



## **Инструкция**

**по использованию Marvel-B2B API**

## Содержание

Введение .....	5
1. Секретный ключ (secretKey).....	5
2. Структура возвращаемых данных.....	5
2.1 Структура данных XML .....	5
2.2 Пример структуры данных XML с сообщением об ошибке .....	6
2.3 Структура данных JSON .....	6
2.4 Пример структуры данных JSON с сообщением об ошибке .....	6
3. Методы API по работе с товарным каталогом.....	7
3.1 Метод GetCatalogCategories .....	7
3.1.1 Вызов метода .....	7
3.1.2 Возвращаемые данные .....	8
3.2 Метод GetFullStock .....	9
3.2.1 Вызов метода .....	9
3.2.2 Возвращаемые данные.....	10
3.3 Метод GetStock .....	12
3.3.1 Вызов метода .....	12
3.3.2 Возвращаемые данные.....	13
3.4 Метод GetItems.....	13
3.4.1 Вызов метода .....	13
3.4.2 Возвращаемые данные.....	14
3.5 Метод GetItemPhotos .....	16
3.5.1 Вызов метода .....	16
3.5.2 Возвращаемые данные.....	16
3.6 Ограничения на выгрузку данных.....	18
4. Методы API для управления резервированием товаров .....	18
4.1 Перечень методов.....	18

4.2	Метод CheckReserveStatus .....	18
4.2.1	Вызов метода .....	18
4.2.2	Возвращаемые данные .....	19
4.3	Метод CreateReserve .....	21
4.3.1	Вызов метода .....	21
4.3.2	Возвращаемые данные .....	23
4.4	Метод ModifyReserve .....	25
4.5	Метод DeleteReserve .....	25
4.5.1	Вызов метода .....	25
4.5.2	Возвращаемые данные .....	26
4.6	Метод TransferReserve .....	28
4.6.1	Вызов метода .....	28
4.6.2	Возвращаемые данные .....	29
4.6.3	Ошибки метода TransferReserve .....	30
5.	Методы API для работы с заказами .....	31
5.1	Метод CreateSalesOrder .....	31
5.1.1	Создание заказа без предварительного резервирования .....	32
5.1.2	Создание заказа из существующего резерва .....	33
5.1.3	Создание комбинированного заказа .....	34
5.1.4	Возвращаемые данные .....	34
5.2	Метод CheckSalesOrderStatus .....	37
5.2.1	Тэг-параметр <SalesOrderShipmentStatus/> requestParams метода CheckSalesOrderStatus .....	37
5.2.2	Тэг-параметр <SalesOrderId/> requestParams метода CheckSalesOrderStatus .....	37
5.2.3	Тэг-параметр <ExternalSalesOrderId/> requestParams метода CheckSalesOrderStatus .....	38
5.2.4	Возвращаемые данные .....	38
5.3	Метод CheckSalesOrderItemStatus .....	39
5.3.1	Тэг-параметр <SalesOrderShipmentStatus/> requestParams метода CheckSalesOrderItemStatus .....	40

5.3.2	Тэг-параметр <SalesOrderId/> requestParams метода CheckSalesOrderItemStatus .....	40
5.3.3	Тэг-параметр <ExternalSalesOrderId/> requestParams метода CheckSalesOrderItemStatus .....	40
5.3.4	Тэг-параметр <ItemId/> requestParams метода CheckSalesOrderItemStatus.....	40
5.3.5	Возвращаемые данные.....	40
6.	Метод ModifySalesOrder .....	42
6.1	Возвращаемые данные.....	43
7.	Метод ModifySalesOrderLines .....	43
7.1	Возвращаемые данные.....	45
8.	Метод GetShipmentData.....	47
8.1	Возвращаемые данные.....	47
9.	Метод GetOpenSending .....	48
9.1	Возвращаемые данные.....	48
10.	Метод ShipmentSalesOrder .....	48
10.1	Возвращаемые данные.....	49
11.	Внешние артикулы .....	50
11.1	Расширенная информация, возвращаемая методами B2B API при использовании внешних артикулов .....	50
11.1.1	Тэг <ExternalItemId /> .....	50
11.1.2	Тэги <ExternalItemIdError /> и <ExternalItemIdErrorCode /> .....	51
11.2	Использование внешних артикулов при обращении к методам API.....	53
11.3	Сообщения об ошибках, возникающие при использовании внешних артикулов .....	53
12.	Метод API Logout .....	54
13.	Сценарий использования данных, возвращаемых B2B API.....	54
14.	Тестовые вызовы методов API .....	55
14.1	Промежуточный прокси-сервер.....	55
14.2	Использование плагина Restlet Client – REST API Testing .....	55

## Введение

Марвел-B2B API (далее **API**) представляет собой набор методов (URL), позволяющих после авторизации получить доступ к информационным ресурсам Marvel B2B (далее **B2B**), в том числе – состояние склада, цены, резервы.

Для доступа к методам API необходимо получить специальный аккаунт B2B. Доступ в B2B через браузер с такого аккаунта невозможен.

Методы API доступны по следующему адресу: <https://b2b.marvel.ru/Api/>

API не является WSDL-сервисом.

## 1. Секретный ключ (secretKey)

Большинство методов API имеют параметр secretKey. Это **устаревший параметр**, который оставлен для совместимости с уже разработанным кодом пользователей API. Этот параметр можно игнорировать и не использовать при вызове методов API.

В ответах методов API присутствует поле (тэг) <Key></Key>. Это поле оставлено для совместимости с уже разработанным кодом пользователей API. Поле всегда пусто и может быть проигнорировано.

## 2. Структура возвращаемых данных

API возвращает данные в форматах XML или JSON. Пользователь выбирает формат возвращаемых данных при вызове функции API.

### 2.1 Структура данных XML

```
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
    <Key></Key>
  </Header>
  <Body>
    Структура данных XML
  </Body>
</Response>
```

Корневой тэг **<Response>** содержит тэги **<Header>** и **<Body>**.

В **<Header>** содержатся три служебных тэга:

**<Code>** – содержит 0, если запрос завершился успешно, и 1, во всех иных ситуациях.

**<Message>** – содержит сообщение об ошибке, если по ходу запроса произошла ошибка.

<Key> – устаревший тэг, оставлен для совместимости, всегда пустой.

<Body> содержит данные по структуре каталога или запрошенной категории (если запрос завершился удачно), либо ничего (если запрос завершился неудачно).

Подробнее о структуре данных см. ниже в разделе «[Сценарий использования данных, возвращаемых B2B API](#)».

## 2.2 Пример структуры данных XML с сообщением об ошибке

```
<Response>
  <Header>
    <Code>1</Code>
    <Message>Вы сможете сгрузить запрошенные данные через 12 мин.</Message>
    <Key></Key>
  </Header>
  <Body />
</Response>
```

Подобное сообщение говорит о том, что лимит по времени на выгрузку запрошенных данных ещё не завершился, и в ближайшие 12 минут выгрузить запрошенные данные не удастся.

## 2.3 Структура данных JSON

```
{
  "Header": {
    "Code": 0,
    "Message": "",
    "Key": ""
  },
  "Body": {
    Структура запрошенных данных
  }
}
```

Поля JSON имеют такое же назначение, как и поля XML.

## 2.4 Пример структуры данных JSON с сообщением об ошибке

```
{
  "Header": {
    "Code": 1,
    "Message": "Вы сможете сгрузить запрошенные данные через 7 мин.",
    "Key": ""
  },
  "Body": null
}
```

### 3. Методы API по работе с товарным каталогом

Для выгрузки информации о товарах на складе и ценах следует использовать следующие методы API: **GetCatalogCategories**, **GetFullStock**, **GetStock** и **GetItems**. Вызовы этих методов и структура возвращаемых данных рассматриваются ниже.

В общем случае методы API вызываются с различным набором параметров. Но есть общие для всех методов правила:

1. Обращение к методу API является POST-запросом.
2. Все параметры имеют строковый тип данных.
3. Первые три параметра у всех методов одинаковые. Это параметры `user`, `password`, `secretKey`(устаревший параметр, можно игнорировать).
4. Методы API можно вызывать в любой последовательности.
5. При вызове в параметре `secretKey` следует передать пустую строку.
6. Значение тэга `<Key>` в ответе сервера **всегда пустое**, этот тэг следует игнорировать.
7. Параметр `responseFormat` – определяет формат данных, возвращаемых сервером (0 – XML, 1 – JSON).

#### 3.1 Метод **GetCatalogCategories**

##### 3.1.1 Вызов метода

Метод **GetCatalogCategories** позволяет выгрузить структуру каталога.

Для обращения к методу следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/GetCatalogCategories?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0>

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **secretKey** – устаревший параметр. Передавать в него пустую строку или игнорировать.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.

### 3.1.2 Возвращаемые данные

В случае успешного запроса API возвращает структуру каталога в тэге **<Body>**, где содержится последовательность тэгов **<Categories>** (пример 1). Каждый тэг **<Categories>** содержит в себе информацию о категории, о входящих в категорию подкатегориях и о вендорах.

```
<Response>
  <Header>
    <Code>0</Code>
    <Message/>
    <Key></Key>
  </Header>
  <Body>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>...</Categories>
  </Body>
</Response>
```

пример 1

Например, категория «Персональные компьютеры» выглядит следующим образом (пример 2):

```
<Response>
  <Header>...</Header>
  <Body>
    <Categories>...</Categories>
    <Categories>...</Categories>
    <Categories>
      <CategoryID>ПК</CategoryID>
      <CategoryName>Персональные компьютеры</CategoryName>
      <ParentCategoryId/>
      <SubCategories>...</SubCategories>
      <SubCategories>...</SubCategories>
      <SubCategories>...</SubCategories>
      <SubCategories>...</SubCategories>
      <SubCategories>...</SubCategories>
      <Vendors>...</Vendors>
      <Vendors>...</Vendors>
      <Vendors>...</Vendors>
      <Vendors>...</Vendors>
      <Vendors>...</Vendors>
      <Vendors>...</Vendors>
      <CategoryTreeId>61</CategoryTreeId>
    </Categories>
    <Categories>...</Categories>
  </Body>
</Response>
```

пример 2



Тэг **<CategoryID>** содержит уникальный идентификатор категории.

Тэг **<CategoryName>** содержит название категории.

Тэг **<ParentCategoryId>** содержит идентификатор категории, для которой текущая категория является подкатегорией. Если категория корневая, то пусто (null).

0 или больше тэгов **<Subcategories>** перечисляют подкатегории, входящие в категорию, в нашем примере – в категорию «Персональные компьютеры».

0 или больше тэгов **<Vendors>** перечисляют производителей артикулов, входящих в категорию. Каждый тэг **<Vendors>** содержит тэг **<WareProducerCode>** (уникальный идентификатор производителя) и тэг **<WareProducerName>** (название производителя).

Тэг **<CategoryTreeld>** идентификатор категории в дереве каталога.

Глубина иерархии может быть произвольной, но на практике не превышает 5 уровней.

## 3.2 Метод **GetFullStock**

### 3.2.1 Вызов метода

Метод **GetFullStock** позволяет выгрузить содержимое всего каталога. Во избежание дублирования артикулов, привязанных к нескольким категориям, выгружаются данные из категории с наибольшим «весом».

Для обращения к методу следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/GetFullStock?user=usrLogin&password=usrPassword  
&secretKey=&packStatus=0&responseFormat=0&instock=0](https://b2b.marvel.ru/Api/GetFullStock?user=usrLogin&password=usrPassword&secretKey=&packStatus=0&responseFormat=0&instock=0)

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **secretKey** – устаревший параметр. Передавать в него пустую строку или игнорировать.
- **packStatus** – состояние упаковки выгружаемых артикулов. 0 – любая упаковка, 1 – только хорошая упаковка, 2 – только некондиция. Например, при значении 2 в выгрузку по категории попадут артикулы, у которых упаковка: МЯТАЯ, из ремонта, некомплект, с теста, вскрытая.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **inStock** – в наличии на складе, необязательный параметр. 0 – есть на складе (будут выгружены артикулы, которые есть на складе), 1 – доступны для резерва (артикулы, которые есть на складе и могут быть зарезервированы), 2 – весь товар (все артикулы, которые есть на складе, и артикулы, которых нет в наличии). По-умолчанию передается 0.

