**desire_qty**) будет равно 20 шт. Если при тех же условиях хотим зарезервировать 12 шт., то в заказе количество (**qty**) и желаемое количество (**desire_qty**) будут равны 10 шт. Если хотим зарезервировать 8 шт., то в заказе количество (**qty**) и желаемое количество (**desire_qty**) будут равны 0.

Для создания строки заказа необходимо:

- 1) Отправить команду **setOrderLineCommand()** на создание строки заказа, обязательно указав параметры: **document_no** (номер документа), **item_no** (код товара) и **qty** (желаемое количество товара).
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). В поле **ProcessingReserved** будет указано зарезервированное количество товара. Успешное завершение выполнения команды означает только статус «Сделано».
- 4) Повторить команды **setOrderLineCommand()** и **getCommandResult()** для добавления другого товара в заказ.

Для редактирования строки заказа необходимо:

- 1) Отправить команду **setOrderLineCommand()** на редактирование строки заказа, обязательно указав параметры: **document_no** (номер документа), **item_no** (код товара) и **qty** (новое значение количества товара).
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). Если товар в заказе существует, то желаемое количество заменяется новым значением. Успешное завершение выполнения команды означает только статус «Сделано».

Для удаления товара из заказа необходимо:

- 1) Отправить команду **setOrderLineCommand()**, обязательно указав параметры: **document_no** (номер документа), **item_no** (код товара) и **qty** (количества товара) = **0**.
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). Успешное завершение выполнения команды означает только статус «Сделано».

3.3.3 УВЕЛИЧЕНИЕ/УМЕНЬШЕНИЕ ЖЕЛАЕМОГО КОЛИЧЕСТВА ТОВАРА В ЗАКАЗЕ

Для добавления желаемого количества товара в заказе используется метод **setAddOrderLineCommand()**, где входными параметрами являются:

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер документа, в котором будет происходить изменение желаемого количества товара	Обязательный параметр!
item_no	string	код товара	Обязательный параметр!
qty	int	величина изменения желаемого количества товара	Обязательный параметр!
price	float	желаемая цена	

Для параметра **qty** (величина изменения желаемого количества товара – дельта) при указании положительного целого числа (например, «5») произойдет увеличение желаемого количества товара для резерва на указанное число. При указании отрицательного целого числа (например, «-2») произойдет уменьшение желаемого количества товара для резерва на указанное число.

Выходные данные:

Параметр	Тип	Значение	Примечание
return	int	номер команды	

Для изменения желаемого количества товара в заказе необходимо:

- 1) Отправить команду **setAddOrderLineCommand()** на изменение желаемого количества товара в заказе, обязательно указав параметры: **document_no** (номер документа), **item_no** (код товара) и **qty** (величина изменения желаемого количества товара).
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). Если товар в заказе существует, то в поле **ProcessingReserved** будет указано значение, на которое произошло изменение количества товара в заказе. Успешное завершение выполнения команды означает только статус «Сделано».

3.3.4 ПЕРЕНОС РЕЗЕРВОВ В НОВЫЙ ЗАКАЗ

Для переноса резервов в новый заказ используется метод **setMoveOrderLineCommand()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер документа, из которого происходит перенос резервов	Обязательный параметр!
item_no	string	код товара	Обязательный параметр!
qty	int	количество товара для переноса в новый заказ	Обязательный параметр!
operation_no	int	номер команды	

Выходные данные:

Параметр	Тип	Значение	Примечание
return	int	номер команды	

Важно! При переносе резервов создается новый заказ; в него можно перенести резервы только из одного существующего заказа.

Перенос резервов в новый заказ предполагает работу с **пачками команд**.

Пачка – это набор команд, выполняемых в одной транзакции; если возникает ошибка при обработке одной из команд пачки, то идет откат всей транзакции (отмена всех предыдущих выполненных команд этой пачки).

Для переноса резервов в новый заказ необходимо:

- 1) Отправить команду **setMoveOrderLineCommand()** для переноса с параметрами: **document_no** – номер документа, откуда будет перенесен резерв; **item_no** – номер товара для переноса в новый заказ; **qty** – количество товара для переноса; **operation_no** – оставить пустым.
- 2) Отправить команду **setMoveOrderLineCommand()** с параметрами: **document_no** – номер документа, откуда будет перенесен резерв; **item_no** – номер товара для переноса в новый заказ; **qty** – количество товара для переноса; **operation_no** – указать номер команды, полученной на первом шаге (значение выходного параметра **return**).
(Если необходимо перенести больше товаров в новый заказ, то повторите эту команду для других товаров необходимое количество раз.)
- 3) Активировать пачку команд для переноса всех товаров в новый заказ: отправить команду **setMoveOrderLineCommand()** с параметрами: **document_no** – номер документа, откуда будет перенесен резерв; **item_no** – оставить пустым; **qty** – оставить пустым; **operation_no** – указать номер команды, полученной на первом шаге (значение выходного параметра **return**).
- 4) Проверить результат выполнения команды и узнать номер нового документа с помощью метода **getCommandResult()**, указав в параметре **operation_no** номер команды, полученной на первом шаге (значение выходного параметра **return**). В поле **DocumentNo2** будет указан номер нового документа, в который были перенесены резервы. Успешное завершение выполнения команды означает только статус «Сделано».

