

ТЕХНИЧЕСКОЕ ОПИСАНИЕ WEB-СЕРВИСОВ

Версия 1.32

СОДЕРЖАНИЕ

Оглавление

1 СПИСОК РЕАЛИЗОВАННЫХ МЕТОДОВ	5
1.1 ПОЛУЧЕНИЕ СПИСКА КАТАЛОГОВ ТОВАРОВ	5
1.2 ПОЛУЧЕНИЕ КАТЕГОРИЙ ТОВАРОВ КАТАЛОГА	5
1.3 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ КАТЕГОРИИ	6
1.4 ПОЛУЧЕНИЕ ОПИСАНИЯ ТОВАРА	6
1.5 ПОЛУЧЕНИЕ ИЗОБРАЖЕНИЙ ТОВАРА	7
1.6 ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ЦЕНАХ И КОЛИЧЕСТВЕ ТОВАРА ПО КОДУ ТОВАРА	7
1.7 ПОЛУЧЕНИЕ СПИСКА НОВЫХ И УЦЕНЕННЫХ ТОВАРОВ	8
1.8 ПОИСК ТОВАРОВ ПО КЛЮЧЕВЫМ СЛОВАМ	8
1.9 ПОИСК ДАТЫ НАКЛАДНОЙ ПО НОМЕРУ СТИКЕРА И СЕРИЙНОМУ НОМЕРУ	8
1.10 ПОЛУЧЕНИЕ ИНФОРМАЦИИ О КУРСАХ ВАЛЮТ И ПР	9
1.11 ПОЛУЧЕНИЕ СПИСКА ЮРИДИЧЕСКИХ ЛИЦ	9
1.12 ПОЛУЧЕНИЕ СПИСКА ТИПОВ ДОСТУПНЫХ ДОКУМЕНТОВ	10
1.13 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ	10
1.14 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ ДОКУМЕНТА	10
1.15 СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ДОКУМЕНТОВ	11
1.16 ПОЛУЧЕНИЕ ИНФОРМАЦИИ И ЗАКАЗ ОТСУТСТВУЮЩИХ ТОВАРОВ	11
1.17 УДАЛЕНИЕ ДОКУМЕНТОВ	11
1.18 РАЗБИЕНИЕ И СЛИЯНИЕ РЕЗЕРВОВ	12
1.19 ПРОДЛЕНИЕ РЕЗЕРВА	12
1.20 ПОИСК ТОВАРА ПО PART NUMBER	13
1.21 ПОЛУЧЕНИЕ СПИСКА СУЩЕСТВУЮЩИХ ТОВАРОВ	13
1.22 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ ДОСТАВКИ	13
1.23 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ СЧЕТОВ ДЛЯ ОФОРМЛЕНИЯ ДОСТАВКИ	14
1.24 ПОЛУЧЕНИЕ ПОДРОБНОЙ ИНФОРМАЦИИ ДОКУМЕНТА ДОСТАВКИ	14
1.25 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ ДАТ ДОСТАВКИ	15
1.26 ПОЛУЧЕНИЕ ВОЗМОЖНЫХ ТИПОВ ДОСТАВКИ	15
1.27 ПОЛУЧЕНИЕ СПИСКА ВАРИАНТОВ ОПЛАТЫ ДОСТАВКИ	15
1.28 ПОЛУЧЕНИЕ СПИСКА ТАРИФОВ ДОСТАВКИ	16
1.29 СОХРАНЕНИЕ ДОКУМЕНТА ДОСТАВКИ	16
1.30 ОТГРУЗКА ПО ДОКУМЕНТУ ДОСТАВКИ	16
1.31 УДАЛЕНИЕ ДОКУМЕНТА ДОСТАВКИ	17
2 ОПИСАНИЕ МЕТОДОВ	18

2.1 МОДУЛЬ «АУТЕНТИФИКАЦИЯ».....	18
2.1.0 ОПИСАНИЕ МОДУЛЯ	18
2.1.1 ПОЛУЧЕНИЕ ВРЕМЕННОГО КОДА ДЛЯ ДАЛЬНЕЙШЕЙ РАБОТЫ.....	18
2.2 МОДУЛЬ «КАТАЛОГ».....	20
2.2.0 ОПИСАНИЕ МОДУЛЯ	20
2.2.1 ПОЛУЧЕНИЕ СПИСКА КАТАЛОГОВ ТОВАРОВ	21
2.2.2 ПОЛУЧЕНИЕ СПИСКА КАТЕГОРИЙ ТОВАРОВ КАТАЛОГА.....	22
2.2.3 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ КАТЕГОРИИ	23
2.2.4 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ КАТЕГОРИИ ВМЕСТЕ С УДАЛЕННЫМИ ТОВАРАМИ.....	24
2.2.5 ПОЛУЧЕНИЕ ОПИСАНИЯ ТОВАРА.....	24
2.2.6 ПОЛУЧЕНИЕ ОПИСАНИЯ ТОВАРА ПО АРТИКУЛУ	25
2.2.7 ПОЛУЧЕНИЕ ОПИСАНИЙ ТОВАРОВ ПО КАТЕГОРИИ	26
2.2.8 ПОЛУЧЕНИЕ ИЗОБРАЖЕНИЙ ТОВАРА	27
2.2.9 ПОЛУЧЕНИЕ ИЗОБРАЖЕНИЙ ТОВАРОВ ПО КАТЕГОРИИ.....	29
2.2.10 ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ЦЕНАХ И КОЛИЧЕСТВЕ ТОВАРА ПО КОДУ ТОВАРА	30
2.2.11 ПОЛУЧЕНИЕ СПИСКА НОВЫХ ТОВАРОВ	30
2.2.12 ПОЛУЧЕНИЕ СПИСКА УЦЕНЕННЫХ ТОВАРОВ.....	31
2.2.13 ПОИСК ТОВАРА ПО PART NUMBER.....	31
2.2.14 ПОИСК ДАТЫ НАКЛАДНОЙ ПО НОМЕРУ СТИКЕРА	32
2.2.15 ПОИСК ДАТЫ НАКЛАДНОЙ ПО СЕРИЙНОМУ НОМЕРУ ТОВАРА.....	32
2.2.16 ПОЛУЧЕНИЕ ПРОЧЕЙ ИНФОРМАЦИИ (КУРСЫ ВАЛЮТ).....	33
2.2.17 ПОИСК ТОВАРОВ ПО КЛЮЧЕВЫМ СЛОВАМ	33
2.3 МОДУЛЬ «ДОКУМЕНТЫ».....	34
2.3.0 ОПИСАНИЕ МОДУЛЯ	34
2.3.1 ПОЛУЧЕНИЕ СПИСКА ЮРИДИЧЕСКИХ ЛИЦ.....	35
2.3.2 ПОЛУЧЕНИЕ СПИСКА ТИПОВ ДОСТУПНЫХ ДОКУМЕНТОВ	35
2.3.3 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ	36
2.3.4 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ ДОКУМЕНТА.....	37
2.3.5 СОЗДАНИЕ ДОКУМЕНТОВ.....	38
2.3.6 РЕДАКТИРОВАНИЕ ДОКУМЕНТОВ.....	39
2.3.7 УДАЛЕНИЕ ДОКУМЕНТОВ	41
2.3.8 СОЗДАНИЕ НОВОГО РЕЗЕРВА	41
2.3.9 ЗАКАЗ ОТСУТСТВУЮЩИХ ТОВАРОВ	42
2.3.10 ЗАПРОС СПИСКА ТОВАРОВ В РЕМОНТЕ.....	43
2.3.11 ЗАПРОС ДАННЫХ ДОКУМЕНТА ОТГРУЗКИ.....	43
2.3.12 РАЗБИЕНИЕ РЕЗЕРВА НА ДВА ДОКУМЕНТА.....	44