### 3.2.2 Возвращаемые данные

В случае успешного запроса API возвращает содержимое всего каталога в тэге **<Body>** (пример 3). Внутри тэга **<Body>** много тэгов **<CategoryItem>**, каждый из которых описывает артикул.

```
<Response>
  <Header>...</Header>
  <Body>
    <CategoryItem>...</CategoryItem>
    <CategoryItem>...</CategoryItem>
    <CategoryItem>
      <AvailableForShippingInMSKCount>1</AvailableForShippingInMSKCount>
      <AvailableForShippingInSPBCount>0</AvailableForShippingInSPBCount>
      <WarePriceRUB>50,000000</WarePriceRUB>
      <WareArticle>4603720423065</WareArticle>
      <WarePackStatus>OK</WarePackStatus>
      <Dimension>осн</Dimension>
      <WareVendor>Accesstyle</WareVendor>
      <WarePrice>50,000000</WarePrice>
      <WarePriceCurrency>RUR</WarePriceCurrency>
      <RRPrice />
      <TotalInventQty>61</TotalInventQty>
      <AvailableForB2BOrderQty>1</AvailableForB2BOrderQty>
      <InNearTransitCount>0</InNearTransitCount>
      <InFarTransitCount>0</InFarTransitCount>
      <NetWeight />
      <Weight />
      <UnitVolume>0.000162656000</UnitVolume>
      <Width>46.000000000000</Width>
      <Height>26.000000000000</Height>
      <Depth>34.000000000000</Depth>
      <TaxPackagingCount>250</TaxPackagingCount>
      <PurchaseQty />
      <WareFullName>Пленка защитная Accesstyle Защитная пленка
AccesStyle для Samsung Galaxy Tab3 10.1", глянцевая</WareFullName>
      <WarePriceUSD>0,803928</WarePriceUSD>
      <CanBeOrdered>true</CanBeOrdered>
      <APIReservedQty>0</APIReservedQty>
      <APIAvailableReservedQty>0</APIAvailableReservedQty>
      <EANUPC>
        <EANUPCCode json:Array="true">4603720423065</EANUPCCode>
      </EANUPC>
      <CategoryId>Нойт_ОпцПл_ЗащПл</CategoryId>
      <CategoryName>Защитные пленки для планшетов</CategoryName>
      <PromoDescription />
      <PromoURL />
    </CategoryItem>
    <CategoryItem>...</CategoryItem>
    <CategoryItem>...</CategoryItem>
    <CategoryItem>...</CategoryItem>
  </Body>
</Response>
```

пример 3

Тэги, описывающие артикул:

<**AvailableForShippingInMSKCount**>: количество, имеющиеся на складах в Москве, доступность отгрузки которых можно выяснить через менеджера. Положительное число, ноль или «» (пустая строка).

<**AvailableForShippingInSPBCount**>: количество, имеющиеся на складах в Санкт-Петербурге, доступность отгрузки которых можно выяснить через менеджера. Положительное число, ноль или «» (пустая строка).

<**WarePriceRUB**>: цена прайс-листа, пересчитанная в рубли по текущему курсу ЦБ РФ.

<**WareArticle**>: идентификатор артикула. Строка ненулевой длины.

<**WarePackStatus**>: состояние упаковки. Возможные значения: ОК, мятая, из ремонта, некомплект, с теста, вскрытая.

<**Dimension**>: локация склада. Возможные значения: осн, net.

<**WareVendor**>: производитель артикула. Строка ненулевой длины или «» (пустая строка).

<**WarePrice**>: стоимость единицы артикула. Число большее нуля или «» (пустая строка).

<**WarePriceCurrency**>: валюта стоимости. Возможные значения: RUR, USD, EUR или «» (пустая строка).

<**RRPrice />**: рекомендованная розничная цена. Число больше нуля или пусто.

<**TotalInventQty**>: общее количество товара на складах Марвел. Ноль или положительное число не больше 100. Если значение больше 100, то выводится «100+».

<**AvailableForB2BOrderQty**>: свободное количество, доступное для отгрузки или резервирования в B2B. Ноль или положительное число не больше 100. Если значение больше 100, то выводится «100+».

<**InNearTransitCount**>: товар в транзите, поступит на склады Марвел в течение 1-7 дней. Ноль или положительное число не больше 100. Если значение больше 100, то выводится «100+».

<**InFarTransitCount**>: товар в транзите, поступит на склады Марвел в течение 8-30 дней. Ноль или положительное число не больше 100. Если значение больше 100, то выводится «100+».

<**NetWeight**>: вес нетто в килограммах. Число большее нуля или «» (пустая строка).

<**Weight**>: вес брутто в килограммах. Число большее нуля или «» (пустая строка).

<**UnitVolume**>: объем в кубических метрах. Число большее нуля или «» (пустая строка).

<**Width**>: ширина упаковки в сантиметрах. Число большее нуля или «» (пустая строка).

<**Height**>: высота упаковки в сантиметрах. Число большее нуля или «» (пустая строка).

<**Depth**>: глубина упаковки в сантиметрах. Число большее нуля или «» (пустая строка).

<**TaxPackagingCount**>: количество артикулов в упаковке. Число большее нуля или «» (пустая строка).

<**PurchaseQty**>: количество артикулов для единовременного заказа. Число большее нуля или «» (пустая строка).

<**WareFullName**>: полное наименование артикула.

<**WarePriceUSD**>: цена прайс-листа, пересчитанная в USD по текущему курсу ЦБ РФ.

<**CanBeOrdered**>: артикул может быть автоматически заказан/зарезервирован. Возможные значения:

- true – доступен для заказа/резерва;
- false – не доступен для заказа/резерва.

<**APIReservedQty**>: содержит суммарное зарезервированное количество данного артикула в пуле резерва пользователя.

<**APIAvailableReservedQty**>: содержит суммарное зарезервированное количество данного артикула в пуле резерва пользователя, не выделенное под внешний идентификатор.

<**CategoryId**>: уникальный идентификатор категории, к которой привязан артикул.

<**CategoryName**>: название категории, к которой привязан артикул.

<**PromoDescription**>: если по артикулу есть промо-акция, то содержит краткое описание акции.

<**PromoURL**>: если по артикулу есть промо-акция, то содержит ссылку (URL) на сайт и страницу с описанием акции.

### 3.3 Метод **GetStock**

#### 3.3.1 Вызов метода

Метод **GetStock** позволяет выгрузить содержимое категории.

Для обращения к методу следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/GetStock?user=usrLogin&password=usrPassword&secretKey=&categoryId=Apple&packStatus=0&responseFormat=0&includeSubCategories=1&inStock=0>

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **secretKey** – устаревший параметр. Передавать в него пустую строку или игнорировать.
- **categoryId** – идентификатор категории для выгрузки, например «ПК», «Ноут».
- **packStatus** – состояние упаковки выгружаемых артикулов. 0 – любая упаковка, 1 – только **хорошая** упаковка, 2 – только некондиция. Например, при значении 2 в выгрузку по категории попадут артикулы, у которых упаковка: мятая, из ремонта, некомплект, с теста, вскрытая.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **includeSubCategories** – включать в результат выгрузки все подкатегории. 0 – нет, 1 – да.

- **inStock** – в наличии на складе, необязательный параметр. 0 – есть на складе (будут выгружены артикулы, которые есть на складе), 1 – доступны для резерва (артикулы, которые есть на складе и могут быть зарезервированы), 2 – весь товар (все артикулы, которые есть на складе, и артикулы, которых нет в наличии). По-умолчанию передается 0.

### 3.3.2 Возвращаемые данные

Возвращаемые данные метода **GetStock** идентичны возвращаемым данным метода **GetFullStock**.

## 3.4 Метод GetItems

### 3.4.1 Вызов метода

Метод **GetItems** выполняет поиск артикулов в каталоге на предмет точного совпадения с указанными в запросе артикулами.

Для обращения к методу следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/GetItems?user=usrLogin&password=usrPassword&secretKey=&packStatus=0&responseFormat=0&items=<Root>>

```
<WareItem>
  <ItemId>PB515-D-RU</ItemId>
</WareItem>
<WareItem>
  <ItemId>A18-7570</ItemId>
</WareItem>
<WareItem>
  <ItemId>SEB-100</ItemId>
</WareItem>
<WareItem>
  <ItemId>E2243Fw2</ItemId>
</WareItem>
</Root>
```

Пример 4

или

<https://b2b.marvel.ru/Api/GetItems?user=usrLogin&password=usrPassword&secretKey=&packStatus=0&responseFormat=0&items={>

```
"WareItem": [
  {"ItemId": "PB515-D-RU"},
  {"ItemId": "A18-7570"},
  {"ItemId": "SEB-100"},
  {"ItemId": "E2243Fw2"}
]
}
```

Пример 5

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **secretKey** – устаревший параметр. Передавать в него пустую строку или игнорировать.
- **packStatus** – состояние упаковки выгружаемых артикулов. 0 – любая упаковка, 1 – только хорошая упаковка, 2 – только некондиция. Например, при значении 2 в выгрузку по категории попадут артикулы, у которых упаковка: мятая, из ремонта, некомплект, с теста, вскрытая.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **getExtendedItemInfo** – если 1, то для каждого артикула выгружается информация о дополнительных характеристиках. Если 0, то дополнительная информация отсутствует.
- **Items** – строка, содержащая список артикулов в формате XML (*пример 4*) или JSON (*пример 5*). В одном запросе будет обработан список, не превышающий 1000 артикулов.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **Items** с помощью URLEncode.

### 3.4.2 Возвращаемые данные

Возвращаемые данные метода **GetItems** расширены на один тэг по сравнению с возвращаемыми данными метода **GetFullStock** при условии, что **getExtendedItemInfo=0**.

Этот дополнительный тэг `<EANUPC/>`. Тэг содержит ноль или много тэгов `<EANUPCCode>` которые содержат EAN/UPC коды, ассоциированные с артикулом.

Если **getExtendedItemInfo=1**, то внутри каждого тэга `<CategoryItem>` добавляется тэг `<ExtendedInfo>` (*пример 6*).

Тэг `<EANUPC/>` выводится всегда, вне зависимости от значения параметра **getExtendedItemInfo**.

```
<CategoryItem>
  <AvailableForShippingInMSKCount>0</AvailableForShippingInMSKCount>
  <AvailableForShippingInSPBCount>0</AvailableForShippingInSPBCount>
  <WarePriceRUB>6859,776320</WarePriceRUB>
  <WareArticle>BE550G-RS</WareArticle>
  <WarePackStatus>OK</WarePackStatus>
  <Dimension>net</Dimension>
  <WareVendor>APC</WareVendor>
  <WarePrice>109,600000</WarePrice>
  <WarePriceCurrency>USD</WarePriceCurrency>
  <RRPrice />
  <TotalInventQty>0</TotalInventQty>
  <AvailableForB2BOrderQty>0</AvailableForB2BOrderQty>
  <InNearTransitCount>0</InNearTransitCount>
  <InFarTransitCount>0</InFarTransitCount>
  <NetWeight>7.369219721462</NetWeight>
```

```

<Weight>7.369219721462</Weight>
<UnitVolume>0.015984000000</UnitVolume>
<Width>27.000000000000</Width>
<Height>32.000000000000</Height>
<Depth>37.000000000000</Depth>
<TaxPackagingCount>2</TaxPackagingCount>
<PurchaseQty />
550VA / 330W, Tower, Schuko, USB<WareFullName>Источник бесперебойного питания APC Back-UPS ES, OffLine,
<WarePriceUSD>109,600000</WarePriceUSD>
<CanBeOrdered>>false</CanBeOrdered>
<APIReservedQty>0</APIReservedQty>
<APIAvailableReservedQty>0</APIAvailableReservedQty>
<EANUPC>
  <EANUPCCode>731304271697</EANUPCCode>
  <EANUPCCode>10731304271694</EANUPCCode>
  <EANUPCCode>107313042786</EANUPCCode>
</EANUPC>
<CategoryId>Сбесп_Back UPS</CategoryId>
<CategoryName>Back-UPS</CategoryName>
<PromoDescription />
<PromoURL />
<ExtendedInfo>
  <Parameter>
    <ParameterName>Серия вендора</ParameterName>
    <ParameterValue>Back-UPS ES</ParameterValue>
  </Parameter>
  <Parameter>
    <ParameterName>Топология</ParameterName>
    <ParameterValue>OffLine</ParameterValue>
  </Parameter>
  <Parameter>
    <ParameterName>Мощность, VA</ParameterName>
    <ParameterValue>550VA</ParameterValue>
  </Parameter>
  <Parameter>
    <ParameterName>Мощность, Вт</ParameterName>
    <ParameterValue>330W</ParameterValue>
  </Parameter>
  <Parameter>
    <ParameterName>Исполнение</ParameterName>
    <ParameterValue>Tower</ParameterValue>
  </Parameter>
  <Parameter>
    <ParameterName>Тип выходных розеток</ParameterName>
    <ParameterValue>Schuko</ParameterValue>
  </Parameter>
  <Parameter>
    <ParameterName>Коммуникационный порт</ParameterName>
    <ParameterValue>USB</ParameterValue>
  </Parameter>
</ExtendedInfo>
</CategoryItem>

```

пример 6

Следует иметь в виду, что набор параметров различен для различных типов артикулов, например, для мониторов и источников бесперебойного питания.

## 3.5 Метод GetItemPhotos

### 3.5.1 Вызов метода

Метод возвращает перечень картинок, привязанных к заданному артикулу. Для вызова метода нужно сформировать URL:

Для обращения к методу следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/GetItemPhotos?user=usrLogin&password=usrPassword&responseFormat=0&items=<Root>>

```
<WareItem>
  <ItemId>PB515-D-RU</ItemId>
</WareItem>
<WareItem>
  <ItemId>A18-7570</ItemId>
</WareItem>
<WareItem>
  <ItemId>SEB-100</ItemId>
</WareItem>
<WareItem>
  <ItemId>E2243Fw2</ItemId>
</WareItem>
</Root>
```

или

<https://b2b.marvel.ru/Api/GetItemPhotos?user=usrLogin&password=usrPassword&responseFormat=0&items={>

```
"WareItem": [
  {"ItemId": "PB515-D-RU"},
  {"ItemId": "A18-7570"},
  {"ItemId": "SEB-100"},
  {"ItemId": "E2243Fw2"}
]
}
```

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **Items** – строка, содержащая список артикулов в формате XML или JSON.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **Items** с помощью URLEncode.