Пример переноса резервов в новый заказ:

Существует заказ №TC123 с товарами №AAA (3 шт.), №BBB (4 шт.), №CCC (5 шт.). Хотим перенести, например, неподтвержденные вашими клиентами резервы: №AAA (1 шт.), №BBB (2 шт.), №CCC (3 шт.). Выполняем команду со следующими шагами:

- 1) **setMoveOrderLineCommand** (\$document_no = 'TC123',
\$item_no = 'AAA',
\$qty = '1',
\$operation_no = '');

Вернулся номер команды **return** = '123123123'

- 2) **setMoveOrderLineCommand** (\$document_no = 'TC123',
\$item_no = 'BBB',
\$qty = '2',
\$operation_no = '123123123');

- 3) **setMoveOrderLineCommand** (\$document_no = 'TC123',
\$item_no = 'CCC',
\$qty = '3',
\$operation_no = '123123123');

- 4) **setMoveOrderLineCommand** (\$document_no = 'TC123',
\$item_no = '',
\$qty = '',
\$operation_no = '123123123');

- 5) **getCommandResult** (\$operation_no = '123123123');
Узнаем номер нового заказа в поле **DocumentNo2**, например, 'TC456' (если статус команды «Сделано»).

В итоге:

В старом заказе №TC123 останутся товары: №AAA (2 шт.), №BBB (2 шт.), №CCC (2 шт.).

В новом заказе №TC456 будут товары: №AAA (1 шт.), №BBB (2 шт.), №CCC (3 шт.).

3.3.5 КОМАНДА ДЛЯ ПОДПИСИ ЗАКАЗА

Для подписи заказа используется метод **setSignOrderCommand()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер документа	Обязательный параметр!
sign_type	int	тип подписи	По умолчанию = 0 (подписано клиентом)

Важно! Для подписи заказа все поля заголовка заказа обязательно должны быть заполнены! (Редактирование заголовка заказа происходит с помощью метода **setOrderHeaderCommand()**). Представитель и контрагент в заказе должны соответствовать друг другу, иначе заказ также не будет подписан.

Возможные значения параметра **sign_type** (тип подписи):

- **"-1"** – поставить все подписи;
- **"0"** – подписать клиентом (редактирование заказа закончено, менеджер компании MERLION может взять его в работу);
- **"1"** – утвердить заказ (за менеджера компании MERLION);
- **"2"** – подписать заказ на отгрузку (за менеджера и финансового менеджера компании MERLION).

По умолчанию, а также при любых других значениях поля **sign_type** ставится 0 (подписано клиентом).

Уточнить, какие типы подписи доступны, можно у Вашего менеджера в компании MERLION.

Выходные данные:

Параметр	Тип	Значение	Примечание
return	int	номер команды	

Для подписи заказа необходимо:

- 1) Отправить команду **setSignOrderCommand()** на создание строки заказа, указав параметры: **document_no** (номер документа) и **sign_type** (тип подписи).
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). Успешное завершение выполнения команды означает только статус «Сделано».
- 4) Для уточнения текущего статуса подписанного заказа отправить команду **getOrdersList()**, передав в параметр **document_no** номер подписанного документа.

3.3.6 УДАЛЕНИЕ ЗАКАЗА

Для удаления заказа используется метод **setDeleteOrderCommand()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер документа	Обязательный параметр!

Выходные данные:

Параметр	Тип	Значение	Примечание
return	int	номер команды	

Для удаления заказа необходимо:

- 1) Отправить команду **setDeleteOrderCommand()** на удаление заказа, передав номер документа (**document_no**).
- 2) Метод вернет номер команды.
- 3) Проверить результат выполнения команды с помощью метода **getCommandResult()**, указав номер команды (**operation_no**). Успешное завершение выполнения команды означает только статус «Сделано».

3.3.7 РЕЗУЛЬТАТ ОБРАБОТКИ КОМАНД

Для проверки результата обработки команды используется метод **getCommandResult()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
operation_no	string	номер команды	

Если параметр **operation_no** (номер команды) не указан, будет выведено 20 последних команд; для пачки команд будет выводиться только одна строка – заголовок пачки. Пачка – это набор команд, выполняемых в одной транзакции; если возникает ошибка при обработке одной из команд пачки, то идет откат всей транзакции (отмена всех предыдущих выполненных команд этой пачки).

Выходные данные:

Параметр	Тип	Значение	Примечание
operation_no	int	номер команды	
CreateTime	string	время создания команды	
ProcessingTime	string	время начала обработки команды	
EndingTime	string	время окончания обработки команды	
ProcessingResult	string	результат обработки	
DocumentNo	string	номер документа	
DocumentNo2	string	номер нового документа	При переносе резервов в новый заказ с помощью метода setMoveOrderLineCommand() в этом поле выводится номер нового созданного заказа
ProcessingResultComment	string	комментарий результата обработки	
ErrorText	string	текст ошибки	
ProcessingReserved	string	обработанное или зарезервированное количество	
OperationLineNo	int	номер строки в пачке команд	

Возможные значения параметра **ProcessingResult** (результат обработки команд):

- "Активно";
- "Подготовка";
- "Обработка";
- "Сделано";
- "Ошибка";
- "Отменено".

Результат выполнения команд при завершении с ошибкой:

Если при выполнении нескольких последовательных операций одна из них завершилась с ошибкой, это повлияет только на последующие операции, которые так или иначе связаны с ошибочной. Если последующие операции не связаны с ошибочной, они выполняются в обычном режиме.

Например: есть два неподтвержденных заказа, один вчерашний, второй - сегодняшний. Хотим добавить в них товары, но предварительно изменить дату вчерашнего заказа на текущую. Предположим, при изменении даты первого заказа мы ошиблись, операция завершилась с ошибкой, и дата осталась вчерашней. А потом мы сразу стали добавлять в эти два заказа товары. При этом в неизмененный вчерашний заказ товары не добавятся (поскольку дата должна быть корректной), а в сегодняшний товары добавятся успешно.