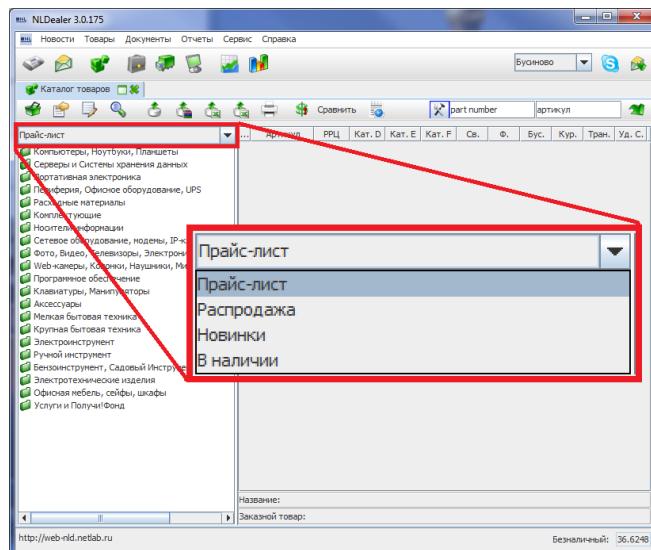
2.3.13 ОБЪЕДИНЕНИЕ РЕЗЕРВОВ В ОДИН ДОКУМЕНТ	44
2.3.14 ПРОДЛЕНИЕ РЕЗЕРВА.....	45
2.4 МОДУЛЬ «ДОСТАВКА».....	46
2.4.0 ОПИСАНИЕ МОДУЛЯ	46
2.4.1 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ ДОСТАВКИ	46
2.4.2 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ СЧЕТОВ ДЛЯ ОФОРМЛЕНИЯ ДОСТАВКИ.....	47
2.4.3 ПОЛУЧЕНИЕ ПОДРОБНОЙ ИНФОРМАЦИИ ДОКУМЕНТА ДОСТАВКИ	48
2.4.4 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ ДАТ ДОСТАВКИ	48
2.4.5 ПОЛУЧЕНИЕ ВОЗМОЖНЫХ ТИПОВ ДОСТАВКИ	49
2.4.6 ПОЛУЧЕНИЕ СПИСКА ВАРИАНТОВ ОПЛАТЫ ДОСТАВКИ.....	50
2.4.7 ПОЛУЧЕНИЕ СПИСКА ТАРИФОВ ДОСТАВКИ	50
2.4.8 СОХРАНЕНИЕ ДОКУМЕНТА ДОСТАВКИ	51
2.4.9 ОТГРУЗКА ПО ДОКУМЕНТУ ДОСТАВКИ.....	52
2.4.10 УДАЛЕНИЕ ДОКУМЕНТА ДОСТАВКИ	52
3 ПРИМЕРЫ СКРИПТОВ ДЛЯ «1С ПРЕДПРИЯТИЕ 8.2».....	53
3.1 ПРИМЕР НАСТРОЙКИ «1С ПРЕДПРИЯТИЕ 8.2» ДЛЯ РАБОТЫ С ВЕБ СЕРВИСАМИ. СОЗДАНИЕ ССЫЛОК НА СЕРВИСЫ.....	53
3.2 СОЗДАНИЕ ФОРМЫ ОБРАБОТКИ ДЛЯ ТЕСТИРОВАНИЯ	55
Образец кода 1с для управляемых форм для работы с изображениями	71
4. ПРИМЕРЫ КОДА НА PHP И JAVASCRIPT	73
4.1 HTML код	73
4.2 JavaScript код.....	79
4.3 PHP код.....	85
5. ПРИМЕРЫ ВЫВОДА МЕТОДОВ	103
6. ЧТО НОВОГО? ИЗМЕНЕНИЯ В ДОКУМЕНТАЦИИ	114

1 СПИСОК РЕАЛИЗОВАННЫХ МЕТОДОВ

Бизнес-логика, представленная веб-сервисами, ранее была реализована в программе NLDealer online. Для пояснения назначения тех или иных методов веб-сервисов, будем ставить им в соответствие операции и элементы интерфейса программы NLDealer online.

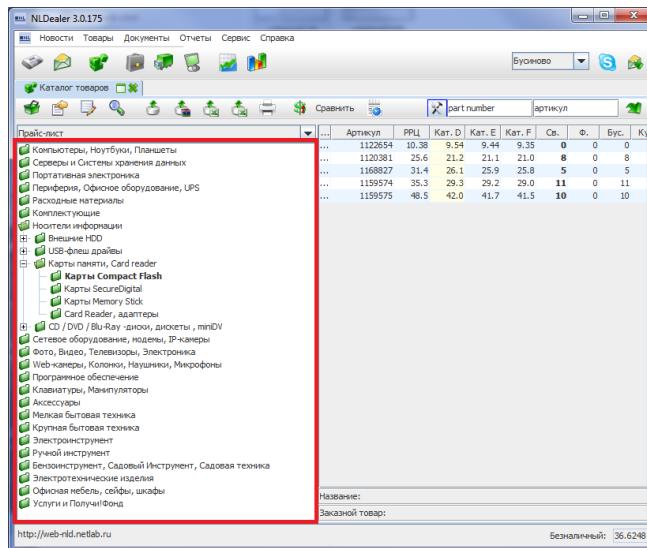
1.1 ПОЛУЧЕНИЕ СПИСКА КАТАЛОГОВ ТОВАРОВ

Аналогичный список представлен в программе **NLDealer online** в каталоге товаров в верхнем левом углу окна, над деревом категорий товаров.



1.2 ПОЛУЧЕНИЕ КАТЕГОРИЙ ТОВАРОВ КАТАЛОГА

Иерархия категорий товаров представлена в каталоге в виде дерева.