### 3.5.2 Возвращаемые данные

Возвращаемые данные содержат перечисление картинок для каждого артикула из запроса. К артикулу может быть привязано одна или несколько картинок. Если для запрошенного артикула картинок нет, то он отсутствует в возвращаемых данных.



Пример данных XML:

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
    <Key></Key>
  </Header>
  <Body>
    <Photo>
      <BigImage>
        <URL>https://b2b.marvel.ru/AppData/items/NX.EFAER.001/NX.EFAER.001_661860130.png</URL>
      >
        <Size>104575</Size>
        <Width>1000</Width>
        <Height>748</Height>
        <WareArticle>NX.EFAER.001</WareArticle>
      </BigImage>
    </Photo>
    <Photo>
      <BigImage>
        <URL>https://b2b.marvel.ru/AppData/items/MJLQ2RU_A/MJLQ2RU_A_208252339.png</URL>
      >
        <Size>10669</Size>
        <Width>150</Width>
        <Height>150</Height>
        <WareArticle>MJLQ2RU/A</WareArticle>
      </BigImage>
    </Photo>
    <Photo>
      <BigImage>
        <URL>https://b2b.marvel.ru/AppData/items/AP9631/AP9631_168317726.png</URL>
      >
        <Size>7353</Size>
        <Width>150</Width>
        <Height>150</Height>
        <WareArticle>AP9631</WareArticle>
      </BigImage>
    </Photo>
  </Body>
</Response>
```

пример 7

Тэг **<URL/>** – месторасположение картинки.

Тэг **<Size>** – размер картинки в байтах.

Тэг **<Width>** – ширина картинки.

Тэг **<Height>** – высота картинки.

Тэг **<WareArticle>** – артикул, для которого запрашивается картинка.

### 3.6 Ограничения на выгрузку данных

Для того чтобы снизить вероятность перегрузки сервера, введены ограничения на выгрузку больших объёмов данных.

Структуру каталога (метод **GetCatalogCategories**) разрешено выгружать один раз в 10 мин.

Содержимое всего каталога (метод **GetFullStock**) разрешено выгружать один раз в 1 час.

На выгрузку категории с помощью метода **GetStock** действуют следующие ограничения:

- одну и ту же категорию 1-го уровня можно выгрузить один раз в 30 мин;
- одну и ту же категорию 2-го и более уровня можно выгрузить один раз в 15 мин.

Поиск артикулов (метод **GetItems**) можно производить без ограничений по времени между вызовами.

## 4. Методы API для управления резервированием товаров

### 4.1 Перечень методов

- <https://b2b.marvel.ru/Api/CreateReserve>
- <https://b2b.marvel.ru/Api/ModifyReserve>
- <https://b2b.marvel.ru/Api/DeleteReserve>
- <https://b2b.marvel.ru/Api/CheckReserveStatus>
- <https://b2b.marvel.ru/Api/TransferReserve>

Методы API по работе с резервированием товаров (далее по тексту – с резервами) сходны с методами API по выгрузке каталога и категорий. Для вызова метода следует передать в него имя пользователя, пароль пользователя, секретную строку (устаревший параметр, передать пустую строку или игнорировать), ожидаемый формат документа-ответа (XML или JSON).

Кроме того, все методы, кроме **CheckReserveStatus**, принимают на вход строку **Items**, содержащую перечень артикулов, над которыми нужно произвести операцию резервирования. Перечень артикулов может передаваться как в формате JSON, так и в формате XML.

Подробнее входные и выходные документы в обоих форматах рассмотрены ниже.

### 4.2 Метод **CheckReserveStatus**

#### 4.2.1 Вызов метода

Для вызова метода **CheckReserveStatus** следует отправить URL (POST запрос):

<https://b2b.marvel.ru/Api/CheckReserveStatus?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0>

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **secretKey** – устаревший параметр. Передавать в него пустую строку или игнорировать.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.

#### 4.2.2 Возвращаемые данные

Метод **CheckReserveStatus** возвращает информацию обо всех зарезервированных артикулах, общую сумму резерва и лимит резервирования.

Ниже приведен документ, возвращённый методом CheckReserveStatus в формате XML (*пример 8*):

```
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
    <Key></Key>
  </Header>
  <Body>
    <ReserveSum>2039,00</ReserveSum>
    <ReserveSumCurrency>USD</ReserveSumCurrency>
    <ReserveSumLimit>650000,00</ReserveSumLimit>
    <ReserveSumLimitCurrency>USD</ReserveSumLimitCurrency>
    <WareItem>
      <ItemId>NX.MGRER.024</ItemId>
      <ConditionId>OK</ConditionId>
      <ReservedQty>5</ReservedQty>
      <ReserveDateExpires>10.06.2014</ReserveDateExpires>
      <ExternalId>BabaLuba</ExternalId>
      <ExternalComment>lorem ipsum dolore anos amet</ExternalComment>
    </WareItem>
    <WareItem>
      <ItemId>SEB-100</ItemId>
      <ConditionId>OK</ConditionId>
      <ReservedQty>2</ReservedQty>
      <ReserveDateExpires>10.06.2014</ReserveDateExpires>
      <ExternalId>Two</ExternalId>
      <ExternalComment>lorem ipsum dolore anos amet</ExternalComment>
    </WareItem>
  </Body>
</Response>
```

пример 8

Структура документа полностью аналогична структуре документов, возвращаемых методами выгрузки структуры каталога и категории.

Внутри корневого тэга **<Response>** находятся тэги **<Header>** и **<Body>**.

Тэг **<ReserveSum>** содержит общую стоимость зарезервированных артикулов.

Тэг **<ReserveSumCurrency>** определяет валюту, в которой приведена суммарная стоимость резервов.

Тэг **<ReserveSumLimit>** содержит максимальный размер суммы резервирования доступный пользователю, а тэг **<ReserveSumLimitCurrency>** определяет валюту лимита резерва.

Далее следует 0 или много тэгов **<WareItem>**, определяющих список зарезервированных артикулов.

Каждый зарезервированный артикул (тэг **<WareItem>**) имеет следующие свойства:

**<ItemId>** – идентификатор артикула;

**<ConditionId>** – состояние упаковки; возможные значения: ОК (нормальная), мятая, из ремонта, некомплект, с теста, вскрытая, «пусто» (если попытка удаления завершилась неудачно);

**<ReservedQty>** – зарезервированное количество;

**<ReserveDateExpires>** – дата окончания резервирования.

**<ExternalId>** – идентификатор резерва, может быть пустым. Длина ExternalId не должна превышать 100 символов.

**<ExternalComment>** – произвольный комментарий, может быть пустым. Может меняться другими пользователями компании в B2B. Длина не более 100 символов.

Далее приведён документ, возвращённый методом CheckReserveStatus в формате JSON (пример 9):

```
{
  "Header": {
    "Code": 0,
    "Message": "",
    "Key": ""
  },
  "Body": {
    "ReserveSum": "2039,00",
    "ReserveSumCurrency": "USD",
    "ReserveSumLimit": "650000,00",
    "ReserveSumLimitCurrency": "USD",
    "WareItem": [
      {
        "ItemId": "NX.MGRER.024",
        "ConditionId": "OK",
        "ReservedQty": 5,
        "ReserveDateExpires": "10.06.2014",
        "ExternalId": "BabaLuba",
        "ExternalComment": "lorem ipsum dolore anos amet"
      },
      {
        "ItemId": "SEB-100",
        "ConditionId": "OK",
        "ReservedQty": 2,
        "ReserveDateExpires": "10.06.2014",
        "ExternalId": "Two",

```

```

    "ExternalComment": "lorem ipsum dolore anos amet"
  }
]
}
}

```

пример 9

## 4.3 Метод CreateReserve

### 4.3.1 Вызов метода

Метод **CreateReserve** проверяет наличие резерва по указанному артикулу и, если он есть, пытается добавить указанное в запросе количество к зарезервированному. В случае отсутствия резерва по указанному артикулу метод создает его.

Для вызова метода **CreateReserve** следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/CreateReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version="1.0" encoding="utf-8"?>](https://b2b.marvel.ru/Api/CreateReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version=)

```

<Root>
  <ReserveItemRequest>
    <ItemId>NX.MGRER.024</ItemId>
    <ConditionId>OK</ConditionId>
    <Qty>5</Qty>
    <ReserveInTransit>0</ReserveInTransit>
    <ExternalId>BabaLuba</ExternalId>
    <ExternalComment>lorem ipsum dolore anos amet</ExternalComment>
    <Dimension>net</Dimension>
  </ReserveItemRequest>
  <ReserveItemRequest>
    <ItemId>E2243Fw2</ItemId>
    <ConditionId>OK</ConditionId>
    <Qty>4</Qty>
    <ReserveInTransit>0</ReserveInTransit>
    <ExternalId>Two</ExternalId>
    <ExternalComment>lorem ipsum dolore anos amet</ExternalComment>
  </ReserveItemRequest>
</Root>

```

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **secretKey** – устаревший параметр. Передавать в него пустую строку или игнорировать.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **Items** – XML строка, содержащая список артикулов для резервирования (в примере выделена красным). Тэги **<ReserveItemRequest>** содержат необходимые данные для резервирования артикула, а именно:

- артикул (тэг **<ItemId>**);
- состояние упаковки (тэг **<ConditionId>**): ОК (нормальная), мятая, из ремонта, некомплект, с теста, вскрытая, «пусто» (если попытка удаления завершилась неудачно);
- количество позиций артикула для резервирования (тэг **<Qty>**);
- признак резервирования артикулов, находящихся в транзите (тэг **<ReserveInTransit>**). Тэг *<ReserveInTransit>* не используется в текущей версии. Однако он должен присутствовать и иметь значение «0».
- идентификатор резерва (тэг **<ExternalId>**). Тэг можно оставить пустым или вовсе не указывать. В этом случае значением тэга будет пустая строка.
- комментарий (тэг **<ExternalComment>**). Тэг можно оставить пустым или вовсе не указывать. В этом случае значением тэга будет пустая строка.
- локация склада (тэг **<Dimension>**). Тэг можно не указывать, в этом случае будет использоваться локация по-умолчанию (осн).

В параметр **Items** можно передавать как XML-строку, так и JSON-строку.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **Items** с помощью URLEncode.

Пример JSON-строки (пример 10):

```
{
  "ReserveItemRequest": [
    {
      "ItemId": "NX.MGRER.024",
      "ConditionId": "OK",
      "Qty": "5",
      "ReserveInTransit": "0",
      "ExternalId": "BabaLuba",
      "ExternalComment": "lorem ipsum dolore anos amet",
      "Dimension": "net"
    },
    {
      "ItemId": "SEB-100",
      "ConditionId": "OK",
      "Qty": "2",
      "ReserveInTransit": "0",
      "ExternalId": "Two",
      "ExternalComment": "lorem ipsum dolore anos amet"
    }
  ]
}
```

пример 10

Если передать методу запрос на резервирование 0 позиций артикула (тэг **<Qty>**), произойдёт следующее:

- Если резерв по данному артикулу уже имелся, то в качестве ответа будет возвращена информация о существующем резерве.

- Если резерва по артикулу не существует (Вы просите к нулю добавить ноль), то получите в качестве ответа, что зарезервировано 0 позиций с упаковкой = «пусто» до даты = «пусто».

### 4.3.2 Возвращаемые данные

Метод **CreateReserve** возвращает XML или JSON стандартной структуры.

Ниже приведен документ, возвращенный методом в формате XML (*пример 11*).

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
    <Key></Key>
  </Header>
  <Body>
    <ReserveSum>2039,00</ReserveSum>
    <ReserveSumCurrency>USD</ReserveSumCurrency>
    <ReserveSumLimit>650000,00</ReserveSumLimit>
    <ReserveSumLimitCurrency>USD</ReserveSumLimitCurrency>
    <WareItem>
      <ItemId>NX.MGRER.024</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <ReservedQty>5</ReservedQty>
      <ReserveDateExpires>10.06.2014</ReserveDateExpires>
      <ErrorCode>0</ErrorCode>
      <ErrorCodeDescription />
      <ExternalId>BabaLuba</ExternalId>
    </WareItem>
    <WareItem>
      <ItemId>SEB-100</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <ReservedQty>2</ReservedQty>
      <ReserveDateExpires>10.06.2014</ReserveDateExpires>
      <ErrorCode>0</ErrorCode>
      <ErrorCodeDescription />
      <ExternalId>Two</ExternalId>
    </WareItem>
  </Body>
</Response>
```

пример 11

Описание тэгов **<ReserveSum>**, **<ReserveSumCurrency>**, **<ReserveSumLimit>** и **<ReserveSumLimitCurrency>** см. в разделе «[Метод CheckReserveStatus](#)».

Вслед за этими тэгами следует один или много тэгов **<WareItem>**, каждый из которых содержит результат попытки резервирования артикулов запроса.

- **<ItemId>** – идентификатор артикула, для которого запросили резервирование.
- **<ConditionId>** – состояние упаковки; возможные значения: ОК (нормальная), мятая, из ремонта, некомплект, с теста, вскрытая, «пусто» (если попытка удаления завершилась неудачно).