3.3.8 СПИСОК ЗАГОЛОВКОВ ЗАКАЗОВ

Для получения списка заголовков заказов используется метод **getOrdersList()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер заказа	

Выходные данные:

Параметр	Тип	Значение	Примечание
document_no	string	номер неучтенного документа	
PostedDocumentNo	string	номер учтенного документа	
TNN	string	номер товарной накладной	
OrderDate	string	дата заказа	
Manager	string	ФИО менеджера	
Contact	string	ФИО пользователя, создавшего заказ	
ShipmentMethod	string	метод отгрузки	
ShipmentMethodCode	string	код метода отгрузки	
ShipmentDate	string	дата отгрузки	
ActualShipmentDate	string	дата фактической отгрузки	
CounterpartyClient	string	контрагент	
CounterpartyClientCode	string	код контрагента	
ShippingAgent	string	агент по доставке	
ShippingAgentCode	string	код агента по доставке	
EndCustomer	string	конечный заказчик	
PostingDescription	string	комментарий к заказу	
Weight	string	вес (кг)	
Volume	float	объем (м ³)	
Amount	float	сумма (USD)	
AmountRUR	float	сумма в валюте документа	
WillDeleteTomorrow	string	пометка заказа к удалению	
Status	string	статус заказа	
EndPointCity	string	город доставки	
EndPointAdress	string	адрес доставки	
EndPointContact	string	контактное лицо	
PackingType	string	тип упаковки	
Representative	string	представитель	

Если не передан параметр **document_no** (номер документа), будет выведен список заголовков всех неотгруженных заказов. Если номер документа передан, будет выведена информация по указанному заказу. Если заказ отгружен, то будет выведен номер учтенного документа (**PostedDocumentNo**) и дата фактической отгрузки заказа (**ActualShipmentDate**). Если заказ отгружался по нескольким документам, будет выведена информация по заголовкам каждого учтенного документа; в таком случае номера учтенных документов могут содержать: «номер заказа» + «две заглавные буквы i» + «код логистического объекта».

Параметр **WillDeleteTomorrow** (пометка заказа к удалению при истечении срока жизни резерва) ставится накануне, содержит дату удаления заказа; такие заказы удаляются ночным заданием.

Возможные значения параметра **Status** (статус заказа):

- "–" (когда нет других статусов; заказ доступен для редактирования)
- "подписан клиентом" (заказ утвержден для согласования Вашим менеджером в компании MERLION);
- "утвержден" (заказ одобрен Вашим менеджером в компании MERLION, но не отправлен на отгрузку);
- "комплектуется" (заказ комплектуется, отправлен на отгрузку);
- "отгружается";
- "отгружен";
- "учтен"

Редактирование заказа доступно только до присвоения статуса. Если статус указан – редактировать заказ уже нельзя.

3.3.9 СПИСОК СТРОК ЗАКАЗА

Для получения строк заказа используется метод **getOrderLines()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер заказа	Обязательный параметр!
details	string	детализация заказа	По умолчанию = 0 (без детализации)

Возможные значения параметра **details** (детализация заказа):

- "0" – без детализации (по умолчанию);
- "1" – с детализацией (с разбивкой по отгрузочным документам);
- "2" – с детализацией только по неучтенным документам;
- "3" – с детализацией только по учтенным документам

Выходные данные:

Параметр	Тип	Значение	Примечание
item_no	string	код товара	
document_no	string	номер документа	
qty	int	зарезервированное количество	
desire_qty	int	желаемое количество	
shipped_qty	int	отгруженное количество	
price	float	цена	
amount	float	сумма	
desire_price	float	желаемая цена	
weight	float	вес (кг)	
volume	float	объем (м ³)	

Если заказ отгружался по нескольким документам, будет выведена информация по строкам каждого учтенного документа; в таком случае номера учтенных документов могут содержать: «номер заказа» + «две заглавные буквы i» + «код логистического объекта».

3.3.10 СПИСОК СТРОК ЗАКАЗА С ГТД

Для получения строк заказа с грузовой таможенной декларацией (ГТД) (по отгруженным заказам) используется метод **getOrderLinesGTD()**, где входным параметром является:

Входные параметры:

Параметр	Тип	Значение	Примечание
document_no	string	номер заказа	Обязательный параметр!

Выходные данные:

Параметр	Тип	Значение	Примечание
document_no	string	номер документа	
item_no	string	код товара	
bundle_item_no	string	код товара из состава бандла (для простого товара будет не заполнен)	<i>Бандл – комплект, состоящий из нескольких товаров, поставляемых как единое целое</i>
gtd_no	string	номер ГТД	
used_qty	int	использованное количество ГТД	
country	string	код страны производителя	

Важно! ГТД предоставляется только после формирования счёта-фактуры.

Название страны можно определить с помощью справочника **getCountry()**, передав в параметр **code** (код страны) полученное значение параметра **country** (код страны производителя).

3.4 СПРАВОЧНИКИ

ВАЖНО! При использовании справочников необходимо организовать локальное хранение полученных данных.

Справочники рекомендуется обновлять раз в сутки. При выборе времени обновления следует учитывать следующее:

- доступные даты отгрузки обновляются после смены суток, и в течение дня не меняются;
- курс валюты на следующий день выставляется вечером после 17:00 по мере его публикации; на текущий день он уже есть и в течение дня не меняется.

Если метод использует в качестве входного параметра справочную информацию, следует передавать данные из локальной базы, избегая при каждом запросе повторного вызова справочника. Не следует вызывать справочник каждый раз в любом другом методе, который требует справочной информации – это приводит к избыточной нагрузке и уменьшает скорость ответа.

3.4.1 СПРАВОЧНИК ДОСТУПНЫХ ДАТ ОТГРУЗКИ

Для вызова справочника доступных дат отгрузки используется метод **getShipmentDates()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
code	string	код даты отгрузки	По умолчанию «пусто» (другие значения не используются)

Выходные данные:

Параметр	Тип	Значение	Примечание
Date	string	дата отгрузки	

Доступные даты отгрузки – ближайшие несколько рабочих дней, считая сегодняшний день. Перечень доступных дат отгрузки не зависит от склада, этот общий список.