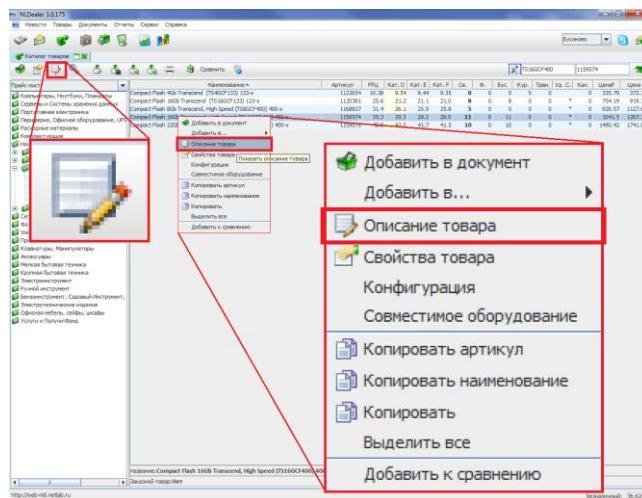
1.3 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ КАТЕГОРИИ

Отображение списка товаров по выбранной категории из дерева категорий осуществляется в главной таблице, как указано на рисунке ниже:

The screenshot shows the NLDealer 3.0.175 application window with a large grid table on the right side. A red arrow points to a specific row in the table, which corresponds to the category selected in the tree view on the left. The grid table has columns for Артикул (Article), Мод. (Model), РРЦ (Retail Price), Кат. D (Category D), Кат. E (Category E), Кат. F (Category F), Ср. (Avg), Ф. (Avg), Бус. (Business), and Цена (Price).

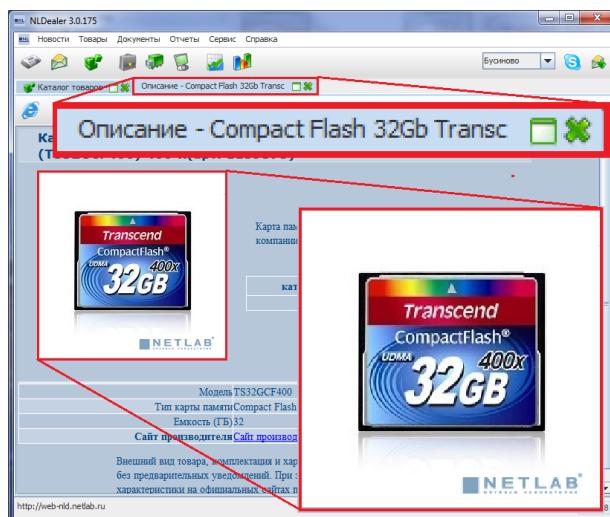
1.4 ПОЛУЧЕНИЕ ОПИСАНИЯ ТОВАРА

Получить описание товара можно при нажатии специальной кнопки на панели инструментов каталога или посредством обращения к контекстному меню в таблице товаров каталога или документа.



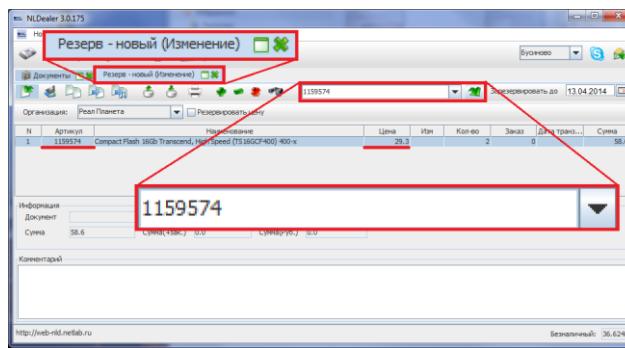
1.5 ПОЛУЧЕНИЕ ИЗОБРАЖЕНИЙ ТОВАРА

Одно из изображений товара используется в форме с описанием.
В общем случае изображений более одного. Все они доступны через веб-сервисы.



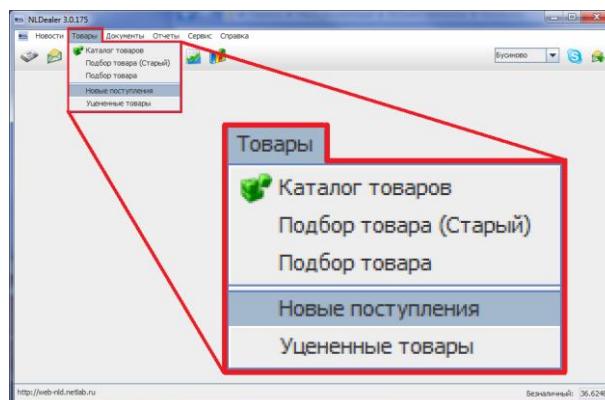
1.6 ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ЦЕНАХ И КОЛИЧЕСТВЕ ТОВАРА ПО КОДУ ТОВАРА

Аналогичная информация возвращается при добавлении товара в документ по артикулу.



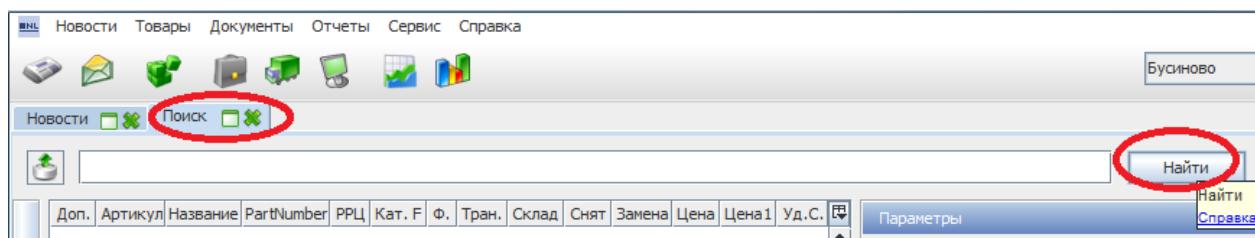
1.7 ПОЛУЧЕНИЕ СПИСКА НОВЫХ И УЦЕНЕННЫХ ТОВАРОВ

Аналогичные списки доступны в программе при обращении через главное меню: «Товары» → «Новые поступления» и «Уцененные товары» соответственно.



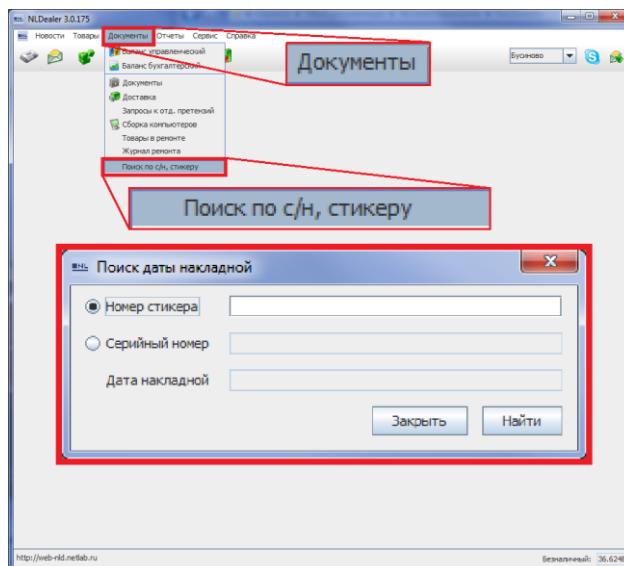
1.8 ПОИСК ТОВАРОВ ПО КЛЮЧЕВЫМ СЛОВАМ

Аналогичные списки доступны в программе при обращении через главное меню: «Товары» → «Новые поступления» и «Уцененные товары» соответственно.