- **<Dimension>** – локация склада; возможные значения: Основная (осн), NET.
- **<ReservedQty>** – реальное количество зарезервированного. Может быть меньше запрошенного.
- **<ReserveDateExpires>** – содержит дату окончания резервирования или «пусто», если попытка резервирования завершилась неудачно.
- **<ErrorCode>** – код ошибки операции резервирования артикула. Коды ошибок (одинаковы для методов **CreateReserve** и **ModifyReserve**):
  - **0** – операция резервирования успешна: зарезервировано столько, сколько запрошено;
  - **1** – операция резервирования успешна частично: зарезервировано меньше, чем запрошено;
  - **2** – превышен лимит резервирования пользователя;
  - **3** – для резервирования артикула следует обратиться к менеджеру;
  - **4** – товара нет на складе;
  - **5** – другая ошибка;
  - **6** – Превышено максимально допустимое количество зарезервированных артикулов.
- **<ExternalId>** – идентификатор резерва.

Пример возвращаемого документа для метода **CreateReserve** в формате JSON (пример 12):

```
{
  "Header": {
    "Code": 0,
    "Message": "",
    "Key": ""
  },
  "Body": {
    "ReserveSum": "4078,00",
    "ReserveSumCurrency": "USD",
    "ReserveSumLimit": "650000,00",
    "ReserveSumLimitCurrency": "USD",
    "WareItem": [
      {
        "ItemId": "NX.MGRER.024",
        "ConditionId": "OK",
        "Dimension": "Основная",
        "ReservedQty": 10,
        "ReserveDateExpires": "10.06.2014",
        "ErrorCode": "0",
        "ErrorCodeDescription": null,
        "ExternalId": "BabaLuba"
      },
      {
        "ItemId": "SEB-100",
        "ConditionId": "OK",
        "Dimension": "Основная",
        "ReservedQty": 4,
        "ReserveDateExpires": "10.06.2014",
        "ErrorCode": "0",
        "ErrorCodeDescription": null,

```



```

    "ExternalId": "Two"
  }
]
}
}

```

пример 12

## 4.4 Метод ModifyReserve

Для вызова метода **ModifyReserve** следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/ModifyReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version="1.0" encoding="utf-8"?>](https://b2b.marvel.ru/Api/ModifyReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version=)

```

<Root>
  <ReserveItemRequest>
    <ItemId>NX.MGRER.024</ItemId>
    <ConditionId>OK</ConditionId>
    <Qty>5</Qty>
    <ReserveInTransit>0</ReserveInTransit>
    <ExternalId>BabaLuba</ExternalId>
    <ExternalComment>lorem ipsum dolore anos amet</ExternalComment>
    <Dimension>осн</Dimension>
  </ReserveItemRequest>
  <ReserveItemRequest>
    <ItemId>SEB-100</ItemId>
    <ConditionId>OK</ConditionId>
    <Qty>2</Qty>
    <ReserveInTransit>0</ReserveInTransit>
    <ExternalId>Two</ExternalId>
    <ExternalComment>lorem ipsum dolore anos amet</ExternalComment>
  </ReserveItemRequest>
</Root>

```

Метод **ModifyReserve** очень похож на метод **CreateReserve**. Форматы входных документов (Items) совпадают. Форматы возвращаемых документов и коды ошибок также совпадают.

Разница в том, что метод **ModifyReserve** замещает имеющееся количество резерва на заданное, а метод **CreateReserve** – добавляет к имеющемуся количеству.

**Важно!** Если в метод **ModifyReserve** передать количество позиций артикула, равное нулю, то имеющийся резерв будет удалён. См. поведение метода **CreateReserve** при передаче нулевого количества позиций в разделе «Метод CreateReserve».

## 4.5 Метод DeleteReserve

### 4.5.1 Вызов метода

Для вызова метода **DeleteReserve** следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/DeleteReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version="1.0" encoding="utf-8"?>](https://b2b.marvel.ru/Api/DeleteReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version=)

```

<Root>
  <ReserveItemRequest>
    <ItemId>NX.MGRER.024</ItemId>
    <ConditionId>OK</ConditionId>

```

```

    <ReserveInTransit>0</ReserveInTransit>
    <ExternalId>BabaLuba</ExternalId>
    <Dimension>net</Dimension>
  </ReserveItemRequest>
<ReserveItemRequest>
  <ItemId>SEB-100</ItemId>
  <ConditionId>OK</ConditionId>
  <ReserveInTransit>0</ReserveInTransit>
  <ExternalId>Two</ExternalId>
</ReserveItemRequest>
</Root>

```

Внутри корневого тэга **<Root>** содержится один или больше тэгов **<ReserveItemRequest>**, каждый из которых является запросом на удаление резерва по указанному артикулу.

➤ **<ItemId>** – идентификатор артикула.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **ItemId** с помощью URLEncode.

➤ **<ConditionId>** – состояние упаковки; возможные значения: ОК (нормальная), мятая, из ремонта, некомплект, с теста, вскрытая, «пусто» (если попытка удаления завершилась неудачно).

➤ **<ReserveInTransit>** – признак резервирования артикулов, находящихся в транзите. *Тэг <ReserveInTransit> не используется в текущей версии. Однако он должен присутствовать и иметь значение «0».*

➤ **<ExternalId>** – идентификатор резерва. Тэг можно оставить пустым или вовсе не указывать. В этом случае значением тэга будет пустая строка.

➤ **<Dimension>** – локация склада. Тэг можно не указывать, в этом случае будет использоваться локация по-умолчанию (осн).

Формат документа, передаваемого в параметре items метода DeleteReserve в JSON (пример 13):

```

"ReserveItemRequest": [
  {
    "ItemId": "NX.MGRER.024",
    "ConditionId": "OK",
    "ExternalId": "BabaLuba",
    "Dimension": "осн"
  },
  {
    "ItemId": "SEB-100",
    "ConditionId": "OK",
    "ExternalId": "Two"
  }
]

```

пример 13

#### 4.5.2 Возвращаемые данные

Метод **DeleteReserve** возвращает документ стандартной структуры в формате XML или JSON.

Пример в формате XML (пример 14):

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
    <Key></Key>
  </Header>
  <Body>
    <ReserveSum>0,00</ReserveSum>
    <ReserveSumCurrency>USD</ReserveSumCurrency>
    <ReserveSumLimit>650000,00</ReserveSumLimit>
    <ReserveSumLimitCurrency>USD</ReserveSumLimitCurrency>
    <WareItem>
      <ItemId>NX.MGRER.024</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <ReservedQty>0</ReservedQty>
      <ReserveDateExpires></ReserveDateExpires>
      <ErrorCode>0</ErrorCode>
      <ErrorCodeDescription />
      <ExternalId>BabaLuba</ExternalId>
    </WareItem>
    <WareItem>
      <ItemId>SEB-100</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <ReservedQty>0</ReservedQty>
      <ReserveDateExpires></ReserveDateExpires>
      <ErrorCode>0</ErrorCode>
      <ErrorCodeDescription />
      <ExternalId>Two</ExternalId>
    </WareItem>
  </Body>
</Response>
```

пример 14

Описание тэгов **<ReserveSum>**, **<ReserveSumCurrency>**, **<ReserveSumLimit>** и **<ReserveSumLimitCurrency>** смотри в разделе [«Метод CheckReserveStatus»](#).

Вслед за этими тэгами следует один или много тэгов **<WareItem>**, каждый из которых содержит результат попытки удаления резерва артикулов из запроса.

- **<ItemId>** – идентификатор артикула, резерв которого запросили удалить.
- **<ConditionId>** – состояние упаковки; возможные значения: ОК (нормальная), мятая, из ремонта, некомплект, с теста, вскрытая, «пусто» (если попытка удаления завершилась неудачно).
- **<Dimension>** – локация склада; возможные значения: Основная (осн), NET.
- **<ReservedQty>** – реальное количество зарезервированного. 0 – если удаление произошло удачно, больше нуля – если при удалении произошли ошибки.
- **<ReserveDateExpires>** – содержит дату окончания резервирования (если при удалении резерва произошла ошибка) или «пусто», если попытка завершилась удачно.

- **<ErrorCode>** – код ошибки операции резервирования артикула. Коды ошибок:
  - **0** – операция удаления успешна;
  - **1** – резерв, указанный к удалению, не найден;
  - **2** – другая ошибка.
- **<ExternalId>** – идентификатор резерва.

## 4.6 Метод TransferReserve

### 4.6.1 Вызов метода

Метод **TransferReserve** позволяет перевести резерв частично или полностью с одного внешнего идентификатора резерва (**ExternalId**) на другой внешний идентификатор. Этот метод гарантирует то, что резерв при изменении внешнего идентификатора не будет потерян. Для вызова метода следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/TransferReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version="1.0" encoding="utf-8"?>](https://b2b.marvel.ru/Api/TransferReserve?user=usrLogin&password=usrPassword&secretKey=&responseFormat=0&items=<?xml version=)

```
<Root>
  <TransferReserve>
    <ItemId>JBLHORIZONBLKEU</ItemId>
    <ConditionId>OK</ConditionId>
    <TransferQty>5</TransferQty>
    <ExternalIdFrom>100</ExternalIdFrom>
    <ExternalIdTo>300</ExternalIdTo>
    <ExternalCommentTo>lorem ipsum dolore anos amet</ExternalCommentTo>
    <Dimension>осн</Dimension>
  </TransferReserve>
</Root>
```

Внутри корневого тэга **<Root>** содержится один или больше тэгов **<TransferReserve>**, каждый из которых является запросом на перемещение резерва по указанному артикулу с одного внешнего идентификатора на другой.

Тэг **<TransferReserve>** содержит в себе следующие тэги:

- **<ItemId>** – идентификатор артикула.
- **<ConditionId>** – состояние упаковки; возможные значения: ОК (нормальная), мятая, из ремонта, некомплект, с теста, вскрытая.
- **<Dimension>** – локация склада; возможные значения: Основная (осн), NET.
- **<TransferQty>** – перемещаемое количество артикула. При попытке переместить больше, чем есть в исходном резерве, возвращается сообщение об ошибке. Если в тэге указать 0 (ноль), то будет перемещено всё зарезервированное количество по данному артикулу.
- **<ExternalIdFrom>** – внешний идентификатор резерва, с которого осуществляется перемещение. Если тэг пустой, перемещение будет осуществляться из резерва без внешнего идентификатора.

- **<ExternalIdTo>** – внешний идентификатор резерва, на который осуществляется перемещение. Если указанный внешний идентификатор не существует, то будет создан. Если тэг пустой, перемещение будет осуществляться в резерв без внешнего идентификатора.

Если тэги `<ExternalIdFrom />` и `<ExternalIdTo />` имеют одинаковое значение, то возвращается сообщение об ошибке.

- **<ExternalCommentTo>** – произвольный комментарий, который будет добавлен к резерву с внешним идентификатором `<ExternalIdTo />`.
- **<Dimension>** – локация склада; возможные значения: `осн`, `нет`.

Все тэги, кроме `<ExternalCommentTo>` и `<Dimension>`, являются обязательными.

Формат документа, передаваемого в параметре `items` метода `TransferReserve` в JSON:

```
{
  "TransferReserve": [
    {
      "ItemId": "JBLHORIZONBLKEU",
      "ConditionId": "OK",
      "TransferQty": "5",
      "ExternalIdFrom": "300",
      "ExternalIdTo": "500",
      "ExternalCommentTo": "lorem ipsum dolore anos amet"
    }
  ]
}
```

Пример 15

#### 4.6.2 Возвращаемые данные

Метод **TransferReserve** возвращает документ стандартной структуры в формате XML или JSON.

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
    <Key></Key>
  </Header>
  <Body>
    <WareItem>
      <ItemId>JBLHORIZONBLKEU</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <ReservedQty>5</ReservedQty>
      <ReserveDateExpires>05.05.2016</ReserveDateExpires>
      <ErrorCode>0</ErrorCode>
      <ErrorCodeDescription>Операция выполнена успешно</ErrorCodeDescription>
      <ExternalId>test</ExternalId>
    </WareItem>
  </Body>
</Response>
```

пример 16

В тэге <Body> содержится один или много тэгов <WareItem>, каждый из которых содержит результат попытки перемещения резерва артикула.

- <ItemId> – идентификатор артикула, резерв которого запросили переместить.
- <ConditionId> – состояние упаковки; возможные значения: ОК (нормальная), мятая, из ремонта, некомплект, с теста, вскрытая.
- <Dimension> – локация склада; возможные значения: Основная (осн), NET.
- <ReservedQty> – количество артикула, перемещенное на новый внешний идентификатор.
- <ReserveDateExpires> – дата окончания резерва, перемещённого на новый внешний идентификатор. В случае ошибки – пусто.
- <ErrorCode> – если операция прошла успешно, то 0. Иное значение есть признак того, что операция не прошла.
- <ErrorCodeDescription> – описание ошибки. Если успешно (код ошибки 0), то «Операция выполнена успешно».
- <ExternalId> – новый внешний идентификатор, на который переместился резерв частично или полностью.