3.4.2 СПРАВОЧНИК ДОСТУПНЫХ МЕТОДОВ ОТГРУЗКИ

Для вызова справочника доступных методов отгрузки используется метод **getShipmentMethods()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
code	string	код метода отгрузки	По умолчанию «пусто»

Выходные данные:

Параметр	Тип	Значение	Примечание
Code	string	код метода отгрузки	
Description	string	описание метода отгрузки	
IsDefault	int	используется по умолчанию, если =1	

Важно! Код метода отгрузки – это способ отгрузки; он определяет, с какого склада будет отгружаться товар и каким способом (самовывоз или доставка).

Уточнить или изменить доступные методы отгрузки можно у Вашего менеджера в компании MERLION, который может настроить через методы отгрузки привязки региональных складов. Это даст возможность из общего списка выбрать необходимый метод.

3.4.3 СПРАВОЧНИК ДОСТУПНЫХ АГЕНТОВ ПО ДОСТАВКЕ

Для вызова справочника доступных агентов по доставке используется метод **getShipmentAgents()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
code	string	код агента по доставке	По умолчанию «пусто»

Выходные данные:

Параметр	Тип	Значение	Примечание
Code	string	код агента по доставке	
Description	string	описание агента по доставке	

В справочнике указаны транспортные компании клиента. Уточнить, какие агенты по доставке доступны, или изменить их можно у Вашего менеджера в компании MERLION.

3.4.4 СПРАВОЧНИК КОНТРАГЕНТОВ ПО КЛИЕНТУ ПАРТНЕРА

Для вызова справочника контрагентов по клиенту партнера используется метод **getCounterAgent()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
code	string	код контрагента по клиенту партнера	По умолчанию «пусто»

Выходные данные:

Параметр	Тип	Значение	Примечание
Code	string	код контрагента по клиенту партнера	
Description	string	описание контрагента по клиенту партнера	

Уточнить, какие контрагенты по клиенту доступны, или изменить их можно у Вашего менеджера в компании MERLION.

3.4.5 СПРАВОЧНИК ПРЕДСТАВИТЕЛЕЙ ПО ОСНОВНОМУ ДОГОВОРУ КОНТРАГЕНТА

Для вызова справочника представителей по основному договору контрагента используется метод **getRepresentative()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
CounterAgentCode	string	код контрагента	По умолчанию «пусто». Используются значения из справочника getCounterAgent()

Выходные данные:

Параметр	Тип	Значение	Примечание
Representative	string	ФИО представителя	
CounterAgentCode	string	код контрагента	
StartDate	string	дата начала срока действия	
EndDate	string	дата окончания срока действия	

В справочнике указаны доверенные лица на получение товара.

В заголовке заказе необходимо указывать представителя, действующего на дату отгрузки заказа.

Уточнить, какие представители по основному договору доступны, или изменить их можно у Вашего менеджера в компании MERLION.

3.4.6 СПРАВОЧНИК КОНЕЧНЫХ ТОЧЕК ДОСТАВКИ

Для вызова справочника конечных точек доставки используется метод **getEndPointDelivery()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
id	int	код конечной точки доставки	По умолчанию «пусто»
ShippingAgentCode	string	код агента по доставке	По умолчанию «пусто»

Выходные данные:

Параметр	Тип	Значение	Примечание
ID	int	код записи	
Endpoint_address	string	адрес доставки	
Endpoint_contact	string	контактное лицо	
ShippingAgentCode	string	код агента по доставке	
City	string	город доставки	

В справочнике указаны адреса доставки товара. Уточнить, какие конечные точки доставки доступны, или изменить их можно у Вашего менеджера в компании MERLION.

3.4.7 СПРАВОЧНИК ТИПОВ УПАКОВКИ

Для вызова справочника типов упаковки используется метод **getPackingTypes()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
code	string	код типа упаковки	По умолчанию «пусто»

Выходные данные:

Параметр	Тип	Значение	Примечание
Code	string	код типа упаковки	
Description	string	описание	

В справочнике указано, в каких упаковках доставляется товар. Уточнить, какие типы упаковки доступны, или изменить их можно у Вашего менеджера в компании MERLION.

3.4.8 СПРАВОЧНИК СТРАН

Для вызова справочника стран используется метод **getCountry()**, где входными параметрами являются:

Входные параметры:

Параметр	Тип	Значение	Примечание
code	string	код страны	

Выходные данные:

Параметр	Тип	Значение	Примечание
Code	string	код страны	
Description	string	название страны	

Справочник позволяет узнать наименование страны по ее коду.

3.4.9 СПРАВОЧНИК КУРС ВАЛЮТЫ

Для вызова справочника по курсу валюты используется метод **getCurrencyRate()**.

Входные параметры:

Параметр	Тип	Значение	Примечание
date	string	дата курса валюты	По умолчанию текущая дата

Выходные данные:

Параметр	Тип	Значение	Примечание
Code	string	код валюты	
Date	string	дата курса валюты	
ExchangeRate	float	курс валюты	

4 ОГРАНИЧЕНИЯ НА ЧАСТОТУ ЗАПРОСОВ

При использовании API необходимо учитывать следующие ограничения на частоту запросов:

Метод API	Интервал (в секундах)	Количество запросов
Default (по умолчанию, если не задано для метода)	1	20
helloWorld	10	5
getCatalog()	1	1
getItems	1	3
getItemsAvail	1	5
getItemsProperties	1	3
getItemsImages	1	3
getOrdersList()	1	1
getOrderLines()	1	1
getOrderLinesGTD()	1	1
getShipmentDates	60	1
getShipmentMethods	60	1
getShipmentAgents	60	1
getCounterAgent	60	1
getRepresentative	60	1
getEndPointDelivery	60	1
getPackingTypes	60	1
getCountry	60	1
getCurrencyRate	60	1

Превышение указанных ограничений вызывает ошибку.