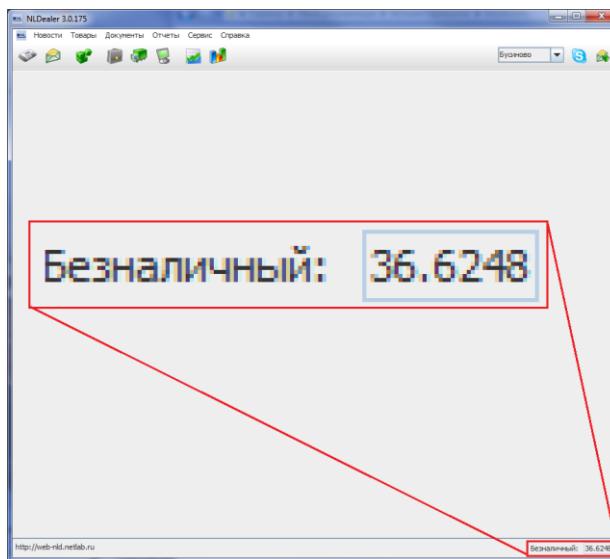
1.9 ПОИСК ДАТЫ НАКЛАДНОЙ ПО НОМЕРУ СТИКЕРА И СЕРИЙНОМУ НОМЕРУ

Аналогичный поиск осуществляется при помощи диалогового окна, доступного в меню «Документы» → «Поиск по с/н, стикеру».



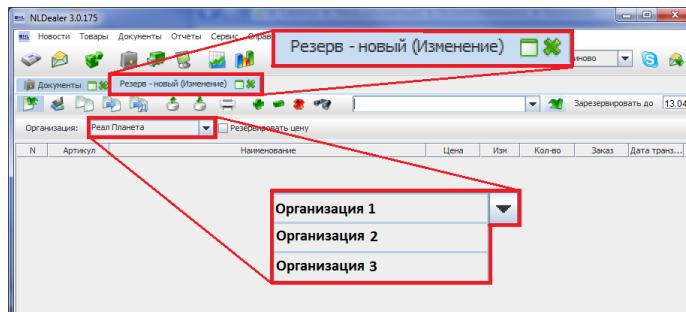
1.10 ПОЛУЧЕНИЕ ИНФОРМАЦИИ О КУРСАХ ВАЛЮТ И ПР

Информация о безналичном курсе доступна в правом нижнем углу главного окна программы.



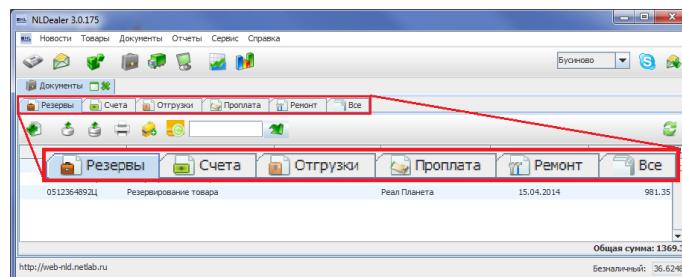
1.11 ПОЛУЧЕНИЕ СПИСКА ЮРИДИЧЕСКИХ ЛИЦ

Список юридических лиц (организаций) доступен для выбора в процессе создания или редактирования документа.



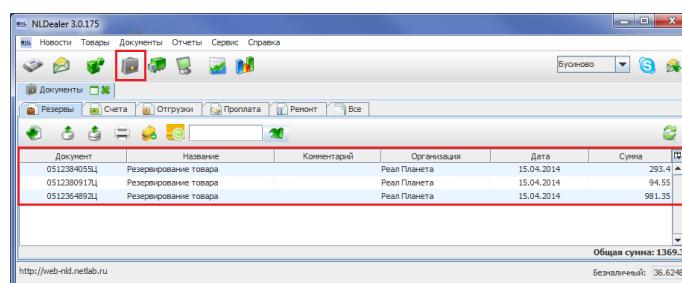
1.12 ПОЛУЧЕНИЕ СПИСКА ТИПОВ ДОСТУПНЫХ ДОКУМЕНТОВ

Через веб-сервисы можно совершать операции с документами, отображаемыми в журнале документов. Каждой закладке в журнале поставлен в соответствие тип документов. В дальнейшем список типов может быть дополнен.



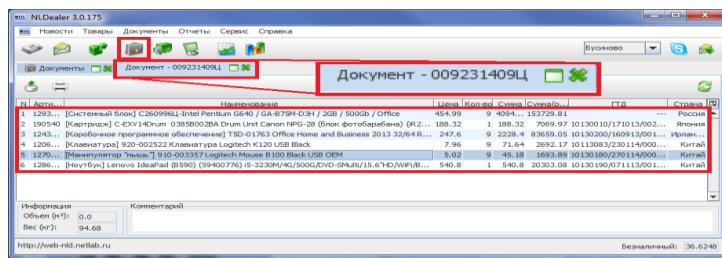
1.13 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ

Список документов доступен в журнале документов, в соответствующей типу документов вкладке (резервы, счета, ...)



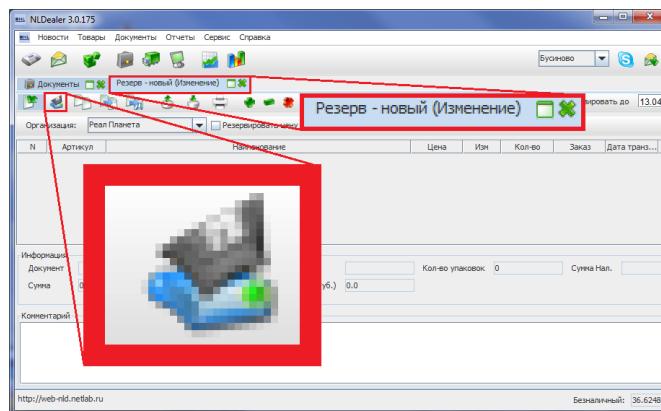
1.14 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ ДОКУМЕНТА

Список товаров отображается в главной таблице формы, при открытии документа.