#### 4.6.3 Ошибки метода TransferReserve

- **Код ошибки 3 – «Товар не доступен для автоматического резервирования»:** при появлении такой ошибки следует связаться с менеджером Марвел для прояснения статуса артикула.
- **Код ошибки 6 – «Превышено максимальное количество зарезервированных артикулов»:** при появлении такой ошибки следует связаться с менеджером Марвел и запросить увеличение числа одновременно зарезервированных позиций.
- **Код ошибки 11 – «Исходный резерв не найден»:** сделана попытка переместить несуществующий резерв. Следует уточнить список резервов, используя метод API CheckReserveStatus.
- **Код ошибки 12 – «Некорректное перемещаемое количество»:** сделана попытка переместить большее количество артикула нежели, чем зарезервировано.
- **Код ошибки 13 – «ExternalIdFrom и ExternalIdTo совпадают»:** внешние идентификаторы исходного резерва и целевого резерва одинаковы. Следует уточнить список резервов, используя метод API CheckReserveStatus.

## 5. Методы API для работы с заказами

### 5.1 Метод CreateSalesOrder

Метод **CreateSalesOrder** позволяет создать заказ как из зарезервированных ранее артикулов, так и без предварительного резервирования артикулов.

Для обращения к методу следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/CreateSalesOrder?user=usrLogin&password=usrPassword&responseFormat=0&OrderId=Order&ContractId=Contract&DeliveryAddressId=DeliveryAddress&items=<?xml version='1.0' encoding='utf-8'?>>

```
<Root>
  <Item>
    <ItemId>DQ.U91ER.004</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>5</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Dimension>осн</Dimension>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </Item>
  <ExternalOrderId>Order_Nomer_Odin</ExternalOrderId>
</Root>
```

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **OrderId** – номер заказа, необязательный параметр.
- **ContractId** – внутренний идентификатор договора в БД, необязательный параметр.
- **DeliveryAddressId** – внутренний идентификатор адреса доставки в БД, необязательный параметр.
- **Items** – XML строка, содержащая список артикулов для добавления в заказ (в примере выделена красным). Подробно эта строка будет рассмотрена ниже.

Параметры **OrderId**, **ContractId**, **DeliveryAddressId** являются необязательными. Если они не переданы в запросе, заказ создастся с настройками по-умолчанию.

**ВАЖНО:** Для добавления артикулов в существующий заказ в запросе в тэге **OrderId** нужно указать номер неотгруженного заказа.

Для создания нового заказа с определенным номером договора и адресом доставки нужно в запросе передать данные в тэгах **ContractId**, **DeliveryAddressId**.

### 5.1.1 Создание заказа без предварительного резервирования

Для создания заказа нужно сформировать документ (XML-строку и поместить ее в параметр `Items`) следующей структуры:

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <Item>
    <ItemId>DQ.U91ER.007</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>5</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Dimension>осн</Dimension>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </Item>
  <Item>
    <ItemId>DQ.U91ER.004</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>1</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </Item>
  <ExternalOrderId>Order_Nomer_Odin</ExternalOrderId>
</Root>
```

Пример 17

Внутри тэга `<Root />` находится список тэгов `<Item />`, которые определяют перечень артикулов, добавляемых в заказ.

Тег `<Item />` (должно быть не менее одного или много) содержит следующий набор тэгов, характеризующих заказываемый артикул:

`<ItemId />` – идентификатор артикула. При попытке заказать артикул, отсутствующий в номенклатуре, создание заказа немедленно прерывается и возвращается сообщение об ошибке («Артикул ... отсутствует в классификаторе»).

`<ConditionId />` – состояние упаковки. Возможные значения: ОК, мятая, из ремонта, некомплект, с теста, вскрытая, «пусто» (любая упаковка).

`<Qty />` – заказываемое количество. Будет зарезервировано количество, свободное на складе. Поэтому может получиться, что не все заказываемое количество товара в заказе зарезервировано на складе.

`<ExternalId />` – произвольный идентификатор строки заказа, может быть пустым, но присутствие тэга обязательно.

`<Dimension />` – локация склада. Можно не указывать, тогда будет использована локация по-умолчанию (осн).

`<Comment />` – произвольный комментарий к строке заказа, может быть пустым, но присутствие тэга обязательно.

Кроме того, внутри тэга `<Root />` находится тэг `<ExternalOrderId />`, в котором можно передать произвольный идентификатор заказа или просто оставить этот тэг пустым. Можно использовать этот тэг как комментарий ко всему заказу, например: «Очень срочно!». Тэг `<ExternalOrderId />` должен присутствовать в документе, даже если он пуст.

В параметр `Items` можно передать и JSON-строку (пример 18).



```

{
  "Item": [
    {
      "ItemId": "DQ.U91ER.007",
      "ConditionId": "Нормальная",
      "Qty": 5,
      "ExternalId": "BabaLuba",
      "Dimension": "осн",
      "Comment": "lorem ipsum dolore anos amet"
    },
    {
      "ItemId": "DQ.U91ER.004",
      "ConditionId": "Нормальная",
      "Qty": 1,
      "ExternalId": "BabaLuba",
      "Comment": "lorem ipsum dolore anos amet"
    }
  ],
  "ExternalOrderId": "Order_Nomer_Odin"
}

```

Пример 18

Значения полей JSON-документа такие же, как и у полей XML-документа.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **Items** с помощью URLEncode.

### 5.1.2 Создание заказа из существующего резерва

При создании заказа из существующего резерва следует понимать, что сначала артикулы должны быть зарезервированы.

Если артикул, который Вы указываете как зарезервированный, реально отсутствует в резерве, то такой артикул в запросе игнорируется и не будет добавлен в заказ.

В заказ можно поместить количество не больше, чем зарезервировано. Меньше – можно. Тогда часть резерва попадёт в заказ, а часть останется зарезервированным.

Структура XML и JSON документов для создания заказа из резервов совпадает со структурой документа для создания заказа без предварительного резервирования за одним исключением: тэг <Item /> следует заменить на тэг <ReserveItem />.

```

<?xml version="1.0" encoding="utf-8"?>
<Root>
  <ReserveItem>
    <ItemId>NX.EFAER.018</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>5</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Dimension>осн</Dimension>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </ReserveItem>
  <ReserveItem>
    <ItemId>NX.EFAER.013</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>1</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Comment>lorem ipsum dolore anos amet</Comment>

```

```
</ReserveItem>
<ExternalOrderId>Order_Nomer_Odin</ExternalOrderId>
</Root>
```

Пример 19

**ВАЖНО:** При создании заказа из резерва ExternalId следует указывать тот же самый, что и у резерва. В противном случае резерв не будет найден.

### 5.1.3 Создание комбинированного заказа

В один заказ можно поместить артикулы, как из резерва, так и предварительно незарезервированные.

Это осуществляется простым комбинированием тэгов <Item /> и <ReserveItem />.

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <Item>
    <ItemId>DQ.U91ER.007</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>5</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Dimension>осн</Dimension>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </Item>
  <Item>
    <ItemId>DQ.U91ER.004</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>1</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </Item>
  <ReserveItem>
    <ItemId>NX.EFAER.018</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>5</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </ReserveItem>
  <ReserveItem>
    <ItemId>NX.EFAER.013</ItemId>
    <ConditionId>Нормальная</ConditionId>
    <Qty>1</Qty>
    <ExternalId>BabaLuba</ExternalId>
    <Comment>lorem ipsum dolore anos amet</Comment>
  </ReserveItem>
  <ExternalOrderId>Order_Nomer_Odin</ExternalOrderId>
</Root>
```

Пример 20

### 5.1.4 Возвращаемые данные

Если создание заказа прошло успешно, то возвращается документ стандартной структуры:

```

<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
  </Header>
  <Body>
    <OrderId>b2b_0102267</OrderId>
    <OrderDate>11.03.2016</OrderDate>
    <OrderAuthor>some_user</OrderAuthor>
    <OrderCurrencyId>RUR</OrderCurrencyId>
    <OrderShippingWarehouseId>СПБ-Киевская</OrderShippingWarehouseId>
    <OrderShipmentType>Доставка до ТК</OrderShipmentType>
    <OrderClientLegalEntityName>000 Крыжопольские
самовары</OrderClientLegalEntityName>
    <OrderStatus>Backorder</OrderStatus>
    <OrderSumCurrency>188645,40</OrderSumCurrency>
    <OrderSumUsd>2376,00</OrderSumUsd>
    <ExternalSalesOrderId>Order_Nomer_Odin< ExternalSalesOrderId>
    <OrderContractId>1040788</OrderContractId>
    <OrderMarvelLegalEntityName>Марвел КТ</OrderMarvelLegalEntityName>
    <SalesOrderItem>
      <ItemId>DQ.U91ER.004</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <OrderedQty>1</OrderedQty>
      <ReservedQty>0</ReservedQty>
      <ReserveDateExpires>01.01.0001</ReserveDateExpires>
      <CurrencyId>USD</CurrencyId>
      <PriceCurrency>421,00</PriceCurrency>
      <PriceUsd>421,00</PriceUsd>
    </SalesOrderItem>
    <SalesOrderItem>
      <ItemId>DQ.U91ER.007</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <OrderedQty>5</OrderedQty>
      <ReservedQty>5</ReservedQty>
      <ReserveDateExpires>18.03.2016</ReserveDateExpires>
      <CurrencyId>USD</CurrencyId>
      <PriceCurrency>391,00</PriceCurrency>
      <PriceUsd>391,00</PriceUsd>
    </SalesOrderItem>
    <NotOrderedItem>NX.EFAER.018</NotOrderedItem>
    <NotOrderedItem>NX.EFAER.013</NotOrderedItem>
  </Body>
</Response>

```

Пример 21

**<OrderId />** – содержит номер заказа

**<OrderDate />** – содержит дату создания заказа

**<OrderAuthor />** – содержит логин пользователя, создавшего заказ

**<OrderCurrencyId />** – содержит идентификатор валюты заказа

**<OrderShippingWarehouseId />** – склад отгрузки заказа

**<OrderShipmentType />** – тип отгрузки заказа

**<OrderClientLegalEntityName />** – юридическое название компании пользователя

**<OrderStatus />** – текущий статус заказа (при создании всегда Backorder)

**<OrderSumCurrency />** – сумма заказа в валюте заказа (определена в тэге **<OrderCurrencyId />**), пересчитанная по текущему курсу ЦБ РФ

**<OrderSumUsd />** – сумма заказа в USD по текущему курсу ЦБ РФ валюте по курсу на день создания заказа

**<ExternalSalesOrderId />** – строка комментария ко всему заказу

**<OrderContractId />** – код договора с клиентом

**<OrderMarvelLegalEntityName />** – юридическое лицо компании Марвел.

Далее следует один или много тэгов **<SalesOrderItem />**, каждый из которых содержит информацию о заказанных артикулах.

Каждый тэг **<SalesOrderItem />** содержит следующие тэги:

**<ItemId />** – идентификатор артикула.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **ItemId** с помощью URLEncode.

**<ConditionId />** – состояние упаковки.

**<Dimension />** – локация склада.

**<OrderedQty />** – заказанное количество артикула.

**<ReservedQty />** – зарезервированное количество артикула, всегда меньше или равно заказанному количеству.

**<ReserveDateExpires />** – если зарезервированное количество (**<ReservedQty />**) больше нуля, то содержит дату окончания резервирования.

**<CurrencyId />** – валюта прайс-листа артикула.

**<PriceCurrency />** – стоимость артикула в валюте прайс-листа.

**<PriceUsd />** – стоимость артикула в USD по текущему курсу ЦБ РФ.

После последовательности тэгов **<SalesOrderItem />** (один или больше) может присутствовать последовательность тэгов **<NotOrderedItem />** (ноль или больше).

Последовательность тэгов **<NotOrderedItem />** содержит артикулы, которые не могут быть автоматически добавлены в заказ. Для заказа таких артикулов необходимо обратиться к менеджеру Марвел.

## 5.2 Метод CheckSalesOrderStatus

Метод **CheckSalesOrderStatus** позволяет получить информацию об одном или нескольких заказах.

Для обращения к методу следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version="1.0" encoding="utf-8"?><Root></Root>](https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version=)

В этом случае (пустой тэг <Root/>) возвращаются все открытые и частично отгруженные заказы пользователя. Внутри тэга <Root/> можно добавлять дополнительные тэги-параметры, которые будут описаны ниже.

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **requestParams** – XML строка, содержащая список необязательных параметров для поиска интересующих заказов (в примере выделена красным).

### 5.2.1 Тэг-параметр <SalesOrderShipmentStatus/> requestParams метода CheckSalesOrderStatus

Тэг <SalesOrderShipmentStatus/> позволяет задать фильтрацию заказов по статусу отгрузки заказа:

- **0** – открытые и частично отгруженные заказы (значение по умолчанию)
- **1** – отгруженные заказы (выводятся отгруженные заказы, созданные за последние 365 дней)
- **2** – все заказы (выводятся все заказы, созданные за последние 365 дней).

При разборе параметров анализируется только первый тэг <SalesOrderShipmentStatus/>, а все последующие игнорируются. Пример использования:

[https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version="1.0" encoding="utf-8"?><Root><SalesOrderShipmentStatus>2</SalesOrderShipmentStatus></Root>](https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version=)

### 5.2.2 Тэг-параметр <SalesOrderId/> requestParams метода CheckSalesOrderStatus

Используется для указания номеров заказов (один или несколько тэгов <SalesOrderId/>), которые ищутся в списке отфильтрованных по статусу отгрузки заказов. Если Вам не известен статус отгрузки заказа, используйте фильтр “все заказы”.