5 ПРИМЕРЫ

5.1 СТИЛЬ RPC/ENCODED

ПРИМЕР СКРИПТА SOAP-КЛИЕНТА НА PHP (СТИЛЬ RPC/ENCODED):

```
<?php
    $wsdl_url = "https://api.merlion.com/re/mlservice2?wsdl";

    $params = array('login' => "ABC0123|MYLOGIN"
        , 'password' => "LIzsl dhf9"
        , 'encoding' => "Windows-1251"
        , 'features' => SOAP_SINGLE_ELEMENT_ARRAYS
    );

    try {
        $client = new SoapClient($wsdl_url, $params);

        $cat = $client->getCatalog('N1');

        foreach ($cat->item as $row) {
            echo $row->ID_PARENT." - ".$row->ID." - ".$row->Description."<br>";
        }
    } catch (SoapFault $E) {
        echo $E->faultstring;
    }

?>
```

РЕЗУЛЬТАТ РАБОТЫ СКРИПТА:

```
N1 - N103 - Аксессуары для планшетных устройств
N1 - N101 - Аксессуары и опции для ноутбуков
N1 - N100 - Ноутбуки
N1 - N102 - Планшетные устройства
```

ПРИМЕР SOAP-ЗАПРОСА (СТИЛЬ RPC/ENCODED):

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ns1="https://api.merlion.com/re/mlservice2"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:enc="http://www.w3.org/2003/05/soap-encoding">
  <env:Body>
    <ns1:getCatalog env:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
      <cat_id xsi:type="xsd:string">N1</cat_id>
    </ns1:getCatalog>
  </env:Body>
</env:Envelope>
```

ПРИМЕР SOAP-ОТВЕТА:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ns1="https://api.merlion.com/re/mlservice2"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:enc="http://www.w3.org/2003/05/soap-encoding">
  <env:Body xmlns:rpc="http://www.w3.org/2003/05/soap-rpc">
    <ns1:getCatalogResponse env:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
      <rpc:result>return</rpc:result>
      <return xsi:type="ns1:ArrayOfCatalogResult">
        <item xsi:type="ns1:CatalogResult"><ID
xsi:type="xsd:string">N103</ID>
          <ID_PARENT xsi:type="xsd:string">N1</ID_PARENT>
          <Description xsi:type="xsd:string">Аксессуары для планшетных
устройств</Description>
        </item>
        <item xsi:type="ns1:CatalogResult"><ID
xsi:type="xsd:string">N101</ID>
          <ID_PARENT xsi:type="xsd:string">N1</ID_PARENT>
          <Description xsi:type="xsd:string">Аксессуары и опции для
ноутбуков</Description>
        </item>
        <item xsi:type="ns1:CatalogResult"><ID
xsi:type="xsd:string">N100</ID>
          <ID_PARENT xsi:type="xsd:string">N1</ID_PARENT>
          <Description xsi:type="xsd:string">Ноутбуки</Description>
        </item>
        <item xsi:type="ns1:CatalogResult"><ID
xsi:type="xsd:string">N102</ID>
          <ID_PARENT xsi:type="xsd:string">N1</ID_PARENT>
          <Description xsi:type="xsd:string">Планшетные
устройства</Description>
        </item>
      </return>
    </ns1:getCatalogResponse>
  </env:Body>
</env:Envelope>
```

5.2 СТИЛЬ DOCUMENT/LITERAL

ПРИМЕР СКРИПТА SOAP-КЛИЕНТА НА PHP (СТИЛЬ DOCUMENT/LITERAL):

```
<?php
$wsdl_url = "https://api.merlion.com/dl/mlservice2?wsdl";

$params = array('login' => "ABC0123|MYLOGIN"
    , 'password' => "LIzsl dhf9"
    , 'encoding' => "Windows-1251"
    , 'features' => SOAP_SINGLE_ELEMENT_ARRAYS
);

try {
    $client = new SoapClient($wsdl_url, $params);

    class Request {
        public $cat_id;
        public $item_id;
        public $shipment_method;
        public $page;
        public $rows_on_page;
    }

    $req = new Request();
    $req->cat_id = "N1";
    $req->shipment_method = "ДОСТАВКА";
    $req->page = 2;
    $req->rows_on_page = 5;

    $items = $client->getItems($req);

    foreach ($items->getItemsResult->item as $row) {
        echo $row->No." - ".$row->Brand." - ".$row->Name."<br>";
    }
} catch (SoapFault $E) {
    echo $E->faultstring;
}

?>
```

РЕЗУЛЬТАТ РАБОТЫ СКРИПТА:

```
551673 - SONY - Адаптер Sony VGP-AC19V32 для FW,NS,CS,SR,Z,NW серии
552246 - LENOVO - Клавиатура Lenovo USB Enhanced Performance / черная
552517 - HP - Док-станция HP 2010 120W Advanced Docking Station (NZ222AA)
552518 - HP - Док-станция HP 2010 230W Advanced Docking Station (NZ223AA)
552524 - HP - Адаптер HP 65W Smart Travel AC Adapter (AU155AA)
```

ПРИМЕР SOAP-ЗАПРОСА (СТИЛЬ DOCUMENT/LITERAL):