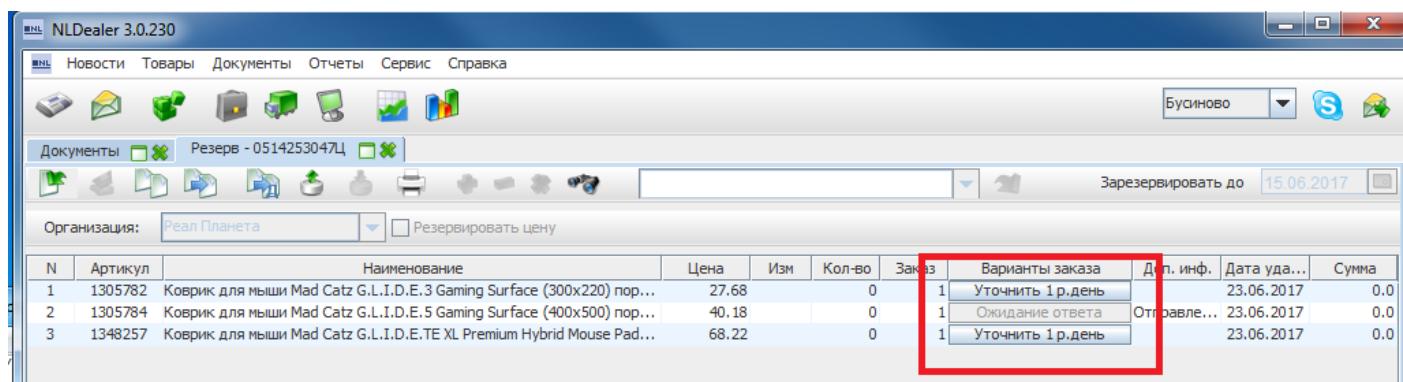
1.15 СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ДОКУМЕНТОВ

Операции создания и редактирования документа соответствуют нажатию на кнопку «Сохранить» в форме резерва или счета.



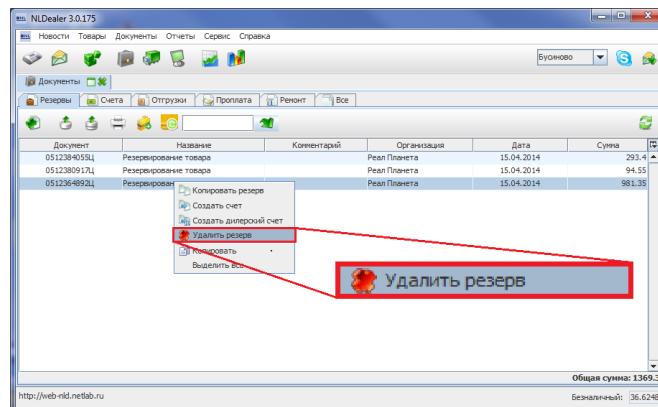
1.16 ПОЛУЧЕНИЕ ИНФОРМАЦИИ И ЗАКАЗ ОТСУТСТВУЮЩИХ ТОВАРОВ

В программе NLDealer существует возможность, узнать информацию по отсутствующему товару и заказать его. Для этого в программе предусмотрена кнопка в колонке «Варианты заказа». (Более подробно о возможных состояниях данных кнопок рассказано в документации к системе NLDealer)



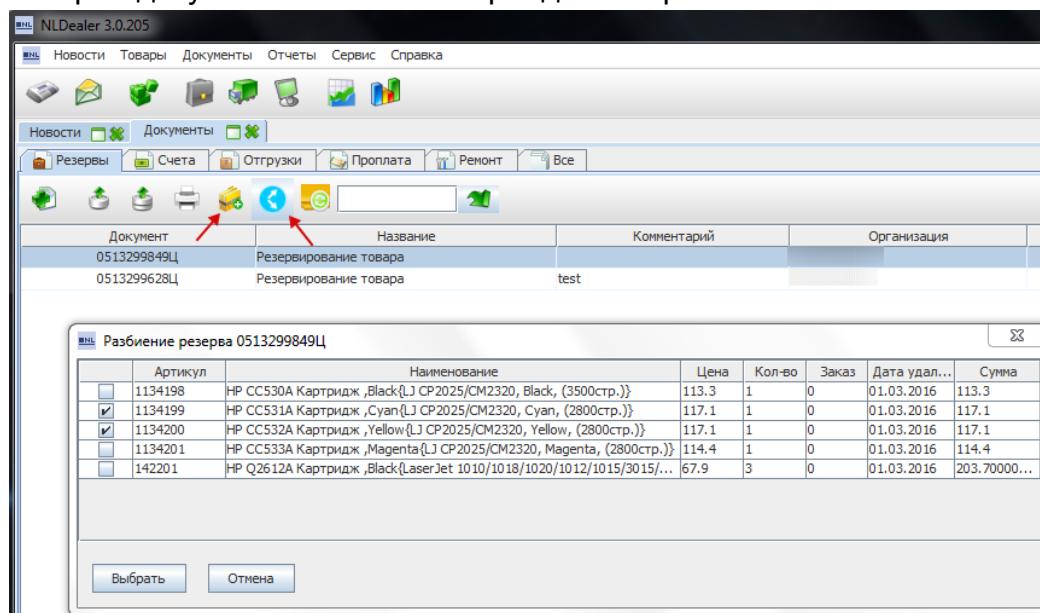
1.17 УДАЛЕНИЕ ДОКУМЕНТОВ

Удалению документа соответствует либо его сохранение без товаров (с пустой таблицей товаров) либо выбор соответствующего пункта контекстного меню в списке документов.



1.18 РАЗБИЕНИЕ И СЛИЯНИЕ РЕЗЕРВОВ

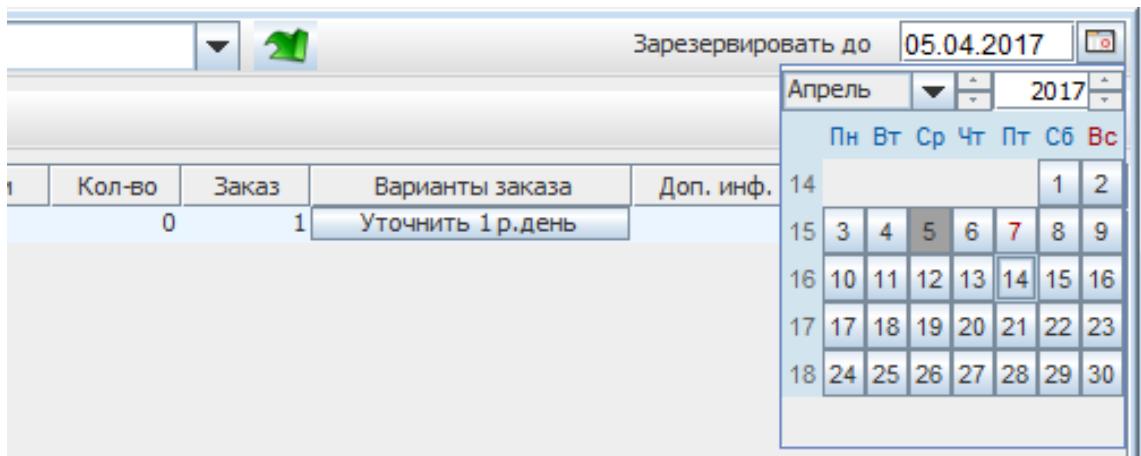
Разбиение позволяет разделить резерв на два документа на основе выбранных товаров. Объединение резервов позволяет объединить несколько документов в один. При этом второй документ исчезнет и перейдет в первый.



