



**MONT Webstore API**

**Documentation**

**Руководство для разработчиков**

**Версия 3.0.7**

**1.04.2016**

# История изменений

## Изменения в версии 2.0.1 от 3.03.2014:

1. Дополнены описания методов GetAllProducts(bool withPrice) и GetProductPrices()
2. В полях UrlCoverImage и ImageUrls в FamilyGroupContract и ProductContract теперь возвращается абсолютный путь до изображения.
3. В ProductContract добавлены 2 новых поля: EndUserPrice и EndUserCurrencyCode

## Изменения в версии 2.0.2 от 19.03.2014:

1. Поправлены и дополнены коды ошибок на стр. 28
2. Добавлены примеры вызова каждого метода из контент сервера в разделе “Доступные методы для работы в сервисе MONT Content.”
3. Название 5-го метода исправлено с “5.GetVendorFamilyGroups(int vendorId)” на “5.GetAllFamilyGroups(

## Изменения в версии 2.0.4 от 28.05.2014

1. Исправлено название метода GetCategoriesFamilyGroup
2. Расширено описание ProductContract
3. Расширено описание методов GetAllChanges, GetContentChanges, GetPriceChanges
4. Поля ProductContract и ProductTranslationContract теперь содержат информацию об акциях

## Изменения в версии 2.0.4.1 от 07.07.2014

1. Исправление опечаток

## Изменения в версии 2.0.4.2 от 08.08.2014

1. Исправление опечаток

## Изменения в версии 3.0.1 от 25.02.2015

1. Полностью изменена документация по MONT Content в связи с выходом 3-й версии
2. Добавлено поле ActualVersion и параметр sinceVersion в методе GetPriceChanges сервиса MONT eKey

### Изменения в версии 3.0.2 от 02.04.2015

1. Добавлено поле OrderNum в FamilyContract и ProductContract, по которому предлагается сортировать продукты внутри семейства и семейства внутри вендора

### Изменения в версии 3.0.3 от 21.04.2015

1. Добавлен метод GetAllVendors в сервис MONT Content
2. Исправлена ошибка наименования поля IsLicence
3. Скорректирован раздел содержание.

### Изменения в версии 3.0.4 от 04.06.2015

1. Исправлены тексты сообщений об ошибках

Код ошибки	Старое название	Новое название
2	Товар не найден.	Продукт не найден
3	Нет заказа с таким номером.	Заказ не найден
4	Заказ с указанным номером уже существует в системе.	Такой номер заказа уже есть в системе
10	Данный товар заблокирован.	Продукт заблокирован
12	Необходимые данные о конечном пользователе не были переданы	Лицензионная форма не передана
16	Клиент заблокирован	Реселлер заблокирован
17	Заказанное количество не попадает в диапазон, указанный в артикуле	Количество задано не верно

2. Исправлен тип результата при возвращении ошибки: теперь Errors - это список ошибок (а не информация об одной ошибке), каждая из которых может содержать несколько элементов PurchaseError.

### Изменения в версии 3.0.5 от 1.04.2016

1. <https://sandbox.webstore.mont.ru/> заменен на <https://sandbox.mont.ru/> Ссылка указывает на тот же ресурс тестовой среды. Но для нее есть валидный сертификат. Предыдущая ссылка на ресурс будет поддерживаться и продолжать работать.
2. Исправлены ошибки в xml примерах (убраны лишние символы в теле xml запросов, которые в Xml over http привязках могли привести к ошибке). Utf-16 в заголовках, которая могла привести к ошибке, исправлена на utf-8.

# Содержание

## MONT Webstore

### Сервис MONT Content

1. [Адрес сервиса](#)
2. [Сущности, которыми оперирует сервис](#)
3. [Примеры использования](#)
4. [Общая структура возвращаемых значений](#)
5. [Описание работы с системой](#)
  - 5.1. [GetMenus](#)
  - 5.2. [GetAvaliableLanguages](#)
  - 5.3. [GetMenu/GetJsonMenu](#)
  - 5.4. [GetDefaultMenu](#)
  - 5.5. [GetAvaliableCountries](#)
  - 5.6. [GetFamily/GetJsonFamily](#)
  - 5.7. [Алгоритм удаления пустых веток меню.](#)
  - 5.8. [GetProduct/GetJsonProduct](#)
  - 5.9. [GetChanges](#)
  - 5.10. [TestConnection](#)
  - 5.11. [GetAllVendors](#)

### Сервис MONT eKey

1. [Примеры использования](#)
2. [Работа с заказами](#)
3. [Работа с лицензионными формами](#)
4. [Размещение заказа](#)
5. [Получение ключей](#)
6. [Пример заказа с файлом](#)
7. [Ожидание результата](#)
8. [Обработка ошибок](#)
9. [Получение справочной информации](#)

# MONT Webstore

Система MONT Webstore предназначена для интеграции с электронными торговыми системами по продаже лицензионного ПО посредством электронных каналов дистрибуции.

Целью системы является предоставление возможности авторизованным партнерам группы компаний MONT получать актуальную информацию об электронных продуктах, доступных через каналы электронной поставки ПО, размещать заказы и получать ключи напрямую от вендора.

Система MONT Webstore является собственной разработкой группы компаний MONT и ей же принадлежит.

Система представляет собой два веб сервиса - MONT Content и MONT eKey.

Доступ в систему осуществляется посредством HTTPS протокола либо с помощью SOAP привязки с negotiation авторизацией (для C# клиентов) либо с помощью XmlOverHttp(s) привязки с Basic аторизацией

## **Сервис MONT Content решает следующие задачи:**

Предоставляет необходимую информацию о продуктах, включающую:

- описание продукта
- системные требования к продукту
- условия лицензионного соглашения и деталей использования
- инструкцию по установке и активации, если это требуется
- дополнительные характеристики продукта

Важно! Вся приведенная выше информация может предоставляться на различных языках.

## **Сервис MONT eKey решает следующие задачи:**

1. Предоставляет информацию, необходимую для размещения заказов:

- цену для партнера
- валюту
- признак о возможности покупать продукт
- дополнительную информацию, необходимую для заказа. Например, нужна ли заполненная лицензионная форма при заказе данного продукта, нужно ли

обновление ссылок через определенный промежуток времени после получения результата заказа и т.д.

2. Размещение заказа,
3. Отслеживание изменений статуса заказа,
4. Получение результатов выполнения заказа и, если возникнет необходимость, повторный запрос результатов выполнения заказа.