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns1="https://api-
iz.merlion.com/dl/mlservice2">
  <env:Body>
    <ns1:getItems>
      <cat_id>N1</cat_id>
      <item_id xsi:nil="true"/>
      <shipment_method>ДОСТАВКА</shipment_method>
      <page>2</page>
      <rows_on_page>5</rows_on_page>
    </ns1:getItems>
  </env:Body>
</env:Envelope>
```

ПРИМЕР SOAP-ОТВЕТА:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns1="https://api-iz.merlion.com/dl/mlservice2">
  <env:Body>
    <ns1:getItemsResponse>
      <getItemsResult>
        <item>
          <No>551673</No>
          <Name>Адаптер Sony VGP-AC19V32 для FW,NS,CS,SR,Z,NW серии</Name>
          <Brand>SONY</Brand>
          <Vendor_part>VGP-AC19V32.CEL3</Vendor_part>
          <Size>Economy</Size>
          <EOL>1</EOL>
          <Warranty>1</Warranty>
          <Weight>0.95</Weight>
          <Volume>0.00328</Volume>
          <Min_Packaged>0</Min_Packaged>
          <GroupName1>НОУТБУКИ</GroupName1>
          <GroupName2>Аксессуары и опции для ноутбуков</GroupName2>
          <GroupName3>Адаптеры и блоки питания</GroupName3>
          <GroupCode1>N1</GroupCode1>
          <GroupCode2>N101</GroupCode2>
          <GroupCode3>N10100</GroupCode3>
          <IsBundle>0</IsBundle>
        </item>
        <item>
          <No>552246</No>
          <Name>Клавиатура Lenovo USB Enhanced Perfomance / черная</Name>
          <Brand>LENOVO</Brand>
          <Vendor_part>73P2646</Vendor_part>
          <Size>Commercial</Size>
          <EOL>0</EOL>
          <Warranty>1</Warranty>
          <Weight>1.62</Weight>
          <Volume>0.0075</Volume>
          <Min_Packaged>0</Min_Packaged>
          <GroupName1>НОУТБУКИ</GroupName1>
          <GroupName2>Аксессуары и опции для ноутбуков</GroupName2>
          <GroupName3>Клавиатуры и мыши</GroupName3>
          <GroupCode1>N1</GroupCode1>
          <GroupCode2>N101</GroupCode2>
          <GroupCode3>N10109</GroupCode3>
          <IsBundle>0</IsBundle>
        </item>
      </getItemsResult>
    </ns1:getItemsResponse>
  </env:Body>
</env:Envelope>
```

```

        <No>552517</No>
        <Name>Док-станция HP 2010 120W Advanced Docking Station
(NZ222AA) </Name>
        <Brand>HP</Brand>
        <Vendor_part>NZ222AA</Vendor_part>
        <Size>Части</Size>
        <EOL>1</EOL>
        <Warranty>12</Warranty>
        <Weight>3.35</Weight>
        <Volume>0.01523</Volume>
        <Min_Packaged>0</Min_Packaged>
        <GroupName1>НОУТБУКИ</GroupName1>
        <GroupName2>Аксессуары и опции для ноутбуков</GroupName2>
        <GroupName3>Док-станции, порт-репликаторы, стенды</GroupName3>
        <GroupCode1>N1</GroupCode1>
        <GroupCode2>N101</GroupCode2>
        <GroupCode3>N10108</GroupCode3>
        <IsBundle>0</IsBundle>
    </item>
    <item>
        <No>552518</No>
        <Name>Док-станция HP 2010 230W Advanced Docking Station
(NZ223AA) </Name>
        <Brand>HP</Brand>
        <Vendor_part>NZ223AA</Vendor_part>
        <Size>Части</Size>
        <EOL>1</EOL>
        <Warranty>12</Warranty>
        <Weight>3.9</Weight>
        <Volume>0.01523</Volume>
        <Min_Packaged>0</Min_Packaged>
        <GroupName1>НОУТБУКИ</GroupName1>
        <GroupName2>Аксессуары и опции для ноутбуков</GroupName2>
        <GroupName3>Док-станции, порт-репликаторы, стенды</GroupName3>
        <GroupCode1>N1</GroupCode1>
        <GroupCode2>N101</GroupCode2>
        <GroupCode3>N10108</GroupCode3>
        <IsBundle>0</IsBundle>
    </item>
    <item>
        <No>552524</No>
        <Name>Адаптер HP 65W Smart Travel AC Adapter (AU155AA) </Name>
        <Brand>HP</Brand>
        <Vendor_part>AU155AA</Vendor_part>
        <Size>Части</Size>
        <EOL>0</EOL>
        <Warranty>12</Warranty>
        <Weight>0.825</Weight>
        <Volume>0.00369</Volume>
        <Min_Packaged>0</Min_Packaged>
        <GroupName1>НОУТБУКИ</GroupName1>
        <GroupName2>Аксессуары и опции для ноутбуков</GroupName2>
        <GroupName3>Адаптеры и блоки питания</GroupName3>
        <GroupCode1>N1</GroupCode1>
        <GroupCode2>N101</GroupCode2>
        <GroupCode3>N10100</GroupCode3>
        <IsBundle>0</IsBundle>
    </item>
</getItemsResult>
</ns1:getItemsResponse>
</env:Body>
</env:Envelope>

```

6 ВЕРСИИ API

6.1 API 1.0

Первая версия сервиса (**API 1.0**) не поддерживается с июня 2013.

Рекомендуем использовать вторую версию веб-сервиса!

API 1.0 доступен по ссылкам: (используется стиль rpc/encoded)

- <https://api-iz.merlion.ru/mlservice.php?wsdl>
- <https://api-iz.merlion.ru/v1/mlservice.php?wsdl>

6.2 API 2.0

Актуальная версия веб-сервиса.