1.19 ПРОДЛЕНИЕ РЕЗЕРВА

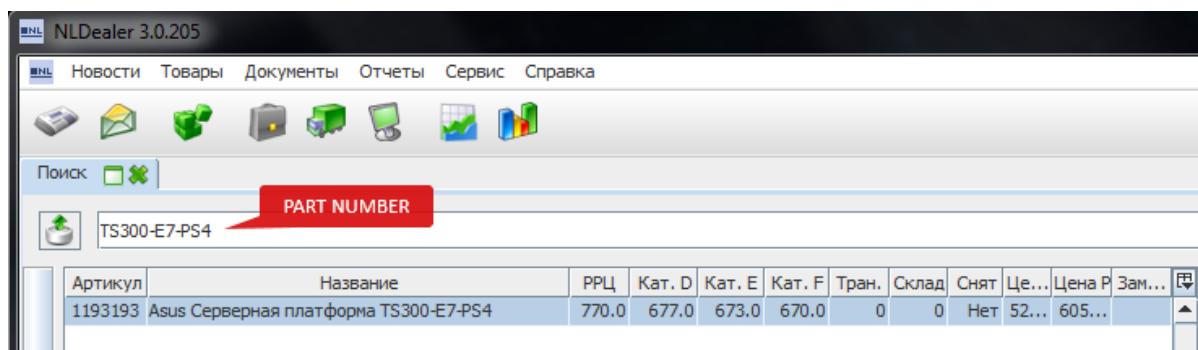
С помощью метода можно продлить резерв на ограниченный срок (в случае превышения возможного срока резервирования, программа автоматически выставит максимально отдаленную дату).

Примечание: каждый товар в резерве имеет свою дату удаления. Если она наступит, то несмотря ни на что, товар будет удален из резерва автоматически.



1.20 ПОИСК ТОВАРА ПО PART NUMBER

С помощью метода можно получить информацию о товаре на основании известного Part number товара.

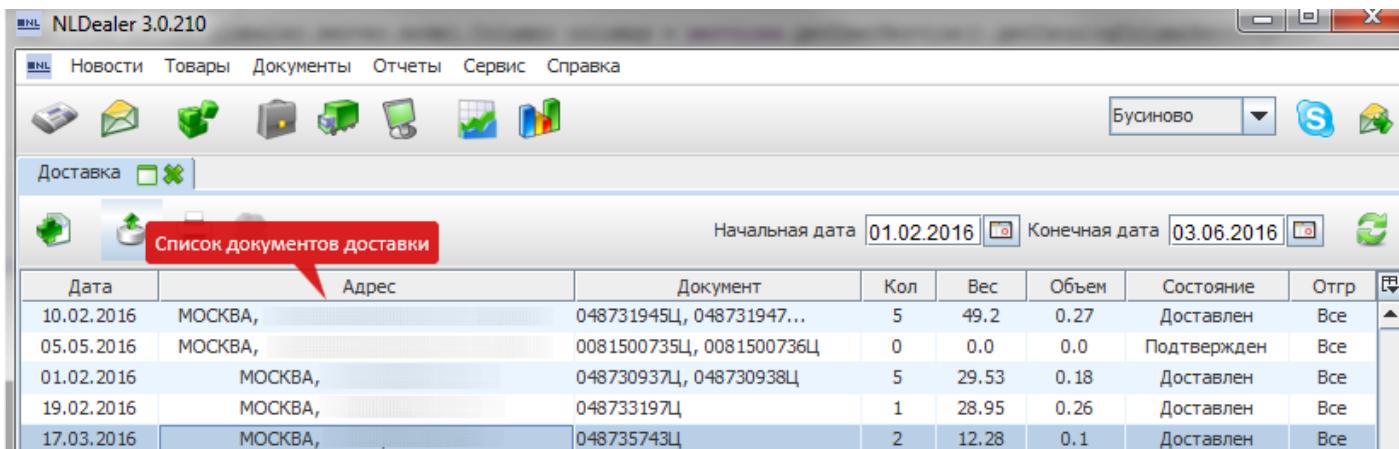


1.21 ПОЛУЧЕНИЕ СПИСКА СУЩЕСТВУЮЩИХ ТОВАРОВ

С помощью метода можно получить информацию о товарах, которые на данный момент существуют в каталоге фирмы. Ответ содержит информацию о названии, ценах по разным категориям и количестве на складах.

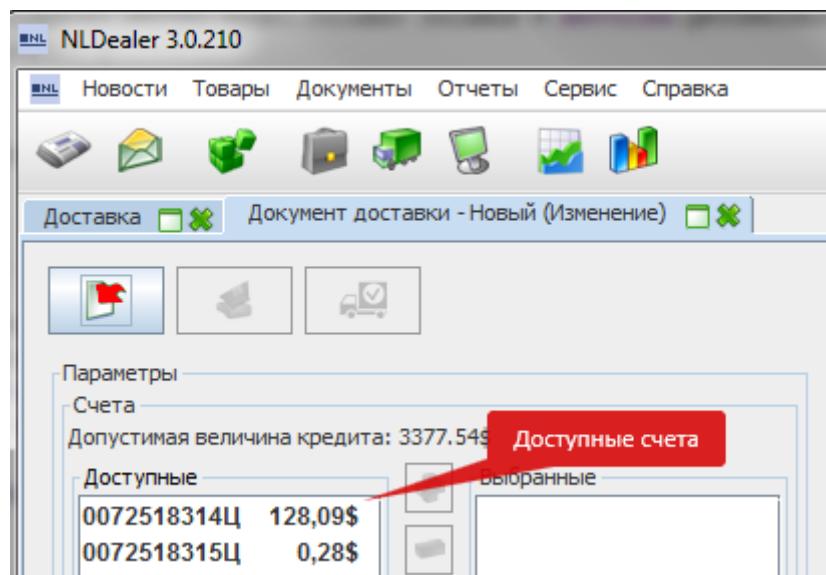
1.22 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ ДОСТАВКИ

Возвращает список созданных документов доставки.



1.23 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ СЧЕТОВ ДЛЯ ОФОРМЛЕНИЯ ДОСТАВКИ

С помощью метода можно получить список доступных документов (счетов) для оформления доставки.



1.24 ПОЛУЧЕНИЕ ПОДРОБНОЙ ИНФОРМАЦИИ ДОКУМЕНТА ДОСТАВКИ

Метод возвращает подробную информацию по документу доставки: содержание упаковки, выбранные документы, вес, объем, статус и т.д.

Скриншот программы для отслеживания доставки. В левой части окна есть список 'Выбранные' с тремя элементами: '048731945Ц (107273356Ц)', '048731947Ц (107273390Ц)' и '048731948Ц (107273391Ц)'. В правой части окна раздел 'Информация' содержит поля: 'Состояние' (Доставлен), 'Количество коробок' (5), 'Вес' (49.2), 'Объем' (0.27) и 'Тел. водителя' (пусто). Красная стрелка указывает на поле 'Вес'. Далее в разделе 'Товары' расположена таблица:

Артикул	Описание	Наименование	Кол-во
1381731	C346983Ц-Intel / Z97M-PLUS RTL / 2 x ...	Box №1	1
1381731	C346983Ц-Intel / Z97M-PLUS RTL / 2 x ...	Box №2	1
1381731	C346983Ц-Intel / Z97M-PLUS RTL / 2 x ...	Box №3	1
1381731	C346983Ц-Intel / Z97M-PLUS RTL / 2 x ...	Box №4	1
1387011	C351399Ц-Intel Core i7-4770 / Z87-A R...	Box №5	1

1.25 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ ДАТ ДОСТАВКИ

Возвращает возможные даты доставки товара.