# Сервис MONT Content

## 1. Адрес сервиса

### Тестовый доступ:

SOAP привязка с negotiation авторизацией (для C# клиентов):

<https://sandbox.mont.ru/ContentApi/Version3/Service/ContentServiceV3.svc>

XmlOverHttp(s) привязка с Basic аторизацией:

<https://sandbox.mont.ru/ContentApi/Version3/Service/ContentServiceV3Xml.svc>

### Продакшн:

SOAP привязка с negotiation авторизацией (для C# клиентов):

<https://webstore.mont.ru/ContentApi/Version3/Service/ContentServiceV3.svc>

XmlOverHttp(s) привязка с Basic аторизацией:

<https://webstore.mont.ru/ContentApi/Version3/Service/ContentServiceV3Xml.svc>

## 2. Сущности, которыми оперирует сервис

### Категория

Категории могут иметь древовидную структуру. Могут использоваться на стороне реселлера для группировки продуктов в логические блоки, например для построения меню на сайте или тематической фильтрации. Категории принадлежат одному или нескольким вендорам и содержат продукты и семейства.

### Семейство

Логическая сущность введенная для группировки конкретных товаров в понятные конечному пользователю продукты.

Например, имеется два продукта:

- LHW-AK-12M-1-A3 - Антивирус Dr.Web, на 12 мес., 1 лиц
- LHW-AK-12M-2-A3 - Антивирус Dr.Web, на 12 мес., 2 лиц

Они являются по сути одним и тем же программным продуктом - Антивирус Dr.Web, поэтому объединяются в одно семейство - Антивирус Dr.Web. При этом оно будет содержать два, связанных с ним продукта: [LHW-AK-12M-1-A3](#) и [LHW-AK-12M-2-A3](#)

## Продукт

Данная сущность может содержать в себе (в зависимости от того, каким методом она получена) полное описание продукта на различных языках. Важно отметить, что большинство продуктов в своем описании содержат список характеристик вида: название характеристики, ее значение и тип. Например, «Тип лицензии» - «Базовая» - «строка». Данные характеристики так же можно использовать для группировки продуктов и семейств при разработке собственных систем.

Клиент сервиса в праве сам решать использовать ему дополнительные сущности – категорию и семейство в своей системе или нет.

## 3. Примеры использования

Работать с сервисом можно, используя и SOAP и HTTP методы.

Скачать пример работы с сервисом MONT Content с помощью HTTP методов можно [здесь](#):

<http://sandbox.webstore.mont.ru/files/ContentApiPhpSample.zip>

Пример работы с сервисом MONT Content с помощью SOAP методов:

<http://sandbox.webstore.mont.ru/files/ContentSOAPEXample.rar>

## 4. Общая структура возвращаемых значений

Каждый ответ от сервера содержит обязательных 3 элемента:

Тег в XML	Значение
Error	Наличие ошибки (true/false)
ErrorDetails	Если во время запроса произошла ошибка, содержит ErrorContract с информацией об ошибке, пустой в случае удачного выполнения запроса
Зависит от ответа сервера	Содержит полезные данные результата запроса

Такую структуру можно представить в общем виде в XML:

```
<SomeResponse
xmlns="http://schemas.datacontract.org/2004/07/ContentServer.WebService.Version2.Co
ntracts.Responses" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Error>false</Error>
  <ErrorDetails i:nil="true"
xmlns:a="http://schemas.datacontract.org/2004/07/ContentServer.WebService.Version2.
Contracts" />
  <ServerResponseData>
    ответ сервера
  </ServerResponseData>
</SomeResponse>
```

Из таблицы видно, что ответ на все методы имеет общий вид. Отличается лишь элемент, содержащий результат выполнения метода на сервере. Поэтому, для простоты, далее будем рассматривать упрощенный вид данной структуры.

Важно! При передаче параметров типа Boolean нужно использовать значения true/false. Значения 0/1 не будут распознаны сервисом.

## 5. Описание работы с системой

Данный API позволяет получать описание продуктов, семейств (групп продуктов) и деревьев категорий, в которые они собраны. Вы можете использовать всю предоставляемую информацию по своему усмотрению. Мы же рассмотрим использование сервиса MONT Content для отрисовки витрины интернет-магазина и на этом примере опишем все методы.

### 5.1 GetMenus

В первую очередь нам понадобится дерево-каталог, на основании которого можно нарисовать меню. Мы детально проработали и предоставляем всем базовое меню. Оно доступно каждому пользователю. При необходимости мы можем добавить персональные меню для каждого пользователя сервиса.

Итак, начинаем с получения списка доступных меню.

Пример вызова. Get: /GetMenus

Требует авторизации

Результатом вызова метода станет массив элементов **MenuDescriptionContract**:

```
<ArrayOfMenuDescriptionContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <MenuDescriptionContract>
    <Id>1</Id>
    <MenuName>Legacy Tree</MenuName>
    <Version>2165460</Version>
  </MenuDescriptionContract>
```

```
<MenuDescriptionContract>
  <Id>2</Id>
  <MenuName>Legacy 2</MenuName>
  <Version>1201654</Version>
</MenuDescriptionContract>
</ArrayOfMenuDescriptionContract>
```

Каждый элемент **MenuDescriptionContract** содержит следующие поля:

Название	Значение
Id	Id меню
MenuName	Название меню
MenuDescription	Описание меню
Version	Версия меню

Данный метод можно вызвать один раз, только для получения ID нужного меню. Особенно в том случае, если вам доступно только 1 меню.

## 5.2 GetAvaliableLanguages

Но ID меню - это еще не все. Для получения самого меню нужно еще определиться с языком интерфейса.

Пример вызова: Get : /GetAvaliableLanguages

Результатом станет список доступных кодов языков

## 5.3 GetMenu/GetJsonMenu

ID меню у нас есть, язык тоже - можем приступить к получению самого меню.

Пример вызова: GET: /GetMenu?id=1&langcode=RU

Требует авторизации

В результате вернется структура **MenuContract** вида:

```
<?xml version="1.0" encoding="utf-8"?>
<MenuContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Version>2014684</Version>
  <MenuElements>
    <MenuElementContract>
      <Title>Microsoft</Title>
      <Level>1</Level>
      <Id>1</Id>
      <ParentId xsi:nil="true" />
      <OrderIndex>0</OrderIndex>
      <FamilyGroupIds />
    </MenuElementContract>
    <MenuElementContract>
      <Title>Операционные системы</Title>
      <Level>2</Level>
      <Id>3</Id>
      <ParentId>1</ParentId>
      <OrderIndex>0</OrderIndex>
      <FamilyGroupIds />
    </MenuElementContract>
    <MenuElementContract>
      <Title>Windows 8</Title>
      <Level>3</Level>
      <Id>4</Id>
      <ParentId>3</ParentId>
      <OrderIndex>0</OrderIndex>
      <FamilyGroupIds>
        <int>43</int>
        <int>47</int>
      </FamilyGroupIds>
    </MenuElementContract>
    <MenuElementContract>
      <Title>Распознавание текста</Title>
      <Level>2</Level>
      <Id>13</Id>
      <ParentId>2</ParentId>
      <OrderIndex>0</OrderIndex>
      <FamilyGroupIds>
        <int>97</int>
        <int>98</int>
      </FamilyGroupIds>
    </MenuElementContract>
  </MenuElements>
</MenuContract>
```

```

</FamilyGroupIds>
</MenuElementContract>
.....
</MenuElements>
</MenuContract>

```

## Структура **MenuContract**:

Название	Значение
Version	версия меню
MenuElements	Массив элементов меню MenuElementContract

Каждый элемент **MenuElementContract** состоит из следующих полей:

Название	Значение
Title	Название пункта меню
Level	Уровень вложенности
Id	Id элемента
ParentId	Id родителя
OrderIndex	Номер по порядку (необязательно)
FamilyGroupIds	Коды закрепленных групп товаров (иначе, идентификаторы семейств, относящихся к данной категории)

## 5.4 GetDefaultMenu/GetDefaultJsonMenu

Если у пользователя всего одно меню, то можно воспользоваться упрощенным способом его получения.