Пример использования:

[https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version="1.0" encoding="utf-8"?>](https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version='1.0' encoding='utf-8'?>)

```
<Root>
  <SalesOrderShipmentStatus>2</SalesOrderShipmentStatus>
  <SalesOrderId>b2b_0105959</SalesOrderId>
  <SalesOrderId>b2b_0105957</SalesOrderId>
  <SalesOrderId>b2b_0105953</SalesOrderId>
</Root>
```

### 5.2.3 Тэг-параметр <ExternalSalesOrderId/> requestParams метода CheckSalesOrderStatus

Используется для указания внешних идентификаторов заказов (один или несколько тэгов <ExternalSalesOrderId />), которые ищутся в списке отфильтрованных по статусу отгрузки заказов. Если Вам не известен статус отгрузки заказа, используйте фильтр “все заказы”.

Пример использования:

[https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version="1.0" encoding="utf-8"?>](https://b2b.marvel.ru/Api/CheckSalesOrderStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version='1.0' encoding='utf-8'?>)

```
<Root>
  <SalesOrderShipmentStatus>2</SalesOrderShipmentStatus>
  <ExternalSalesOrderId>заказ_01</ExternalSalesOrderId>
  <ExternalSalesOrderId>важный заказ</ExternalSalesOrderId>
  <ExternalSalesOrderId>самый важный заказ</ExternalSalesOrderId>
</Root>
```

### 5.2.4 Возвращаемые данные

Возвращается список найденных заказов.

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
  </Header>
  <Body>
    <OrderInfo>
      <SalesOrderId>b2b_0105957</SalesOrderId>
      <OrderShipmentStatus>Открыто</OrderShipmentStatus>
      <OrderPaymentStatus>Не оплачен</OrderPaymentStatus>
      <OrderDate>07.04.2016</OrderDate>
      <OrderAuthor>Пользователь</OrderAuthor>
      <ExternalSalesOrderId>Заказ из АПИ</ExternalSalesOrderId>
      <OrderShippingWarehouseId>11201</OrderShippingWarehouseId>
      <OrderShipmentType>Доставка</OrderShipmentType>
      <OrderClientLegalEntity>7811071820</OrderClientLegalEntity>
      <OrderClientLegalEntityName>000 "Фирма"</OrderClientLegalEntityName>
      <OrderContractId></OrderContractId>
    </OrderInfo>
    <OrderInfo>
      <SalesOrderId>b2b_0105953</SalesOrderId>
      <OrderShipmentStatus>Открыто</OrderShipmentStatus>
      <OrderPaymentStatus>Не оплачен</OrderPaymentStatus>
      <OrderDate>06.04.2016</OrderDate>
      <OrderAuthor>Пользователь </OrderAuthor>
```

```

    <ExternalSalesOrderId>Заказ из API</ExternalSalesOrderId>
    <OrderShippingWarehouseId>11201</OrderShippingWarehouseId>
    <OrderShipmentType>Доставка</OrderShipmentType>
    <OrderClientLegalEntity>7811071820</OrderClientLegalEntity>
    <OrderClientLegalEntityName>000 "Фирма"</OrderClientLegalEntityName>
    <OrderContractId></OrderContractId>
  </OrderInfo>
</Body>
</Response>

```

Пример 22

**<SalesOrderId />** – содержит номер заказа.

**<OrderShipmentStatus/>** – статус заказа; возможные значения: "Открыто", "Частично отгружено", "Отгружено".

**<OrderPaymentStatus/>** – статус оплаты заказа; возможные значения: "Не оплачен", "Оплачен частично", "Оплачен".

**<OrderDate/>** – дата создания заказа.

**<OrderAuthor/>** – автор заказа.

**<ExternalSalesOrderId/>** – внешний идентификатор заказа, произвольная строка определяемая пользователем.

**<OrderShippingWarehouseId/>** – код склада отгрузки; возможные значения:

11201: СПб – Киевская

12203: МСК – Климовск

11219: Электронно (отгрузка электронных ключей и электронных лицензий).

**<OrderShipmentType />** – тип отгрузки; возможные значения: "Доставка", "Доставка до ТК", "Самовывоз", "Электронно".

**<OrderClientLegalEntity/>** – ИНН компании пользователя.

**<OrderClientLegalEntityName/>** – юридическое название компании пользователя.

**<OrderContractId/>** – код договора с клиентом.

**<OrderSumCurrency/>** – сумма заказа в валюте заказа.

**<OrderSumUSD/>** – сумма заказа в USD.

### 5.3 Метод CheckSalesOrderItemStatus

Метод **CheckSalesOrderStatus** позволяет получить информацию о строках одного или нескольких заказов.

Для обращения к методу следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/CheckSalesOrderItemStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version="1.0" encoding="utf-8"?><Root></Root>](https://b2b.marvel.ru/Api/CheckSalesOrderItemStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version=)

В этом случае (пустой тэг <Root/>) возвращаются все строки всех открытых и частично отгруженных заказов пользователя. Внутри тэга <Root/> можно добавлять дополнительные тэги-параметры, которые будут описаны ниже.

Описание параметров метода:

- **user** – имя пользователя API.

- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **requestParams** – XML строка, содержащая список необязательных параметров для поиска интересующих заказов (в примере выделена красным).

### 5.3.1 Тэг-параметр <SalesOrderShipmentStatus/> requestParams метода CheckSalesOrderItemStatus

Тэг <SalesOrderShipmentStatus/> позволяет задать фильтрацию заказов по статусу отгрузки заказа. Аналогично методу **CheckSalesOrderStatus**.

### 5.3.2 Тэг-параметр <SalesOrderId/> requestParams метода CheckSalesOrderItemStatus

Один или несколько тэгов <SalesOrderId /> производят поиск строк указанных заказов среди набора заказов, отфильтрованных по статусу отгрузки заказов. Аналогично методу **CheckSalesOrderStatus**.

### 5.3.3 Тэг-параметр <ExternalSalesOrderId/> requestParams метода CheckSalesOrderItemStatus

Один или несколько тэгов <ExternalSalesOrderId/> производят поиск строк заказов с указанными внешними идентификаторами среди набора заказов, отфильтрованных по статусу отгрузки заказов. Аналогично методу **CheckSalesOrderStatus**.

### 5.3.4 Тэг-параметр <ItemId/> requestParams метода CheckSalesOrderItemStatus

Один или несколько тэгов <ItemId/> производят поиск строк заказов, содержащих <ItemId>, среди набора заказов отфильтрованных по статусу отгрузки заказов.

Например, чтобы получить все строки с артикулом <ItemId> всех открытых и частично отгруженных заказов надо сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/CheckSalesOrderItemStatus?user=usrLogin&password=usrPassword&responseFormat=0&requestParams=<?xml version='1.0' encoding='utf-8'?>>

```
<Root>
  <SalesOrderShipmentStatus>0</SalesOrderShipmentStatus>
  <ItemId>DQ.VM8ER.039</ItemId>
</Root>
```

### 5.3.5 Возвращаемые данные

Возвращается список найденных строк заказов.

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
```



```

</Header>
<Body>
  <Item>
    <SalesOrderId>b2b_0105959</SalesOrderId>
    <ExternalSalesOrderId>Заказ из АПИ</ExternalSalesOrderId>
    <ItemId>DQ.VM8ER.039</ItemId>
    <ExternalId>res 2</ExternalId>
    <ConditionId>OK</ConditionId>
    <Dimension>Основная</Dimension>
    <OrderedQty>2</OrderedQty>
    <RemainingForShipmentQty>2</RemainingForShipmentQty>
    <ReservedQty>2</ReservedQty>
    <ReserveDateExpires>14.04.2016</ReserveDateExpires>
    <CurrencyId>USD</CurrencyId>
    <PriceCurrency>699,000000</PriceCurrency>
    <PriceUSD>699,000000</PriceUSD>
    <OrderShipmentStatus>Открыто</OrderShipmentStatus>
    <OrderPaymentStatus>Не оплачен</OrderPaymentStatus>
    <Comment />
  </Item>
</Body>
</Response>

```

Пример 23

**<SalesOrderId />** – содержит номер заказа.

**<ExternalSalesOrderId />** – внешний идентификатор заказа, определяемый пользователем АПИ. Может быть пустым.

**<ItemId />** – артикул в заказе.

**<ExternalId />** – внешний идентификатор строки заказа, определяемый пользователем. Может быть пустым. При создании заказа из резерва копируется из строки резерва.

**<ConditionId />** – состояние упаковки заказанного артикула.

**<Dimension />** – локация склада.

**<OrderedQty />** – заказанное количество артикула.

**<RemainingForShipmentQty />** – неотгруженное количество артикула.

**<ReservedQty />** – зарезервированное количество артикула.

**<ReserveDateExpires />** – дата снятия резерва.

**<CurrencyId />** – валюта прайс-листа артикула.

**<PriceCurrency />** – стоимость артикула в валюте прайс-листа.

**<PriceUSD />** – стоимость артикула в USD по текущему курсу ЦБ РФ.

**<OrderShipmentStatus />** – статус заказа; возможные значения: "Открыто", "Частично отгружено", "Отгружено".

**<OrderPaymentStatus />** – статус оплаты; возможные значения: "Не оплачен", "Оплачен частично", "Оплачен".

**<Comment />** – комментарий к строке заказа.

**<ShipmentLocationId />** – код склада отгрузки. Этот тэг появляется только после начала отгрузки заказа. Склад в строке заказа может отличаться от склада в заказе. То есть отгрузка позиций одного заказа может происходить с разных складов. Возможные значения:

11201: СПб – Киевская

12203: МСК – Климовск

11219: Электронно (отгрузка электронных ключей и электронных лицензий).

## 6. Метод **ModifySalesOrder**

Метод **ModifySalesOrder** позволяет изменить в неотгруженном заказе: № договора, адрес доставки, склад отгрузки. Узнать доступные параметры договоров можно с помощью метода **GetShipmentData**, описанного в п. 8.

Для обращения к методу **ModifySalesOrder** следует сформировать URL:

<https://b2b.marvel.ru/Api/ModifySalesOrder?user=usrLogin&password=usrPassword&responseFormat=0&OrderId=0&ContractId=0&DeliveryAddressId=0&OrderShippingWareHouseId=0>

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **OrderId** – номер заказа.
- **ContractId** – внутренний идентификатор договора в БД, необязательный параметр.
- **DeliveryAddressId** – внутренний идентификатор адреса доставки в БД, необязательный параметр.
- **OrderShippingWareHouseId** – склад отгрузки, необязательный параметр.

В запросе можно одновременно передать **ContractId**, **DeliveryAddressId** и **OrderShippingWareHouseId**. Если данные соответствуют актуальному договору, параметры заказа будут изменены. В противном случае будет возвращена соответствующая ошибка.

## 6.1 Возвращаемые данные

В зависимости от успешности выполнения запроса метод возвращается код ошибки Code. Значения:

- 250 – параметры заказа изменены;
- 251 – заказ не найден;
- 252 – параметры заказа не могут быть изменены;
- 253 – номер договора не может быть изменен, измените параметры запроса или обратитесь к менеджеру;
- 254 – адрес доставки не может быть изменен, измените параметры запроса или обратитесь к менеджеру;
- 255 – склад отгрузки не может быть изменен, измените параметры запроса или обратитесь к менеджеру;
- 256 – Вы не передали данные для изменения параметров заказа.

## 7. Метод `ModifySalesOrderLines`

Метод `ModifySalesOrderLines` позволяет изменять количество артикулов в строках неотгруженного заказа.

Для обращения к методу следует сформировать URL (POST запрос):

[https://b2b.marvel.ru/Api/ModifySalesOrderLines?user=usrLogin&password=usrPassword&responseFormat=0&orderId=order&items=<?xml version="1.0" encoding="utf-8"?>](https://b2b.marvel.ru/Api/ModifySalesOrderLines?user=usrLogin&password=usrPassword&responseFormat=0&orderId=order&items=<?xml version='1.0' encoding='utf-8'?>)

```
<Root>
  <Item>
    <ItemId>273V5LSB (00/01)</ItemId>
    <ConditionId>OK</ConditionId>
    <Qty>3</Qty>
    <ExternalId>осн</ExternalId>
    <Dimension>осн</Dimension>
    <Comment></Comment>
  </Item>
  <ExternalOrderId>Order_Nomer_Odin</ExternalOrderId>
  <Item>
    <ItemId>80YN0009RK</ItemId>
    <ConditionId>OK</ConditionId>
    <Qty>2</Qty>
    <ExternalId>осн</ExternalId>
    <Dimension></Dimension>
    <Comment></Comment>
  </Item>
  <ExternalOrderId>Order_Nomer_Odin</ExternalOrderId>
</Root>
```

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **orderId** – номер заказа.
- **Items** – XML строка, содержащая список артикулов для добавления в заказ (в примере выделена красным). Подробно эта строка будет рассмотрена ниже.

Внутри тэга <Root /> находится список тэгов <Item />, которые определяют перечень артикулов, добавляемых в заказ.