Скриншот выпадающего меню с параметрами доставки. Поля: 'Дата' (05.05.2016, Четверг), 'Тип' (выбор из списка: <выбрать>, 04.05.2016 (Ср(Экспресс)), 05.05.2016 (Четверг), 06.05.2016 (Пятница), 10.05.2016 (Вторник), 11.05.2016 (Среда), 12.05.2016 (Четверг), 13.05.2016 (Пятница)). Красная стрелка указывает на поле 'Тип'.

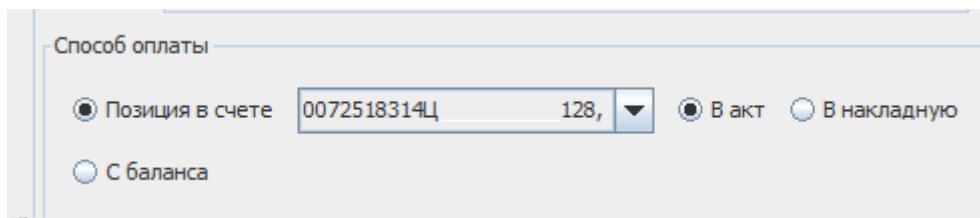
1.26 ПОЛУЧЕНИЕ ВОЗМОЖНЫХ ТИПОВ ДОСТАВКИ

Метод возвращает доступные типы доставки.

Скриншот выпадающего меню с параметрами доставки. Поля: 'Дата' (05.05.2016, Четверг), 'Тип' (выбор из списка: Доставка, <выбрать>), 'Тариф' (пусто), 'Адрес' (пусто), 'ТК' (пусто). Красная стрелка указывает на поле 'Тип'.

1.27 ПОЛУЧЕНИЕ СПИСКА ВАРИАНТОВ ОПЛАТЫ ДОСТАВКИ

Возвращает данные о доступных вариантах оплаты доставки (в акт, в накладную или с баланса).



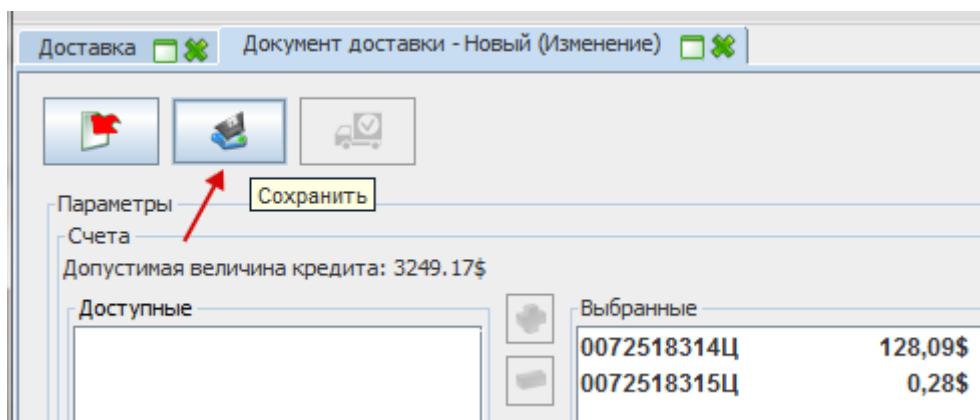
1.28 ПОЛУЧЕНИЕ СПИСКА ТАРИФОВ ДОСТАВКИ

С помощью этого метода можно получить список доступных тарифов и адресов доставки.

Выбор тарифа		OK	Очистить	Отмена
все	недавние			
	Адрес и ТК	Авто - до двери	Другие возможные варианты доставки	
119607, Москва,		<input type="radio"/> 5.00\$		
МОСКВА,		<input type="radio"/> 0.00\$		
МОСКВА, Ленинские горы,		<input type="radio"/> 5.00\$		

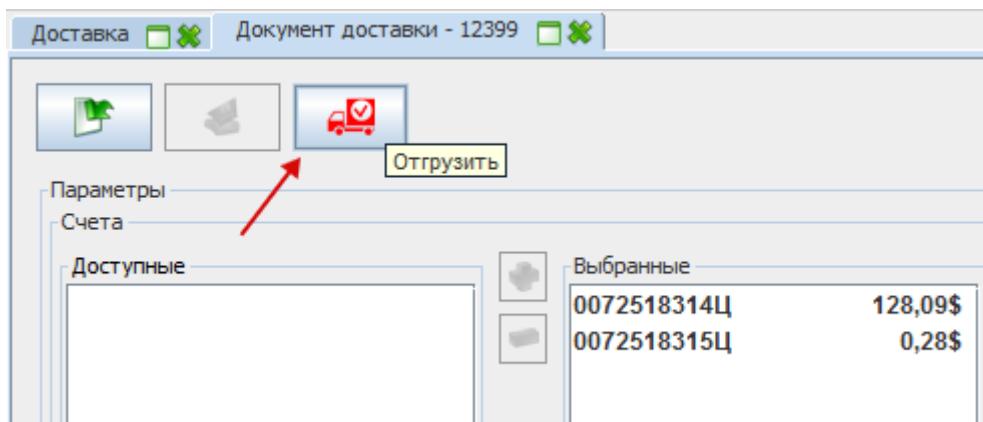
1.29 СОХРАНЕНИЕ ДОКУМЕНТА ДОСТАВКИ

С помощью этого метода можно сохранить документ доставки.



1.30 ОТГРУЗКА ПО ДОКУМЕНТУ ДОСТАВКИ

С помощью этого метода происходит подтверждение отгрузки товара. После выполнения метода документ будет недоступен для редактирования.



1.31 УДАЛЕНИЕ ДОКУМЕНТА ДОСТАВКИ

С помощью этого метода можно удалить сохраненный документ (кроме подтвержденных документов доставки).

2 ОПИСАНИЕ МЕТОДОВ

Ниже приводится описание всех имеющихся на данный момент методов веб сервисов. Раздел, посвященный Модулю «Аутентификация» является служебным, и предназначен для ознакомления пользователя с процедурой получения временного кода доступа к сервисам. В последующих разделах описаны методы реализующие бизнес-логику.

Ответ сервера может формироваться в двух видах – сжатым (gzip) и без сжатия. Мы настоятельно рекомендуем использовать методы, использующие сжатие для уменьшения кол-ва трафика. В зависимости от данных, сжатие может составлять от 5 до 10 раз (например, ответ метода catalogAction занимает 33 Кб в сжатом виде и 316 Кб без использования сжатия). Использования сжатия на более тяжелых методах может уменьшить принимаемый трафик на десятки мегабайт.