Пример вызова: `Get: /GetDefaultMenu?langcode=RU.`

Результат вызова аналогичен результату вызова метода `GetMenu`

## 5.5 GetAvaliableCountries

Если мы ограничимся только прорисовкой красивой витрины, то нам будет достаточно воспользоваться методом **GetFamily**, который вернет нам подробное описание каждого семейства. Бывают случаи, когда одно и то же семейство имеет разное описание в разных странах, поэтому нам потребуется определиться со страной.

Список возможных стран можно получить методом `GetAvaliableCountries`.

Пример вызова: `Get : /GetAvaliableCountries`

Результатом будет список кодов стран.

**Важно:** стоит отметить тот факт, что понятие страны введено для определенных товаров, которые могут иметь в своем описании(к тому же на разных языках) фактически различия от страны к стране. Так как таких товаров немного, то наиболее распространенным кодом страны для всех переводов является код DF - Default. Это значит, что для товара, обладающим таким кодом страны все переводы на различные языки будут одинаковы вне зависимости от страны, за исключением отдельного описания товара с кодом страны не равным DF. Например: есть описание товара ABC-1111 на RU языке для DF страны и описание для этого же товара на RU языке для UA страны(отличающееся фактически от DF страны). Все запросы вернуть описание на RU языке для товара ABC-1111 вернут описание RU-DF. Напрмер для страны KZ и языка RU - вернется описание RU-DF. Однако для страны UA и языка RU вернется описание RU-UA.

## 5.6 GetFamily/GetJsonFamily

Теперь все готово для получения описания семейства.

Пример вызова: `Get: /GetFamily?id=123&langcode=RU&countrycode=RU`

Метод используется для получения описания семейства (товарной группы) на заданном языке, для заданной страны.

Результатом будет структура **FamilyContract** вида:

```
<FamilyContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Title>G Data AntiVirus</Title>
  <UrlImage>https://webstore.mont.ru/ContentData/</UrlImage>
  <Id>162</Id>
  <VendorId>0</VendorId>
  <OrderNum>0</OrderNum>
  <Preview>Мощная защита от вирусов, троянских и других вредоносных
программ.</Preview>
  <ShortDescription>G Data AntiVirus 2014 обеспечивает самую лучшую оперативную
защиту от вирусов и вредоносных программ с помощью комбинации из двух
антивирусных движков, облачных технологий и модуля контроля
системы.</ShortDescription>
  <FullDescription>Защитные модули очень просты в настройке, работают
незаметно в фоновом режиме и не замедляют систему.&lt;br /&gt;&lt;br /&gt;
- &lt;i&gt;Мгновенная интернет-защита:&lt;/i&gt; G Data Web Cloud блокирует
опасные сайты&lt;br /&gt;&lt;br /&gt;
- &lt;i&gt;Мгновенная защита почты:&lt;/i&gt; G Data Mail Cloud блокирует вирусные
эпидемии&lt;br /&gt;&lt;br /&gt;
.....
  </FullDescription>
  <SystemRequirements>&lt;p&gt;&lt;h4&gt;Системные требования&lt;/h4&gt;&lt;/p&gt;
&lt;p&gt;(32/64 бита): Windows 7 / Vista, мин. 1 GB ОЗУ, DVD-дисковод&lt;/p&gt;
&lt;p&gt;32 бита): Windows XP (от SP2), мин. 512 MB ОЗУ,
DVD-дисковод&lt;/p&gt;</SystemRequirements>
  <InstallInfo>Скачайте дистрибутив программы &lt;a
href="https://www.gdatasoftware.com/rdk/dl-ru-av/" target="_blank"&gt;G DATA
AntiVirus&lt;/a&gt;&lt;br /&gt;&lt;br /&gt;
```

2. Дважды нажмите мышкой на установочном файле, чтобы запустить процесс установки.<br /><br />

```
.....  
</InstallInfo>  
<Specs />  
</FamilyContract>
```

**FamilyContract** может содержать следующие поля:

Название	Значение
Id	Id семейства (товарной группы)
Title	Наименование
UrlImage	Адрес картинки
VendorId	Id вендора
Preview	Анонс
ShortDescription	Краткое описание
FullDescription	Подробное описание
ImportantInformation	Важная информация
SystemRequirements	Системные требования
InstallInfo	Инструкция по установке
EventText	Акция
Specs	Массив характеристик

OrderNum	Поле, по которому рекомендуется сортировать список семейств внутри вендора
----------	--

Каждая характеристика описывается структурой **SpecContract**:

Название	Значение
Id	Id характеристики
Title	Название
Value	Значение
SpecType	Тип, где 1 - строка, 2 - число, 3- истина

**Важно:** Многие поля могут содержать html форматирование. При этом все символы тегов "<>" заменены на соответствующие последовательности символов "&lt;" и "&gt;". Для того, чтобы тэги "заработали" у нас на витрине нужно сделать обратную замену.

Базовое меню является общим для всех пользователей и основано на полном каталоге всех продуктов. В том числе и архивных. В таком виде оно годится только для создания библиотеки. А для интернет-магазина весь этот архив мало интересен. На витрине должны быть только те семейства, в которых есть хотя бы 1 продукт **доступный к продаже**. К тому же уменьшение числа ветвей в меню сильно ускорит его отрисовку. Поэтому мы пока не будем торопиться с получением детального описания семейств, а перейдем к следующему шагу - это получение продуктов доступных к продаже. Для этого используем метод **GetPriceChanges(\_SOAP)** из сервиса MONT eKey. Подробно он описан в соответствующем разделе документации.

## 5.7 Алгоритм удаления пустых веток меню.

Теперь можем исключить из меню те ветви, в которых нет ни одного продукта. В зависимости от реселлера, доступности ему различных товаров в прайс листе, возникает необходимость удаления (сокрытия) элементов меню, в которых нет ни одного доступного для продажи товара. Предлагаемый ниже алгоритм является одним из вариантов реализации этой задачи.