WSDL-описание сервиса API 2.0 (в различных стилях WSDL) доступно по ссылкам:

- <https://api.merlion.com/dl/mlservice2?wsdl> (стиль document/literal)
- <https://api.merlion.com/re/mlservice2?wsdl> (стиль rpc/encoded)
- <https://api.merlion.com/rl/mlservice2?wsdl> (стиль rpc/literal)

Техническое описание API 2.0 можно скачать в разделе «Помощь» в личном кабинете торговой системы MERLION B2B (<https://b2b.merlion.com/>) по ссылке:

- https://b2b.merlion.com/?action=Y4A20502&action1=Y914A037&chapter=api_func

6.3 РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ В API 2.0

Список всех расширений на данный момент, доступных в API 2.0 (по сравнению с API 1.0):

Дата	Метод	Действие	Описание
17.07.2013	WSDL-описание API	<ul style="list-style-type: none"> ▪ https://api-iz.merlion.ru/v2/mlservice.php?wsdl=re (стиль rpc/encoded) ▪ https://api-iz.merlion.ru/v2/mlservice.php?wsdl=dl (стиль document/literal) ▪ https://api-iz.merlion.ru/v2/mlservice.php?wsdl=rl (стиль rpc/literal) 	API 2.0 позволяет использовать WSDL-описание сервиса в любом <u>стиле WSDL</u>
24.07.2013	getItems() (Справочник товаров)	Добавлен входной параметр: <ul style="list-style-type: none"> ▪ Item_id (код товара) 	Справочник товаров теперь можно запрашивать отдельно по каждому товару, а не по всей товарной группе
24.07.2013	getOrdersList() (Список заказов)	Возвращает дополнительные данные: <ul style="list-style-type: none"> ▪ WillDeleteTomorrow (пометка заказа к удалению); ▪ Status (статус заказа) 	В список неучтенных заказов добавлена информация о пометке заказа к удалению, а также статус самого заказа
01.08.2013	getItemsAvail() (Получение доступного количества товаров и цен)	Возвращает дополнительные данные: <ul style="list-style-type: none"> ▪ AvailableClient_RG (доступное количество товара на региональном складе); ▪ AvailableClient_MSK (доступное количество товара на московском складе) 	Получение остатков доступно на разных складах (региональном и московском), а не только по указанному в настройках партнера складу
21.08.2013	getItems() (Справочник товаров)	Добавлен входной параметр: <ul style="list-style-type: none"> ▪ shipment_method (код метода отгрузки (из справочника getShipmentMethods())) 	Добавлена возможность получать справочник товаров по разным настройкам метода отгрузки. По умолчанию информация будет выводиться, как и раньше, по московскому складу. Также это может повлиять на минимальное количество товара в упаковке – эта настройка различна для складов

01.10.2013	setAddOrderLineCommand() (Увеличение/уменьшение желаемого количества товара в заказе)	Добавлен новый метод	Добавлен новый метод для изменения количества товара в заказе на желаемое число. Новая функция будет полезна для изменения количества товара в одном заказе несколькими менеджерами, которые не связаны друг с другом и не знают общего текущего количества товара в заказе. Например, компания имеет несколько филиалов. В течение дня менеджеры каждого филиала формируют общий заказ, добавляя в него товары. Менеджеры могут добавлять нужное количество товара, не запрашивая текущее количество. Это позволит более оперативно формировать заказ, а также исключить ошибки в количестве резерва
01.10.2013	setMoveOrderLineCommand() (Перенос резервов в новый заказ)	Добавлен новый метод	Добавлен новый метод для переноса резервов в новый заказ. Новая функция будет особенно полезна в Интернет-торговле. Например, в течение дня партнер получает несколько заказов от своих клиентов (из своей торговой системы). Если клиент партнера подтвердил свой заказ, резерв остается в общем заказе. Если клиент не подтвердил заказ, но срок подтверждения еще не истек (например, клиент должен подтвердить свой заказ в течение 3-х дней) и партнер хочет оставить этот товар в резерве, он может перенести его в новый заказ (с сохранением резерва) до выяснения, будет ли клиент оплачивать и забирать свой товар
01.10.2013	getCommandResult() (Результат обработки команд)	Возвращает дополнительные данные: ▪ DocumentNo2 (номер нового документа)	При проверке результата обработки команд добавлена информация о номере нового документа (при переносе резервов в новый заказ с помощью метода setMoveOrderLineCommand() в этом поле выводится номер нового созданного заказа)
01.10.2013	getCommandResult() (Результат обработки команд)	Добавлена возможность выполнения команды без указания параметра: ▪ operation_no (номер команды)	Добавлена возможность просмотра последних 20 команд, не указывая номер операции
01.10.2013	getCommandResult() (Результат обработки команд)	Возвращает дополнительные данные: ▪ OperationLineNo (номер строки в пачке команд)	При проверке результата обработки команд добавлена информация о номере строки в пачке команд (пачка – это набор команд, выполняемых в одной транзакции)

15.11.2013	setSignOrderCommand() (Команда для подписи заказа)	Добавлен новый метод	Добавлен новый метод для подписи заказа на стороне клиента. Новая функция поможет сократить время для отгрузки сформированного заказа. Подписать заказ можно следующими способами: <ul style="list-style-type: none"> подписать клиентом (редактирование заказа закончено, менеджер компании MERLION может взять его в работу); утвердить заказ (за менеджера компании MERLION); подписать заказ на отгрузку (за менеджера и финансового менеджера компании MERLION); поставить все подписи. Уточнить, какие типы подписи доступны, можно у Вашего менеджера в компании MERLION.
15.11.2013	setOrderHeaderCommand() (Создание/редактирование заголовка заказа)	Добавлены входные параметры: <ul style="list-style-type: none"> representative – представитель (из справочника getRepresentative()); endpoint_delivery_id – код конечной точки доставки (из справочника getEndPointDelivery()); packing_type – код типа упаковки (из справочника getPackingTypes()) 	В связи с добавлением нового метода для подписи заказа на стороне клиента, добавлены входные параметры для создания/редактирования заголовка заказа (для подписи заказа новые параметры обязательно должны быть заполнены)
15.11.2013	getRepresentative() (Справочник представителей по основному договору контрагента)	Добавлен новый метод	В связи с добавлением новых входных параметров для создания/редактирования заголовка заказа, добавлен новый справочник
15.11.2013	getEndPointDelivery() (Справочник конечных точек доставки)	Добавлен новый метод	В связи с добавлением новых входных параметров для создания/редактирования заголовка заказа, добавлен новый справочник
15.11.2013	getPackingTypes() (Справочник типов упаковки)	Добавлен новый метод	В связи с добавлением новых входных параметров для создания/редактирования заголовка заказа, добавлен новый справочник
15.11.2013	getOrdersList() (Список заказов)	Возвращает дополнительные данные: <ul style="list-style-type: none"> EndPointCity (город доставки); EndPointAdress (адрес доставки); EndPointContact (контактное лицо); PackingType (тип упаковки); Representative (представитель) 	В связи с добавлением нового метода для подписи заказа на стороне клиента, в список заголовков неучтенных заказов добавлена информация о городе, адресе доставки, контактном лице, типе упаковки и представителе