Тег <Item /> (должно быть не менее одного или много) содержит следующий набор тэгов, характеризующих заказываемый артикул:

<ItemId /> – идентификатор артикула. При попытке заказать артикул, отсутствующий в номенклатуре, создание заказа немедленно прерывается и возвращается сообщение об ошибке («Артикул ... отсутствует в классификаторе»).

<ConditionId /> – состояние упаковки. Возможные значения: ОК, мятая, из ремонта, некомплект, с теста, вскрытая, «пусто» (любая упаковка).

<Qty /> – количество. Количество артикула в заказе будет изменено на запрошенное. Будет зарезервировано количество, свободное на складе. Поэтому может получиться, что не все заказываемое количество товара в заказе зарезервировано на складе.

Если будет передан 0, строка с артикулом будет удалена из заказа.

<ExternalId /> – произвольный идентификатор строки заказа, может быть пустым, но присутствие тэга обязательно.

<Dimension /> – локация склада. Можно не указывать, тогда будет использована локация по-умолчанию (осн).

<Comment /> – произвольный комментарий к строке заказа, может быть пустым, но присутствие тэга обязательно.

Кроме того, внутри тэга <Root /> находится тэг <ExternalOrderId />, в котором можно передать произвольный идентификатор заказа или просто оставить этот тэг пустым. Можно использовать этот тэг как комментарий ко всему заказу, например: “Очень срочно!”. Тэг <ExternalOrderId /> должен присутствовать в документе, даже если он пуст.

В параметр **Items** можно передать и JSON-строку (пример 24).

```
{
  "Item": [
    {
      "ItemId": "DQ.U91ER.007",
      "ConditionId": "Нормальная",
      "Qty": 5,
      "ExternalId": "BabaLuba",
      "Dimension": "осн",
      "Comment": "lorem ipsum dolore anos amet"
    }
  ],
}
```

```

    {
      "ItemId": "DQ.U91ER.004",
      "ConditionId": "Нормальная",
      "Qty": 1,
      "ExternalId": "BabaLuba",
      "Comment": "lorem ipsum dolore anos amet"
    }
  ],
  "ExternalOrderId": "Order_Nomer_Odin"
}

```

Пример 24

Значения полей JSON-документа такие же, как и у полей XML-документа.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **Items** с помощью URLEncode.

## 7.1 Возвращаемые данные

Если изменение строк заказа прошло успешно, то возвращается документ стандартной структуры:

```

<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>0</Code>
    <Message></Message>
  </Header>
  <Body>
    <OrderId>b2b_0102267</OrderId>
    <OrderDate>11.03.2016</OrderDate>
    <OrderAuthor>some_user</OrderAuthor>
    <OrderCurrencyId>RUR</OrderCurrencyId>
    <OrderShippingWarehouseId>СПБ-Киевская</OrderShippingWarehouseId>
    <OrderShipmentType>Доставка до ТК</OrderShipmentType>
    <OrderClientLegalEntityName>000 Крыжопольские
самовары</OrderClientLegalEntityName>
    <OrderStatus>Backorder</OrderStatus>
    <OrderSumCurrency>188645,40</OrderSumCurrency>
    <OrderSumUsd>2376,00</OrderSumUsd>
    <ExternalSalesOrderId>Order_Nomer_Odin< ExternalSalesOrderId>
    <OrderContractId>1443908</OrderContractId>
    <OrderMarvelLegalEntityName>Марвел КТ</OrderMarvelLegalEntityName>
    <SalesOrderItem>
      <ItemId>DQ.U91ER.004</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>
      <OrderedQty>1</OrderedQty>
      <ReservedQty>0</ReservedQty>
      <ReserveDateExpires>01.01.0001</ReserveDateExpires>
      <CurrencyId>USD</CurrencyId>
      <PriceCurrency>421,00</PriceCurrency>
      <PriceUsd>421,00</PriceUsd>
    </SalesOrderItem>
    <SalesOrderItem>
      <ItemId>DQ.U91ER.007</ItemId>
      <ConditionId>OK</ConditionId>
      <Dimension>Основная</Dimension>

```

```

    <OrderedQty>5</OrderedQty>
    <ReservedQty>5</ReservedQty>
    <ReserveDateExpires>18.03.2016</ReserveDateExpires>
    <CurrencyId>USD</CurrencyId>
    <PriceCurrency>391,00</PriceCurrency>
    <PriceUsd>391,00</PriceUsd>
  </SalesOrderItem>
  <NotOrderedItem>NX.EFAER.018</NotOrderedItem>
  <NotOrderedItem>NX.EFAER.013</NotOrderedItem>
</Body>
</Response>

```

Пример 25

**<OrderId />** – содержит номер заказа

**<OrderDate />** – содержит дату создания заказа

**<OrderAuthor />** – содержит логин пользователя, создавшего заказ

**<OrderCurrencyId />** – содержит идентификатор валюты заказа

**<OrderShippingWarehouseId />** – склад отгрузки заказа

**<OrderShipmentType />** – тип отгрузки заказа

**<OrderClientLegalEntityName />** – юридическое название компании пользователя

**<OrderStatus />** – текущий статус заказа (при создании всегда Backorder)

**<OrderSumCurrency />** – сумма заказа в валюте заказа (определена в тэге <OrderCurrencyId />), пересчитанная по текущему курсу ЦБ РФ

**<OrderSumUsd />** – сумма заказа в USD по текущему курсу ЦБ РФ валюте по курсу на день создания заказа

**<ExternalSalesOrderId />** – строка комментария ко всему заказу

**<OrderContractId />** – код договора с клиентом

**<OrderMarvelLegalEntityName />** – юридическое лицо компании Марвел.

Далее следует один или много тэгов <SalesOrderItem />, каждый из которых содержит информацию о заказанных артикулах.

Каждый тэг <SalesOrderItem /> содержит следующие тэги:

**<ItemId />** – идентификатор артикула.

Поскольку название артикула может содержать специальные символы, перед отправкой запроса необходимо кодировать **ItemId** с помощью URLEncode.

**<ConditionId />** – состояние упаковки.

**<Dimension />** – локация склада.

**<OrderedQty />** – заказанное количество артикула.

**<ReservedQty />** – зарезервированное количество артикула, всегда меньше или равно заказанному количеству.

**<ReserveDateExpires />** – если зарезервированное количество (**<ReservedQty />**) больше нуля, то содержит дату окончания резервирования.

**<CurrencyId />** – валюта прайс-листа артикула.

**<PriceCurrency />** – стоимость артикула в валюте прайс-листа.

**<PriceUsd />** – стоимость артикула в USD по текущему курсу ЦБ РФ.

После последовательности тэгов **<SalesOrderItem />** (один или больше) может присутствовать последовательность тэгов **<NotOrderedItem />** (ноль или больше).

Последовательность тэгов **<NotOrderedItem />** содержит артикулы, которые не могут быть автоматически добавлены в заказ. Для заказа таких артикулов необходимо обратиться к менеджеру Марвел.

## 8. Метод **GetShipmentData**

Метод **GetShipmentData** позволяет получить информацию о параметрах договоров, которые доступны пользователю, для создания и отгрузки заказов.

Для обращения к методу следует сформировать URL (POST-запрос):

<https://b2b.marvel.ru/Api/GetShipmentData?user=usrLogin&password=usrPassword&responseFormat=0>

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.

### 8.1 Возвращаемые данные

- **OrderContractId** – внутренний идентификатор договора в БД.
- **OrderContractName** – номер договора.
- **OrderClientLegalEntityName** – юридическое название компании пользователя.
- **OrderMarvelLegalEntityName** – юридическое лицо компании Марвел.
- **OrderShipmentType** – тип отгрузки.

Если доступны договора с типом отгрузки «доставка» или «доставка до ТК», далее следует один или много тэгов **<DeliveryAddress />**, каждый из которых содержит информацию об адресах доставки:

- **DeliveryAddressName** – адрес доставки.
- **DeliveryAddressId** – внутренний идентификатор адреса доставки в БД.

## 9. Метод **GetOpenSending**

Метод **GetOpenSending** позволяет получить данные об открытых отправлениях, в которые можно добавлять заказы для отгрузки.

Для обращения к методу следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/GetOpenSending?user=usrLogin&password=usrPassword&responseFormat=0>

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.
- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.

### 9.1 Возвращаемые данные

Если у пользователя нет открытых отправок, в которые можно добавить заказы, возвращается ответ «Открытые отгрузки не найдены».

Если есть открытые отправки, в которые можно добавить заказы, возвращается один или много тэгов **<LogCars>**, которые содержат:

- **LogCarsId** – внутренний номер отправки.
- **DeliveryAddressName** – адрес доставки.
- **DeliveryAddressId** – внутренний идентификатор адреса доставки в БД.

## 10. Метод **ShipmentSalesOrder**

Метод **ShipmentSalesOrder** позволяет отгрузить один неотгруженный заказ.

Для обращения к методу следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/ShipmentSalesOrder?user=usrLogin&password=usrPassword&responseFormat=0&OrderId=0&LogCarsId=0&contact=0&mark=0>

Описание параметров метода:

- **user** – имя пользователя API.
- **password** – пароль пользователя API.



- **responseFormat** – формат данных, возвращаемых сервером. 0 – XML, 1 – JSON.
- **OrderId** – номер заказа.
- **LogCarsId** – номер отправки, в которую нужно добавить заказ для отгрузки, необязательный параметр.
- **contact** – ФИО получателя, здесь же можно указать телефон для связи, необязательный параметр.
- **mark** – маркировка заказа, необязательный параметр.

## 10.1 Возвращаемые данные

В зависимости от успешности выполнения запроса метод возвращает код ошибки Code. Значения:

- 200 – заказ успешно отправлен на сборку;
- 201 – заказ не найден;
- 202 – заказ не может быть отгружен по API, за отгрузкой заказа обратитесь к менеджеру;
- 203 – в заказе нет строк;
- 204 – в заказе есть артикулы с нулевой ценой, заказ не может быть отгружен по API, обратитесь к менеджеру;
- 205 – в заказе есть незарезервированные артикулы, заказ не может быть отгружен по API, обратитесь к менеджеру;
- 206 – в заказе есть позиции с необработанным запросом цены, заказ не может быть отгружен по API, обратитесь к менеджеру;
- 207 – в заказе есть артикулы Cisco/Extreme/HP, заказ не может быть отгружен по API, обратитесь к менеджеру;
- 208 – в заказе есть артикулы с неодобренной ценой, заказ не может быть отгружен по API, обратитесь к менеджеру;
- 209 – требуется ручного одобрение кредита, обратитесь к менеджеру;
- 210 – превышен кредитный лимит, обратитесь к менеджеру;
- 211 – не проставлены (не хватает) ГТД, обратитесь к менеджеру;
- 212 – заказ добавлен в перемещение, для отгрузки заказа требуется оплатить заказ и обратиться к менеджеру;
- 213 – на заказ на сумму меньше X долларов нельзя создать отгрузку, обратитесь к менеджеру;
- 214 – заказ уже отгружен;

- 215 – в указанную отгрузку нельзя добавить заказ, обратитесь к менеджеру;
- 216 – невозможна отгрузка в одном заказе артикула с разной ценой. Удалите один из артикулов для продолжения отгрузки или обратитесь к Вашему менеджеру.

Если у пользователя есть кредитная линия и заказ может быть отгружен, в результате вызова метода вернется код ошибки 200 – «заказ успешно отправлен на сборку».

В случае ошибки при отгрузке вернется соответствующая ошибка.

Если ошибку можно устранить самостоятельно, то после исправления, можно повторно отправить заказ на отгрузку с помощью метода **ShipmentSalesOrder**. Если нет, следует обратиться к менеджеру для отгрузки заказа.

## 11. Внешние артикулы

В ВТВ API у пользователя есть возможность работать с собственными идентификаторами артикулов (далее «внешние артикулы») в параметрах методов и в возвращаемых данных. Для этого нужно выполнить следующие требования:

- Проинформировать Марвел о желании работать с внешними артикулами. Данная возможность должна быть явно включена для пользователя API. По умолчанию она выключена.
- Передать ответственному менеджеру Марвел заполненную таблицу соответствия внешних артикулов и артикулов Марвел. Таблица соответствия номенклатурных кодов на сегодняшний день поддерживается вручную на основании информации, полученной от партнера.

При передаче внешнего артикула в метод API, если артикул не найден в таблице соответствия, API возвращает ошибку с кодом 11 «Не найдено соответствие внешнему артикулу». В данном случае необходимо связаться с ответственным менеджером Марвел для дополнения Таблицы соответствия артикулов, либо использовать номенклатурный код Марвел.