Используя список товаров, имеющих FamilyGroupId (из метода **GetPriceChanges**) и список элементов меню, каждый из которых имеет свойство Level (из метода **GetMenu**) можно довольно просто построить меню доступных товаров. Для этого :

1. Находим максимальный уровень вложенности меню по полю Level.
2. В цикле от нижнего уровня меню к верхнему проверяем каждое семейство узла меню (все Id из поля FamilyGroupIds) на наличие его в списке продуктов:
  - A. с признаком не блокирован,
  - B. с ценой больше 0,
  - C. в нужной валюте.
3. Если таких продуктов нет, то помечаем сам узел меню “на удаление”.
4. Если FamilyGroupIds для узла является пустым, то ищем хотя бы одного потомка данного узла, не помеченного “на удаление”. Если таких потомков не найдено, помечаем узел меню “на удаление”.
5. Удаляем все узлы, помеченные “на удаление”.

## 5.8 GetProduct/GetJsonProduct

А вот теперь, когда мы получили реальное меню, самое время воспользоваться описанным ранее методом **GetFamily** для получения описания семейств и **GetProduct** для получения описания продуктов.

Пример вызова: GET: /GetProduct?partnum=ABC-11111&langcode=RU&countrycode=RU

Данный метод используется для получения полного описания товара, с его характеристиками на запрошенном языке для заданной страны. Метод GetJsonProduct возвращает описание в JSON формате.

Результатом вызова будет структура **ProductContract** вида:

```
<?xml version="1.0" encoding="utf-8"?>
<ProductContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PartNumber>10011</PartNumber>
  <Title>G Data Antivirus 1 год 1 ПК  </Title>
  <UrlImage>https://webstore.mont.ru/ContentData/</UrlImage>
  <FamilyGroupId xsi:nil="true" />
  <VendorId>0</VendorId>
  <Preview>Антивирус GData</Preview>
  <ShortDescription>Антивирус от немецкого производителя
</ShortDescription>
  <ImportantInformation>Только для домашнего использования
</ImportantInformation>
  <SystemRequirements>86x Compatible</SystemRequirements>
  <InstallInfo>Скачать и установить, следуя инструкциям</InstallInfo>
  <OrderNum>0</OrderNum>
  <Specs>
    <SpecContract>
      <Id>16</Id>
      <Title>Тип лицензии</Title>
      <Value>Базовая</Value>
      <SpecType>String</SpecType>
    </SpecContract>
  </Specs>
</ProductContract>
```

**ProductContract** может содержать следующие поля:

Название	Значение
PartNumber	Партномер
Title	Наименование
UrlImage	Адрес картинки
FamilyGroupId	Код семейства (товарной группы)

VendorId	Id вендора
Preview	Анонс
ShortDescription	Краткое описание
FullDescription	Подробное описание
ImportantInformation	Важная информация
SystemRequirements	Системные требования
InstallInfo	Инструкция по установке
EventText	Акция
Specs	Массив характеристик
OrderNum	Поле, по которому рекомендуется сортировать список продуктов

Каждая характеристика описывается структурой **SpecContract**:

Название	Значение
Id	Id характеристики
Title	Название
Value	Значение
SpecType	Тип, где 1 - строка, 2 - число, 3- истина

Наша витрина готова. У нас есть меню-дерево и полное описание к каждому узлу и элементу дерева. При этом мы имеем актуальную на 100% онлайн информацию о продаваемых продуктах.

## 5.9 GetChanges

Для пользователей, которые хотят хранить описания товаров и семейств “у себя” предусмотрен метод **GetChanges**, позволяющий, после первоначальной загрузки всех описаний в собственную систему, отслеживать какие товары/семейства были изменены на каком языке и для какой страны.

Данный метод сожержит, помимо списка партномеров/семеств с кодами стран и переводов, номер версии каталога, актуальной на момент вызова процедуры. Этот номер должен использоваться в дальнейшем при вызове метода GetChanges для получения списка изменений.

Пример использования:

1. Первый вызов следует произвести с входным параметром sinceVersion = 0. В результате будут получены все имеющиеся партномера и коды семейств, а так же текущий номер версии контента в системе Mont Content.
2. Перебором, вызывая метод GetProduct/GetFamily можно сохранить у себя все интересующие описания. Например все товары, которые присутствуют в прайс листе.
3. Примерно раз в день можно производить вызов метода GetChanges с версией, полученной при предыдущем вызове для получения списка измененных товаров/семейств. Интересующие описания следует обновить в своей системе.

Пример вызова: `Get: GetChanges?sinceVersion=204145`

В ответ придет структура вида :

```
<ChangesContract xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <FamilyGroupChanges>
    <ChangedFamilyContract>
      <FamilyGroupId>102</FamilyGroupId>
      <LangCode>RU</LangCode>
      <CountryCode>DF</CountryCode>
    </ChangedFamilyContract>
    <ChangedFamilyContract>
```

```

    <FamilyGroupId>43</FamilyGroupId>
    <LangCode>RU</LangCode>
    <CountryCode>DF</CountryCode>
  </ChangedFamilyContract>
</FamilyGroupChanges>
<ProductChanges>
  <ChangedProductContract>
    <PartNum>NMPHL</PartNum>
    <LangCode>RU</LangCode>
    <CountryCode>DF</CountryCode>
  </ChangedProductContract>
</ProductChanges>
<CurrentVersion>277835</CurrentVersion>
</ChangesContract>

```

**FamilyGroupChanges** содержит массив элементов **ChangedFamilyContract**, каждый из которых

Название	Значение
FamilyGroupId	Код семейства
LangCode	Код языка
CountryCode	код страны

**ProductChanges** содержит массив элементов **ChangedProductContract**, каждый из которых

Название	Значение
PartNum	Партномер
LangCode	Код языка
CountryCode	Код страны

Также структура содержит поле CurrentVersion, указывающее текущую версию контента в системе.

### 5.10 TestConnection()

Предназначен для проверки подключения к сервису.

Пример вызова: GET: /TestConnection

### 5.11 GetAllVendors()

Предназначен для получения названия вендора и кода вендора.