26.12.2013	WSDL-описание API	<ul style="list-style-type: none"> ▪ https://api-iz.merlion.ru/dl/mlservice2?wsdl (стиль document/literal) ▪ https://api-iz.merlion.ru/re/mlservice2?wsdl (стиль rpc/encoded) ▪ https://api-iz.merlion.ru/rl/mlservice2?wsdl (стиль rpc/literal) 	WSDL-описание сервиса (с различными стилями WSDL) доступно по новым ссылкам. Ссылки на WSDL-описание сервиса API 2.0, указанные в предыдущих версиях документа, полностью поддерживаемые
24.03.2014	getOrdersList() (Список заказов)		Добавлена возможность вывода информации по отгруженным заказам. Если в метод getOrdersList() (Список заказов) передан номер документа по отгруженному заказу, будет выведена информация по этому заказу. Если заказ отгружался по нескольким документам, будет выведена информация по заголовкам каждого учтенного документа; в таком случае номера учтенных документов могут содержать: «номер заказа» + «две заглавные буквы i» + «код логистического объекта».
24.03.2014	getOrderLines() (Список строк заказа)	Добавлен входной параметр: <ul style="list-style-type: none"> ▪ details (детализация заказа) 	Добавлена возможность вывода детализации строк заказа. Возможные значения параметра details (детализация заказа): <ul style="list-style-type: none"> ▪ "0" – без детализации (по умолчанию); ▪ "1" – с детализацией (с разбивкой по отгрузочным документам); ▪ "2" – с детализацией только по неучтенным документам; ▪ "3" – с детализацией только по учтенным документам. Если заказ отгружался по нескольким документам, будет выведена информация по строкам каждого учтенного документа; в таком случае номера учтенных документов могут содержать: «номер заказа» + «две заглавные буквы i» + «код логистического объекта»
24.03.2014	getOrderLinesGTD() (Список строк заказа с ГТД)	Добавлен новый метод	Добавлена возможность получения строк заказа с грузовой таможенной декларацией (ГТД) (по отгруженным заказам)
24.03.2014	getCountry() (Справочник стран)	Добавлен новый метод	В связи с добавлением нового метода по получению строк заказа с ГТД, добавлен новый справочник

24.03.2014	getItems() (Справочник товаров)	Добавлены входные параметры: ▪ page (номер страницы; по умолчанию = 0, постраничный вывод выключен); ▪ rows_on_page (количество записей на странице; по умолчанию 10 000 строк)	Добавлена возможность постраничного вывода данных для получения справочника товаров, а также характеристик и изображений товаров. Постраничный вывод позволяет получать данные для запросов по достаточно большим товарным подгруппам без превышения времени выполнения
24.03.2014	getItemsProperties() (Характеристики товаров)	Добавлены входные параметры: ▪ page (номер страницы; по умолчанию = 0, постраничный вывод выключен); ▪ rows_on_page (количество записей на странице; по умолчанию 10 000 строк)	Добавлена возможность постраничного вывода данных для получения справочника товаров, а также характеристик и изображений товаров. Постраничный вывод позволяет получать данные для запросов по достаточно большим товарным подгруппам без превышения времени выполнения
24.03.2014	getItemsImages() (Изображения товаров)	Добавлены входные параметры: ▪ page (номер страницы; по умолчанию = 0, постраничный вывод выключен); ▪ rows_on_page (количество записей на странице; по умолчанию 10 000 строк)	Добавлена возможность постраничного вывода данных для получения справочника товаров, а также характеристик и изображений товаров. Постраничный вывод позволяет получать данные для запросов по достаточно большим товарным подгруппам без превышения времени выполнения
24.03.2014	getItemsProperties() (Характеристики товаров)	Добавлен входной параметр: ▪ last_days_changed (изменения за n последних дней; по умолчанию = 0 (все характеристики))	Добавлена возможность вывода изменений по характеристикам товаров за последние n дней
24.03.2014	getItems() (Справочник товаров)	Возвращает дополнительные данные: ▪ IsBundle (признак бандла)	В справочник товаров добавлена информация о признаке бандла (является ли товар комплектом)
24.03.2014	getRepresentative() (Справочник представителей по основному договору контрагента)	Возвращает дополнительные данные: ▪ StartDate (дата начала срока действия); ▪ EndDate (дата окончания срока действия)	При получении представителей по основному договору контрагента добавлена информация о сроке действия представителя
24.03.2014	WSDL-описание API	▪ https://api.merlion.com/dl/mlservice2?wsdl (стиль document/literal) ▪ https://api.merlion.com/re/mlservice2?wsdl (стиль rpc/encoded) ▪ https://api.merlion.com/rl/mlservice2?wsdl (стиль rpc/literal)	WSDL-описание сервиса (с различными стилями WSDL) доступно по новым ссылкам. Ссылки на WSDL-описание сервиса API 2.0, указанные в предыдущих версиях документа, полностью поддерживаемые