### 11.1 Расширенная информация, возвращаемая методами B2B API при использовании внешних артикулов

#### 11.1.1 Тэг <ExternalItemId />

Во все методы B2B API, возвращающие информацию об артикуле, добавляется тэг <ExternalItemId />. Этот тэг содержит внешний (пользовательский) идентификатор артикула. Если соответствие артикулу Марвел не найдено, тэг пустой.

```
<CategoryItem>
  <WareArticle>FD-BLACK</WareArticle>
  <ExternalItemId>10008348</ExternalItemId>
  <WarePackStatus>BCKPыТАЯ</WarePackStatus>
  <Dimension>осн</Dimension>
  <WareVendor>ADVOCAM</WareVendor>
```

```

<WarePrice>0,000000</WarePrice>
<WarePriceCurrency></WarePriceCurrency>
<RRPrice />
<RRPriceCurrency />
<TotalInventQty>2</TotalInventQty>
<AvailableForB2BOrderQty>2</AvailableForB2BOrderQty>
<InNearTransitCount>0</InNearTransitCount>
<InFarTransitCount>0</InFarTransitCount>
<NetWeight />
<Weight />
<UnitVolume>0.001548450000</UnitVolume>
<Width>37.000000000000</Width>
<Height>27.000000000000</Height>
<Depth>31.000000000000</Depth>
<TaxPackagingCount>20</TaxPackagingCount>
<PurchaseQty />
<WareFullName>Видеокамера ADVOCAM Профессиональный автомобильный видеорегистратор
FD-BLACK</WareFullName>
<AvailableForShippingInMSKCount>2</AvailableForShippingInMSKCount>
<AvailableForShippingInSPBCount>0</AvailableForShippingInSPBCount>
<WarePriceUSD>0,000000</WarePriceUSD>
<WarePriceRUB>0,000000</WarePriceRUB>
<CanBeOrdered>>true</CanBeOrdered>
<APIReservedQty>0</APIReservedQty>
<CategoryId>НИнф_АвтоВидРегистр</CategoryId>
<CategoryName>Автомобильные видеорегистраторы</CategoryName>
<PromoDescription />
<PromoURL />
</CategoryItem>

```

Пример 26

Тэг `<ExternalItemId />` добавляется для каждого артикула в возвращаемые данные методов:

- `GetFullStock`
- `GetStock`
- `CheckReserveStatus`
- `GetItems`
- `GetItemPhotos`
- `CreateReserve`
- `ModifyReserve`
- `DeleteReserve`
- `TransferReserve`
- `CreateSalesOrder`
- `CheckSalesOrderItemStatus`.

### 11.1.2 Тэги `<ExternalItemIdError />` и `<ExternalItemIdErrorCode />`

Если метод B2b API принимает на вход список артикулов, то информация о каждом артикуле расширяется еще на два тэга: `<ExternalItemIdError />` и

<ExternalItemIdErrorCode />. Эти тэги содержат информацию о возможных ошибках при поиске соответствия внешнего артикула и артикула Марвел.

```
<CategoryItem>
  <WareArticle>HDR-ST1003</WareArticle>
  <ExternalItemId>10007753</ExternalItemId>
  <ExternalItemIdError>Артикул недоступен в B2B</ExternalItemIdError>
  <ExternalItemIdErrorCode>12</ExternalItemIdErrorCode>
  <WarePackStatus />
  <Dimension />
  <WareVendor />
  <WarePrice />
  <WarePriceCurrency />
  <RRPrice />
  <RRPriceCurrency />
  <TotalInventQty />
  <AvailableForB2BOrderQty />
  <InNearTransitCount />
  <InFarTransitCount />
  <NetWeight />
  <Weight />
  <UnitVolume />
  <Width />
  <Height />
  <Depth />
  <TaxPackagingCount />
  <PurchaseQty />
  <WareFullName />
  <AvailableForShippingInMSKCount />
  <AvailableForShippingInSPBCount />
  <WarePriceUSD />
  <WarePriceRUB />
  <CanBeOrdered>>false</CanBeOrdered>
  <APIReservedQty />
  <PromoDescription />
  <PromoURL />
</CategoryItem>
```

#### Пример 27

Если ошибок нет, то тэги <ExternalItemIdError /> и <ExternalItemIdErrorCode /> остаются пустыми.

Тэги <ExternalItemIdError /> и <ExternalItemIdErrorCode /> добавляются для каждого артикула в возвращаемые данные методов:

- GetItems
- GetItemPhotos
- CreateReserve
- ModifyReserve
- DeleteReserve
- TransferReserve
- CreateSalesOrder
- CheckSalesOrderItemStatus.

## 11.2 Использование внешних артикулов при обращении к методам API

Методы B2B API:

- GetItems
- GetItemPhotos
- CreateReserve
- ModifyReserve
- DeleteReserve
- TransferReserve
- CreateSalesOrder
- CheckSalesOrderItemStatus

принимают на вход списки артикулов для получения расширенной информации об артикуле, изображений артикула, резервирования артикула и создания заказа. Во всех этих методах могут быть использованы внешние артикулы.

Для этого во входном XML (или JSON) документе (см. описание соответствующих методов) следует вместо тэга <ItemId /> использовать тэг <ExternalItemId />.

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <WareItem>
    <ExternalItemId>10007753</ExternalItemId>
  </WareItem>
  <WareItem>
    <ItemId>JBLHORIZONBLKEU</ItemId>
  </WareItem>
  <WareItem>
    <ItemId>JBLHORIZONBLK1U</ItemId>
  </WareItem>
  <WareItem>
    <ItemId>E719SD</ItemId>
  </WareItem>
  <WareItem>
    <ExternalItemId>10008347</ExternalItemId>
  </WareItem>
</Root>
```

Пример 28

Пример 28 представляет фрагмент обращения к методу GetItems с использованием внешних артикулов.

Как видно из примера, в одном обращении к методу можно использовать как внешние артикулы, так и артикулы Марвел, однако одновременное использование тэгов <ItemId /> и <ExternalItemId /> для одного и того же артикула запрещено и приводит к ошибке.

## 11.3 Сообщения об ошибках, возникающие при использовании внешних артикулов

**Код ошибки 10 – «Некорректное использование внешнего артикула в параметрах»**

В запросе для одного и того же артикула присутствуют оба тэга <ItemId /> и <ExternallItemId />, что недопустимо.

#### **Код ошибки 11 – «Не найдено соответствие внешнему артикулу»**

В таблице соответствия артикулов отсутствует информация о внешнем артикуле, указанном в тэге <ExternallItemId />. Для прояснения статуса такого артикула необходимо связаться с менеджером Марвел.

#### **Код ошибки 12 – «Артикул недоступен в B2B»**

Информация в таблице соответствия найдена, однако артикул Марвел по различным причинам недоступен для резервирования, заказа и иных операций. Для прояснения статуса такого артикула необходимо связаться с менеджером Марвел.

## **12. Метод API Logout**

Для вызова метода API Logout следует сформировать URL (POST запрос):

<https://b2b.marvel.ru/Api/Logout?user=usrLogin&password=usrPassword>

Иногда по причине внутренних ошибок сервера или перегрузки сервера метод API выполняется неуспешно и возвращает сообщение об ошибке. Получив сообщение об ошибке, рекомендуем вызвать метод Logout, а потом повторить запрос.

## **13. Сценарий использования данных, возвращаемых B2B API**

- Получить и сохранить локально структуру категорий, используя метод GetCatalogCategories. Этот метод возвращает дерево категорий.
- Получить и сохранить локально каталог, используя метод GetFullStock. Этот метод возвращает все артикулы, доступные в каталоге, с указанием к какой из категорий относится артикул.
- Используя идентификатор категории как ключ, можно получать выборки артикулов для той или иной категории из локальной копии набора данных, полученных методом GetFullStock.
- Локально сохранённые данные есть смысл обновлять не более двух-трёх раз в сутки.
- Для получения наиболее актуальных данных по интересующему артикулу (артикулам), например, перед резервированием, следует использовать метод GetItems.

## 14. Тестовые вызовы методов API

Каждый метод B2B API можно вызвать в тестовом режиме.

Для вызова метода API в тестовом режиме следует использовать логин testApiUser с паролем password1. Все остальные параметры вызываемого метода должны соответствовать правилам, описанным выше. Например:

<https://b2b.marvel.ru/Api/GetItems?User=testApiUser&password=password1&secretKey=&packStatus=0&responseFormat=0&items=PB515-D-RU;NX.MJ4ER.003;A18-7570;SEB-100;>

Параметр secretKey при тестовом вызове можно оставить пустым. Если взаимодействие с сервером в тестовом режиме завершилось успешно, то Вы получите следующий XML-документ (пример 29):

```
<Response>
  <Header>
    <Code>1</Code>
    <Message>Вы успешно протестировали вызов метода
    [GetCatalogCategories]</Message>
    <Key/>
  </Header>
  <Body/>
</Response>
```

пример 29

При вызове различных методов API в тестовом режиме в документе ответа изменяться будет только название метода (указывается в квадратных скобках).

### 14.1 Промежуточный прокси-сервер

Возможно возникновение проблем, когда между компьютером, на котором исполняются запросы к B2B API, и интернетом, находится прокси-сервер.

Мы рекомендуем проводить предварительное тестирование B2B API на компьютере, который подключен к интернету напрямую, например, с ноутбука, подключенного к интернету через сотовый модем. Убедившись, что доступ к сайту B2B в тестовом режиме работает, следует настроить прокси-сервер. Подробные рекомендации по настройке выходят за рамки данного документа, но следует отметить, что для успешной работы с API на прокси-сервере должно быть реализовано правило, позволяющее пропускать авторизацию для URL.

### 14.2 Использование плагина Restlet Client – REST API Testing

Все приведённые выше URL-примеры НЕЛЬЗЯ проверить из адресной строки браузера. Потому что браузер отправляет GET запросы, а наше API ожидает от пользователей POST запросы. Однако для тестирования API можно использовать браузер Google Chrome с установленным расширением «Restlet Client – REST API Testing».

Введите тестовый URL (например, <https://b2b.marvel.ru/Api/GetCatalogCategories?user=testApiUser&password=userPassword&secretKey=&responseFormat=0>) в поле «type an URL» (рисунок 1).

Для вызова метода нажмите на кнопку, отмеченную на рисунке 1. Вы увидите ответ сервера API для тестового пользователя (рисунок 2).

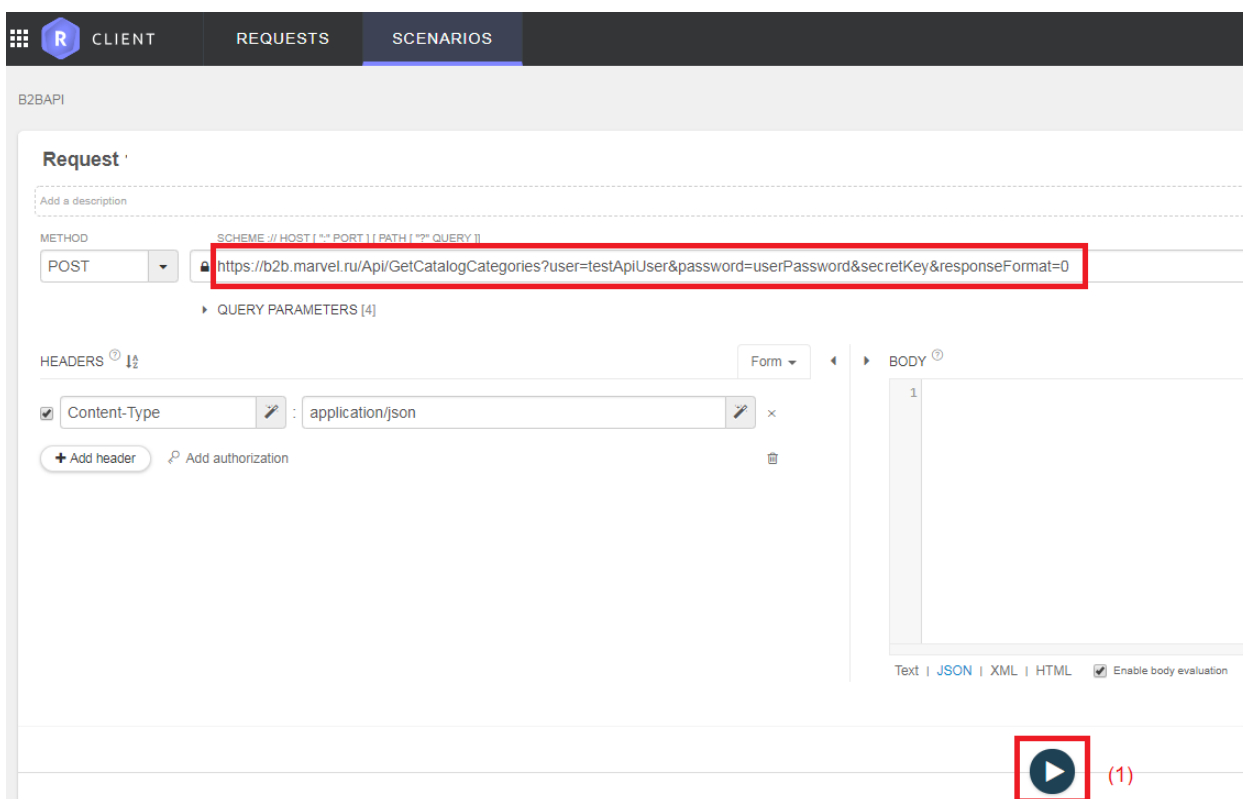


Рисунок 1

```
<?xml version="1.0" encoding="utf-8"?>
<Response>
  <Header>
    <Code>1</Code>
    <Message>Вы успешно вызвали протестировали вызов метода [GetCatalogCategories]</Message>
    <Key></Key>
  </Header>
  <Body >
</Response>
```

Рисунок 2

При условии, что указан реальный пользователь и корректный пароль, сервер вернёт реальные данные.