Пример вызова: GET: /GetAllVendors

В ответ придет структура вида :

```
<ArrayOfVendorContract
xmlns="http://schemas.datacontract.org/2004/07/ContentServer2.WebService.Version
3.Service.Contracts" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <VendorContract>
    <Id>1</Id>
    <VendorCode>MS</VendorCode>
    <VendorName>Microsoft</VendorName>
  </VendorContract>
  <VendorContract>
    <Id>2</Id>
    <VendorCode>ABBYY</VendorCode>
    <VendorName>ABBYY</VendorName>
  </VendorContract>
</ArrayOfVendorContract>
```

# Сервис MONT eKey

## Актуальная версия сервиса 1.1 от 09 февраля 2015

### Тестовый доступ

SOAP привязка с negotiation авторизацией (для .net клиентов):

<https://sandbox.mont.ru/B2bService.svc>

XmlOverHttp(s) привязка с Basic аторизацией:

<https://sandbox.mont.ru/B2bService.svc/web/>

SOAP привязка с Basic авторизацией: <https://sandbox.mont.ru/B2bService.svc/SOAP/>

### Продакшн

SOAP привязка с negotiation авторизацией (для .net клиентов):

<https://webstore.mont.ru/B2bService.svc>

XmlOverHttp(s) привязка с Basic аторизацией:

<https://webstore.mont.ru/B2bService.svc/web/>

SOAP привязка с Basic авторизацией: <https://webstore.mont.ru/B2bService.svc/SOAP/>

## 1. Примеры использования

Пример работы с сервисом MONT WebStore, используя [XmlOverHttp](#) привязку из PHP:

<http://sandbox.webstore.mont.ru/files/phpsample.zip>

Пример работы с сервисом MONT WebStore, используя SOAP (negotiate) привязку из C#:

<http://sandbox.webstore.mont.ru/files/WebStoreSOAPEExample.zip>

### Общее описание алгоритма работы с системой

Данный API позволяет размещать заказы и получать по размещенным заказам ключи. Также сервис имеет несколько вспомогательных методов, для получения партнером информации о своем статусе в системе, и для получения прайса на доступные продукты.

Основными методами сервиса являются [AddOrder](#) и [GetOrder](#), позволяющие размещать заказы и получать информацию по размещенным заказам.

Метод [AddOrder](#) позволяет разместить заказ в системе дистрибьютора на 1 позицию (1 парт номер) на 1го конечного пользователя (или организацию). Максимальное количество штук в заказе ограничено, и определяется для каждого продукта индивидуально. (Информацию об этом может содержать любой продукт, полученный методом [GetPriceChanges\(\\_SOAP\)](#)). После успешного размещения в системе дистрибьютора заказ попадает в обработку.

Следует учитывать, что не все вендоры могут достаточно быстро обрабатывать заказы, поэтому, размещенный в электронной системе дистрибьютора заказ будет находиться некоторое время (некоторые вендора могут возвращать результат обработки заказа в течение нескольких часов) в состоянии “в процессе”. Таким образом, следует учитывать этот факт при разработке собственных систем, периодически опрашивая заказы в состоянии “в процессе” до тех пор, пока они не перейдут в состояние “завершен успешно” или “завершен с ошибкой”.

Любой успешно размещенный заказ в электронной системе дистрибьютора можно запросить позднее методом [GetOrder](#).

Используя данный метод можно получить полное описание заказа, его состояния и результатов в виде ключей, ссылок и файлов (польз. лицензий).

Особенностью данного метода является то, что для продуктов Майкрософт, содержащих параметр [FulfillmentType = 1](#) в результате вызова [GetOrder](#) приходят временные ссылки (срок действия 24 часа). Таким образом, если купленный товар не активирован по ссылке, пришедшей в ответе в течение 24 часов, то необходимо через 24 часа (или позднее по запросу пользователя) вызвать метод [GetOrder](#) еще раз. В таком случае система дистрибьютора сама обновит все ссылки на лету и вернет в результате уже актуальные на момент вызова ссылки.

## Наиболее распространенные ситуации, возникающие при размещении новых заказов:

- реселлер (клиент) заблокирован
- стоимость заказа превышает остаточный лимит реселлера.

- товар не найден либо заблокирован для продажи
- номер заказа уже существует. Клиент сам следит за уникальностью собственных номеров заказов.
- данные лицензионной формы, если такая необходима для заказа позиции, заполнены неверно.

Информация о том, нужна ли лицензионная форма, содержится в виде булевого поля ([IsLicence](#)) для каждого товара, полученного методом [GetPriceChanges\(\\_SOAP\)](#). Стоит отметить, что требования к полям самой лицензионной формы могут меняться от вендора к вендору. (Подробности см. в разделе “Работа с лицензионными формами”) В таких случаях заказ не будет размещен, при вызове метода будет возвращена ошибка с соответствующим описанием.

Для наглядности будем рассматривать привязку [XmlOverHttp](#)

## 2. Работа с заказами

Прежде чем перейти непосредственно к рассмотрению процесса размещения заказов, изучим результат вызова процедуры [GetPriceChanges\(\\_SOAP\)](#), которая возвращает прайс лист товаров с указанием дополнительных параметров, необходимых для заказа, а так же рассмотрим лицензионные формы.

Пример вызова: GET: /GetPriceChanges?sinceVersion = 2310353

Первый вызов следует произвести с входным параметром sinceVersion = 0. В результате будут получены все имеющиеся партномера и коды семейств, а так же текущий номер версии контента в системе Mont Content.

В результате вызова данного метода получим XML следующего вида:

```
<?xml version="1.0" encoding="utf-8"?>
<WebstoreProducts xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ActualDate>2015-02-24T15:06:35.1067821+03:00</ActualDate>
  <ActualVersion>7996115</ActualVersion>
  <Products>
    <WebstoreProduct>
      <PartNumber>KL2128RDEFS</PartNumber>
```

```

<Price>0</Price>
<CurrencyCode>RUR</CurrencyCode>
<Blocked>true</Blocked>
<IsLicence>true</IsLicence>
<FulfillmentType>1</FulfillmentType>
<EndUserPrice>3900</EndUserPrice>
<EndUserCurrencyCode>RUR</EndUserCurrencyCode>
<InstantDelivery>true</InstantDelivery>
<HasVat>true</HasVat>
<FamilyGroupId>56</FamilyGroupId>
<MinCountRequired>0</MinCountRequired>
<MaxCountRequired xsi:nil="true" />
</WebstoreProduct>
.....
</Products>
</WebstoreProducts>

```

Где **WebstoreProduct** элементы **Products** , каждый из которых представляет собой информацию о товаре.

Рассмотрим каждый параметр.

Тег в XML	Значение
PartNum	Номенклатура товара
Price	Цена реселлера
EndUserPrice	РРЦ
CurrencyCode	Код валюты цены реселлера
EndUserCurrencyCode	Код валюты РРЦ
FulfillmentType	Тип товара
IsLicence	Признак - нужна ли лиц. форма конечного пользователя

InstantDelivery	Признак - товар с большой вероятностью будет обработан в течении нескольких минут (при небольшом количестве штук в заказе)
HasVat	Признак - товар с НДС
Blocked	товар заблокирован для продажи
FamilyGroupld	код семейства
MinCountRequired	требуемое минимальное количество для заказа
MaxCountRequired	Требуемое максимальное количество для заказа

Поле [ActualVersion](#) показывает текущую версию цен

Значение поля [FulfillmentType](#) = 11 означает, что возвращаемый результат имеет тип "файл"

### 3. Работа с лицензионными формами.

Как уже было сказано ранее, параметр [IsLicence](#) определяет необходимость прикрепления лиц. формы о конечном пользователе к заказу.

Рассмотрим структуру данных с лицензионной формой.

```
<?xml version="1.0" encoding="utf-8"?>
<EndUserInfo xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Phone>4958880066</Phone>
  <Fax>4958880066</Fax>
  <Email>ivanpetrov@mail.com</Email>
  <CustomerName>Ivan</CustomerName>
  <CustomerMiddleName>Petrovich</CustomerMiddleName>
  <CustomerLastName>Petrov</CustomerLastName>
```

```

<Country>Russia</Country>
<Postcode>123456</Postcode>
<City>Moscow</City>
<Street>Tverskaya 48</Street>
<CustomerCode>cst09</CustomerCode>
<CustomerOrganization>Roga i Kopyta</CustomerOrganization>
<RegionCode>77</RegionCode>
<AuthorizationNumber>acv-1234</AuthorizationNumber>
<KasperskyPIN>CA0324</KasperskyPIN>
<LegalEntity>>true</LegalEntity>
</EndUserInfo>

```

Тег в XML	Описание
Phone	Телефон конечного пользователя
Fax	Факс конечного пользователя
Email	Е-mail конечного пользователя
CustomerName	Имя конечного пользователя
PostCode	Почтовый код (индекс) конечного пользователя
City	Город конечного пользователя
Street	Адрес конечного пользователя
CustomerOrganization	Наименование юр. Лица
RegionCode	Код региона
CustomerCode	ИНН физического или юридического лица

KasperskyPIN	ПИН реселлера у вендора Kaspersky Lab (для его получения необходимо запросить его, написав на почту esdTeam@mont.com)
CustomerMiddleName	Отчество
CustomerLastName	Фамилия
RegionCode	Код региона
AuthorizationNumber	Номер авторизации
IsLegalPerson	признак, что регистрируем на Юр. лицо
Country	Страна

Очевидно, что для каждого продукта требуется предоставить различный набор этих полей, а так же меняется способ их заполнения, вводятся различные ограничения.

Для того, чтобы понять каким образом должна быть заполнена та или иная форма существует метод [GetUserInfoDescriptionFor](#) , возвращающий описание и требования к каждому полю вышерассмотренной структуры в зависимости от продукта (или нескольких продуктов) и языка описания.

Для вызова метода необходимо передать методом POST по адресу <https://sandbox.mont.ru/B2bService.svc/web/GetUserInfoDescriptionFor> XML следующего вида:

```
<?xml version="1.0" encoding="utf-8"?>
<EndUserFor>
  <LangCode>RU</LangCode>
  <PartNums>
    <string>KL1030RDAFS</string>
  </PartNums>
</EndUserFor>
```

Например, вызов этого метода `GetUserInfoDescriptionFor("RU", {"KL2128RDEFS"})` вернет следующую структуру:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfEndUserInfoDescription xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <EndUserInfoDescription>
    <FieldName>Email</FieldName>
    <FieldTitle>email</FieldTitle>
    <FieldDescription>Введите email</FieldDescription>
    <RegExprs />
    <AvaliableValues />
    <HasError>>false</HasError>
    <RelatedPartNums />
  </EndUserInfoDescription>
  <EndUserInfoDescription>
    <FieldName>KasperskyPIN</FieldName>
    <FieldTitle>Пин Касперского</FieldTitle>
    <FieldDescription>Пин в системе KORM</FieldDescription>
    <RegExprs />
    <AvaliableValues />
    <HasError>>false</HasError>
    <RelatedPartNums />
  </EndUserInfoDescription>
</ArrayOfEndUserInfoDescription>
```

Где каждый элемент `EndUserInfoDescriptionModel` является описанием поля из структуры `EndUserInfo`.

Значение `FieldName` относится к узлу из структуры `EndUserInfo`. Таким образом, если поле не присутствует в описании - обязательным оно не является.

Рассмотрим узлы структуры-описания:

Тег в XML	Описание
FieldName	Название узла структуры EndUserInfo
FieldTitle	Название поля на запрошенном языке
FieldDescription	Комментарий к полю
FieldType	Тип поля. Если 1 - string, если 2 - boolean
RegExprs	Регулярное выражение, относящееся к полю
AvaliableValues	Список возможных значений данного поля
HasError	Поле не прошло проверку
ErrorText(не присутствует в примере)	Описание ошибки
RelatedPartNums	Для какого партномера актуально данное правило. Если массив пуст - то считается, что правило актуально <b>для всех</b> партномеров, переданных в запросе

Стоит отдельно сказать про последние 3 параметра. Они непосредственно связаны с методом [CheckEndUserInfo](#), который принимает на вход заполненную структуру [EndUserInfo](#), партномера(для которого происходит проверка), кода языка, и возвращает список [EndUserInfoDescriptionModel](#). Если в этом списке присутствуют элементы с признаком `HasError = true`, то форма не прошла проверку. Таким образом существует возможность проверить лиц. форму до размещения заказа.

Последний параметр (`RelatedPartNums`) указывает для какого партномера, непосредственно, актуально правило. Может сложиться ситуация, когда придется заказывать несколько различных товаров, разных вендоров, для которых нужны лицензионные формы(для каждого своя). В таком случае система вернет набор правил, которые позволят заполнить лицензионную форму 1 раз, и затем, передать заполненные формы для каждого заказа свою.

Пример: допустим, что существует 2 вендора А и Б, для товаров А1 и Б1 которых определены следующие правила заполнения лицензионных форм.

А: **поле email** - обязательно, **поле CustomerName** - должно содержать только латинские буквы

Б: **поле email** - обязательно, **поле CustomerName** - должно содержать только кириллицу

При запросе правил заполнения лиц форм для обоих товаров сразу, используя метод `GetUserInfoDescriptionFor("RU", new string[] {"A1","Б1"})` получим следующий набор правил:

1. `FieldName : Email`  
`RegExprs : ^([\w\.\-]+)@([\w\.-]+)(\.\w){2,3}+$`  
`RelatedPartNums: count = 0`
2. `FieldName: CustomerName`  
`Regexprs: ^([A-z]).`  
`RelatedPartNums: {"A1"}`
3. `FieldName: CustomerName`  
`RegExprs: ^([А-я]).`  
`RelatedPartNums: {"Б1"}`

Таким образом, в данном примере пользователю нужно заполнить 1 раз email и 2 раза Имя (1 раз на русском и 1 раз на англ языке).

Далее, при заказе партномера А1 должна уйти лицензионная форма с полями Email и CustomerName(заполнено латиницей);

при заказе партномера Б1 должна уйти лицензионная форма с полями Email и CustomerName(заполнено кириллицей);

## 4. Размещение заказа

Для размещения заказа в сервисе eKey используется метод: [AddOrder](#)

Для заказа продукта необходимо передать методом POST по адресу <https://sandbox.mont.ru/B2bService.svc/web/AddOrder> XML следующего вида:

```
<?xml version="1.0" encoding="utf-8"?>
<TransactionOrderLine>
  <OrderNum>155-6786-97912</OrderNum>
  <PartNum>T5D-00432</PartNum>
  <quantity>1</quantity>
</TransactionOrderLine>
```

Где,

Тег в XML	Значение
OrderNum	Номер заказа у реселлера. (Должен быть уникален)
PartNum	Номенклатура продукта.
quantity	Количество единиц продукта.

В случае, когда заказывается продукт, у которого значение поля **IsLicence = true**, требуется передать информацию о конечном пользователе. Для этого в XML, передаваемый методу [AddOrder](#), добавляется структура. Ее описание и пример приведены ниже.

Пример для физического лица:

```
<TransactionOrderLine xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<OrderNum>123-4567-522</OrderNum>
<quantity>1</quantity>
<endUserInfo>
  <Phone>123456</Phone>
  <Fax>654321</Fax>
  <Email>ii@mail.ru</Email>
  <CustomerName>Ivan</CustomerName>
  <CustomerLastName>Ivanov</CustomerLastName>
  <Postcode>123444</Postcode>
  <City>Moscow</City>
  <Street>Pushkinskaya 12, flat 12</Street>
  <RegionCode>77</RegionCode>
</endUserInfo>
</TransactionOrderLine>
```

Пример для юридического лица:

```
<TransactionOrderLine xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <OrderNum>123-4567-522</OrderNum>
  <PartNum>T5D-00432</PartNum>
  <quantity>1</quantity>
  <endUserInfo>
    <Phone>123456</Phone>
    <Fax>654321</Fax>
    <Email>ii@mail.ru</Email>
    <CustomerName>Ivan</CustomerName>
    <CustomerLastName>Ivanov</CustomerLastName>
    <Postcode>123444</Postcode>
    <City>Moscow</City>
    <Street>Pushkinskaya 12, flat 12</Street>
    <RegionCode>77</RegionCode>
    <CustomerCode>12345566777</CustomerCode>
    <CustomerOrganization>ООО Рога и Копыта</CustomerOrganization>
    <LegalEntity>true</LegalEntity>
```

```
</endUserInfo>
</TransactionOrderLine>
```

## 5. Получение ключей

В случае правильности заполнения всех полей и работоспособности системы результатом вызова метода `AddOrder` станет XML вида:

```
<TransactionOrderLine xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <OrderNum>155-6786-979</OrderNum>
  <DateOn>2012-11-29T16:34:46.04</DateOn>
  <quantity>1</quantity>
  <PartNum>T5D-00432</PartNum>
  <Keys>
    <ProductKey>
      <ActiveKey>65451-51499-92851-19639-61472</ActiveKey>
      <DateAdd>2012-11-29T16:34:47.9594557+04:00</DateAdd>
    </ProductKey>
  </Keys>
  <VendorClientTransactionId>60a95697-2644-4f60-aca7-44c3933fb16f</VendorClientTransactionId>
  <keyTypeId>3</keyTypeId>
  <SequenceNumber>80107651820832040427</SequenceNumber>
  <Description>Product activation code.</Description>
</ProductKey>
</Keys>
<Links>
  <ProductLink>
    <Link>http://posa.vo.msecnd.net/db/Office2010_English_x32.exe?e=1354278887&h=48996ff787513b395b2b4022e73c4014</Link>
    <DateAdd>2012-11-29T16:34:47.9674565+04:00</DateAdd>
  </ProductLink>
</Links>
  <VendorClientTransactionId>60a95697-2644-4f60-aca7-44c3933fb16f</VendorClientTransactionId>
```

```

    <linkTypeld>1</linkTypeld>
    <Description>Download link for a 32 bit Microsoft software product (large setup file/s
or ISO).</Description>
  </ProductLink>
  <ProductLink>
<Link>http://posa.vo.msecnd.net/db/Office2010_English_x64.exe?e=1354278887&h
=3eea8462ae94b8296bd548db2eda222b</Link>
    <DateAdd>2012-11-29T16:34:47.9704568+04:00</DateAdd>

<VendorClientTransactionId>60a95697-2644-4f60-aca7-44c3933fb16f</VendorClientTra
nsactionId>
  <linkTypeld>1</linkTypeld>
  <Description>Download link for a 64 bit Microsoft software product (large setup file/s
or ISO).</Description>
  </ProductLink>
</Links>
  <Files />
  <Errors />
  <Status>Завершен успешно</Status>
</TransactionOrderLine>

```

## Заказ

Тег в XML	Описание
OrderNum	Номер заказа в eShop
DateOn	дата/время размещения заказа у вендора в формате GMT
PartNum	внутренний уникальный идентификатор продукта в webstore
Quantity	количество

Status	статус заказа
Errors	Блок ошибки
Keys	Блок с ключами
Links	Блок ссылок для скачивания
Status	статус заказа

## Блок с ключами

Если заказанное количество более 1, то ключей в заказе будет более 1. Они представлены структурой:

```

<Keys>
  <ProductKey>
  ...
</ProductKey>
  <ProductKey>
  ...
</ProductKey>
</Keys>

```

## Ключ

Тег в XML	Описание
ActiveKey	ключ активации
DateAdd	дата/время выдачи ключа в формате GMT

Description	Описание
SequenceNumber	номер для техподдержки
VendorClientTransactionId	код транзакции
keyTypeId	тип ключа

Если заказанное количество более 1, то ссылок как и ключей будет тоже много. Они представлены структурой:

```

<Links>
  <ProductLink>
  ...
  </ProductLink>
  <ProductLink>
  ...
  </ProductLink>
</Links>

```

При покупке нескольких единиц товара возможны следующие ситуации:

- для каждой добавляется свой набор ссылок и ключей, т.е. на 1 ключ может быть 1 или более ссылок (на каждый ключ своя ссылка и даже с выбором дистрибутива);
- для всех добавленных ключей имеется 1 ссылка (например на скачивание единого дистрибутива);

Для однозначного сопоставления ключей и ссылок в теге [ProductKey](#) , [ProductLink](#) и [ProductFile](#) имеется тег [VendorClientTransactionId](#).

Тег в XML	Описание
-----------	----------

DateAdd	дата/время выдачи ссылки в формате GMT
Description	Описание
Link	Ссылка на дистрибутив
VendorClientTransactionId	код транзакции
linkTypeId	тип ссылки

В примере был представлен XML, который вернется, если сервису удалось при добавлении заказа (методом AddOrder) сразу получить для него ключи и ссылки. Об этом указывает статус заказа «Завершен успешно».

## 6. Пример заказа с файлом

**Важно:** Продукты, которые возвращают результат выполнения заказа в виде файла имеют свойство Fulfillmenttype=11. Если Ваш магазин еще не готов работать с таким типом результата, то Вы можете не отображать данные продукты на витрине, отфильтровав их по этому полю.

```
<?xml version="1.0" encoding="utf-8"?>
<TransactionOrderLine xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <OrderNum>1gdsfgs</OrderNum>
  <DateOn>2015-02-10T12:57:26.8535424</DateOn>
  <quantity>1</quantity>
  <TotalSumm>99</TotalSumm>
  <ItemPrice>99</ItemPrice>
  <currency>RUR</currency>
  <ProductId>dd0f9cd5-3fd7-4fc8-9bdb-a4d8ef4e7b2d</ProductId>
  <PartNum>PAV-08-001-</PartNum>
```

```

<endUserInfo>
  <Id>00000000-0000-0000-0000-000000000000</Id>
  <Email>qq@mail.ru</Email>
  <CustomerName>ivan</CustomerName>
  <CustomerLastName>ivanov</CustomerLastName>
  <Country>russia</Country>
  <City>mosc</City>
  <KasperskyPIN>SF09RU02</KasperskyPIN>
  <LegalEntity>>false</LegalEntity>
</endUserInfo>
<Keys />
<Links>
  <ProductLink>
    <Link>https://avast.com</Link>
    <DateAdd>2015-02-10T12:57:50.707</DateAdd>
  </ProductLink>
</Links>
<VendorClientTransactionId>43164545-b361-4d03-a9ca-7c8468e98367</VendorClientT
ransactionId>
  <linkTypeId>1</linkTypeId>
  <Description>avast link description</Description>
</ProductLink>
</Links>
<Files>
  <ProductFile>
    <DateAdd>2015-02-10T12:57:50.71</DateAdd>
  </ProductFile>
</Files>
<VendorClientTransactionId>43164545-b361-4d03-a9ca-7c8468e98367</VendorClientT
ransactionId>
  <Description>LICENSE FILE</Description>
  <FileName>lic_1gdsfsg</FileName>
  <FileExtension>avast</FileExtension>
  <FileSize>3544</FileSize>
</ProductFile>
<DownloadUrl>http://tayga:99/Home/Get?fileid=882a07ec-7403-4913-8cb9-6deebcba23
15&amp;password=DRDBZSgG</DownloadUrl>
  <Password>DRDBZSgG</Password>
</ProductFile>

```

```

</Files>
<Errors />
<Status>Завершен успешно</Status>
<ResellerPresetNum>0</ResellerPresetNum>
</TransactionOrderLine><?xml version="1.0" encoding="utf-8"?>

```

## Файл

Тег в XML	Описание
DateAdd	дата/время создания файла в формате GMT
Description	Описание
FileName	Название файла
VendorClientTransactionId	код транзакции
FileExtension	Тип файла
FileSize	размер в байтах
DownloadUrl	Ссылка на скачивание
Password	Пароль к файлу (уже присутствует в ссылке на скачивание)

## 7. Ожидание результата

Если сервису не удастся получить ключи по каким-либо причинам для заказа сразу при его размещении, то такой заказ становится в очередь для повторных попыток получить эти ключи. В этом случае ответ на вызов метода [AddOrder](#) будет выглядеть следующим образом:

```
<TransactionOrderLine xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <OrderNum>155-6786-979</OrderNum>
  <DateOn>2012-11-29T16:34:46.04</DateOn>
  <quantity>1</quantity>
  <PartNum>T5D-00432</PartNum>
  <Keys />
  <Links />
  <Files />
  <Errors />
  <Status>В процессе</Status>
</TransactionOrderLine>
```

Заказ будет находиться в обработке (“в процессе”). Необходимо периодически (на усмотрение магазина 1 – 10 мин) запрашивать по нему обновленную информацию методом `<Service_url>/GetOrder/<номер заказа>`.

[GetOrder \(https://sandbox.mont.ru/B2BService.svc/web/GetOrder/123\)](https://sandbox.mont.ru/B2BService.svc/web/GetOrder/123)

Входные параметры:

123 - Номер заказа в eShop

На выходе:

Заказ

## 8. Обработка ошибок

Заказ в ходе своего жизненного цикла из состояния «в процессе» может перейти либо в состояние «Завершен успешно», либо в состояние «Завершен с ошибкой».

Пример. XML заказа с ошибкой.

```
<TransactionOrderLine xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<OrderNum>155-6786-979</OrderNum>
<DateOn>2012-11-29T16:34:46.04</DateOn>
<quantity>1</quantity>
<PartNum>T5D-00432</PartNum>
<Keys />
<Links />
<Files />
<Errors>
  <PurchaseError>
    <Details>Ключи для товара закончились на складе</Details>
    <Code>11</Code>
    <VendorDetails>Данный ключ был ранее использован для
продления.</VendorDetails>
  </PurchaseError>
</Errors>
<Status>Завершен с ошибкой</Status>
</TransactionOrderLine>
```

Результатом работы метода `AddOrder` может стать сообщение об ошибке (`http_code = 500`), в результате которой заказ не будет даже размещен в системе. Вид ответа с сообщением об ошибке выглядит так:

```
<BusinessError>
  <ErrorDetails>Внутренняя ошибка системы</ErrorDetails>
  <ErrorCode>1</ErrorCode>
  <VendorDetails>Продлеваемый ключ не найден.</VendorDetails>
</BusinessError>
```

Список таких ошибок с их кодами приведен ниже:

1, "Внутренняя ошибка системы."

- 2, "Продукт не найден"
- 4, "Такой номер заказа уже есть в системе"
- 5, "Ошибка работы системы вендора."
- 6, "Превышен кредитный лимит."
- 8, "Номер заказа не указан."
- 9, "Неверный формат данных."
- 10, "Продукт заблокирован"
- 12, "Лицензионная форма не передана"
- 13, "Необходимые данные о предыдущей лицензии не были переданы"
- 15, "Лицензия заполнена неверно"
16. "Реселлер заблокирован"
17. "Количество задано не верно"

В поле [VendorDetails](#) попадает детальное описание от вендора причины отказа обработки заказа.

Следует отметить, что данное поле заполняется только при [ErrorCode = 5](#).

## К ошибкам, которые могут возникнуть в ходе обработки размещенного заказа можно отнести:

- 1, "Внутренняя ошибка системы."
- 5, "Ошибка работы системы вендора."
- 11, "Ключи для товара закончились на складе."

1. Получение информации о размещенном заказе.

Получить XML, содержащий всю необходимую информацию о заказе можно, воспользовавшись методом [<Service\\_url>/GetOrder/<OrderNum>](#). Результатом работы этого метода может быть XML, представленный в примере 4, 5 или 6. Кроме того, результатом может стать следующий набор ошибок:

1, "Внутренняя ошибка системы."

3, "Заказ не найден"

## 9. Получение справочной информации

### 1. CheckMyBalance()

Для проверки состояния (блокировки) клиента и его кредитного лимита можно воспользоваться методом [CheckMyBalance\(\)](#)

Пример вызова: GET: /CheckMyBalance

В ответ придет структура вида :

```
<ClientBalance xmlns="http://schemas.datacontract.org/2004/07/MontB2B.Contracts"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <AvailableLimit>10000000</AvailableLimit>
  <BlockReason />
  <Blocked>>false</Blocked>
</ClientBalance>
```