2.1 МОДУЛЬ «АУТЕНТИФИКАЦИЯ»

2.1.0 ОПИСАНИЕ МОДУЛЯ

Методы этого модуля описываются следующими URL:

<http://services.netlab.ru/AuthenticationService?wsdl> – wsdl описание

<http://services.netlab.ru/rest/authentication/token> - RESTful метод.

2.1.1 ПОЛУЧЕНИЕ ВРЕМЕННОГО КОДА ДЛЯ ДАЛЬНЕЙШЕЙ РАБОТЫ

Для работы с сервисами необходимо получить **временный код (токен)**, по которому, в процессе дальнейшей работы, будет идентифицироваться пользователь. Код имеет срок годности, по прошествии которого, токен будет считаться устаревшим. По истечении срока годности следует обратиться за получением нового. В случае если происходит попытка использования устаревшего токена, выдается соответствующее информационное сообщение.

Для получения токена используется **метод authenticate**.

В качестве входных параметров метод принимает **логин и пароль пользователя**.

В полях **«token»** и **«expired_in»** содержится токен и его срок годности соответственно.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:

```

TokenResponse
  status
    code
    message
  data
    token
    expired_in

```

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно, обратившись с **get** запросом по следующему URL-адресу:

<http://services.netlab.ru/rest/authentication/token.xml?username=<username>&password=<password>> - для формата **xml**

<http://services.netlab.ru/rest/authentication/token.json?username=<username>&password=<password>> - для формата **json**,

где вместо <username> и <password> следует подставить логин и пароль соответственно

ЗАМЕЧАНИЕ!

Ответ от сервера можно получать в двух форматах: **xml** и **json**. Для смены формата достаточно поменять в URL-адресе «**.xml**» на «**.json**». Все последующие примеры будут приводиться для формата **xml**, подразумевая, что смена формата очевидна.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

ПРИМЕР ПОЛУЧЕНИЯ ТОКЕНА С ПОМОШЬЮ WSDL НА ЯЗЫКЕ PHP (SOAP)

```

$client = new
SoapClient("http://services.netlab.ru/AuthenticationService?wsdl");
// Создаем soap контейнер

$params = array('arg0' => $username, 'arg1' => $password);
// Задаем массив параметров для запроса

$result = $client->authenticate($params);
// Вызываем метод authenticate

```

Входные параметры	Значение	Описание
arg0	Имя пользователя	Имя пользователя, заданное в кабинете NLDealer с правами доступа к веб сервисам
arg1	Пароль пользователя	Имя пользователя, заданное в кабинете NLDealer с правами доступа к веб сервисам

На выходе получаем стандартный массив stdClass:

```
stdClass Object
(
    [return] => stdClass Object
    (
        [status] => stdClass Object
        (
            [code] => 200
            [message] =>
        )

        [data] => stdClass Object
        (
            [token] => 28234234-1542-12d6-8tt6-176653db29da
            [expired_in] => 06.04.2017 13:15
        )
    )
)
```

Выходные параметры	Описание
code	Код ошибки (если есть)
message	Сообщение об ошибке
token	Значение токена
expired_in	Срок истечения токена

Данный массив можно обрабатывать любым удобным способом.

2.2 МОДУЛЬ «КАТАЛОГ»

2.2.0 ОПИСАНИЕ МОДУЛЯ

Методы этого модуля описываются следующими URL:

<http://services.netlab.ru/CatalogServiceZip?wsdl> – дескриптор, методы которого возвращают ответ в сжатом виде (gzip). Рекомендуется к использованию.

<http://services.netlab.ru/CatalogService?wsdl> – дескриптор, методы которого возвращают ответ в не сжатом виде.

http://services.netlab.ru/rest/catalogsZip/* - RESTful методы с ответом в сжатом виде. Рекомендуется к использованию.

<http://services.netlab.ru/rest/catalogs/> * - RESTful методы с ответом в не сжатом виде.

2.2.1 ПОЛУЧЕНИЕ СПИСКА КАТАЛОГОВ ТОВАРОВ

Каталоги товаров представляют собой различные способы группировки ассортимента. В настоящий момент существует 4 типа каталогов:

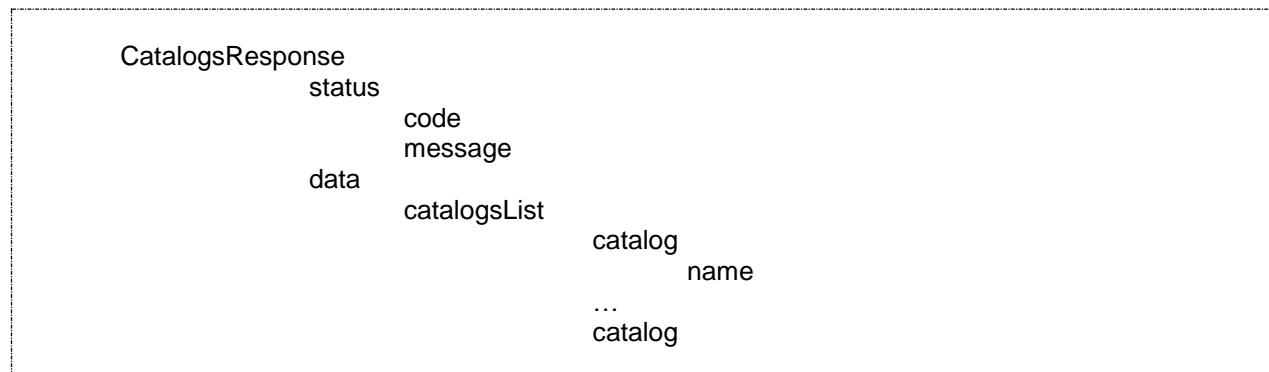
Прайс-лист	Товары, когда-либо продаваемые нашей компанией.
Новинки	Товары, ранее не присутствовавшие на рынке, и предлагаемые нашей компанией.
В наличии	Товары, находящиеся на складах нашей компании.
Распродажа	Товары с уцененной стоимостью.

В будущем возможно пополнение этого списка. Для получения актуального состояния используется метод **catalogsAction**.

Метод не принимает входных параметров.

В поле «**name**» возвращаемых данных содержатся имена каталогов товаров.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно, обратившись с **get** запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/list.xml?oauth_token=<token>

где вместо **<token>** следует подставить **временный код**.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.2 ПОЛУЧЕНИЕ СПИСКА КАТЕГОРИЙ ТОВАРОВ КАТАЛОГА

Каждый каталог товаров содержит **иерархическую структуру (дерево)** категорий товаров. Категория, находящаяся на уровень выше рассматриваемой, считается для нее родительской. Категории, находящиеся на самом нижнем уровне дерева, и содержащие списки товаров, называются листовыми.

Для получения списка категорий товаров выбранного каталога используется метод **catalogAction**.

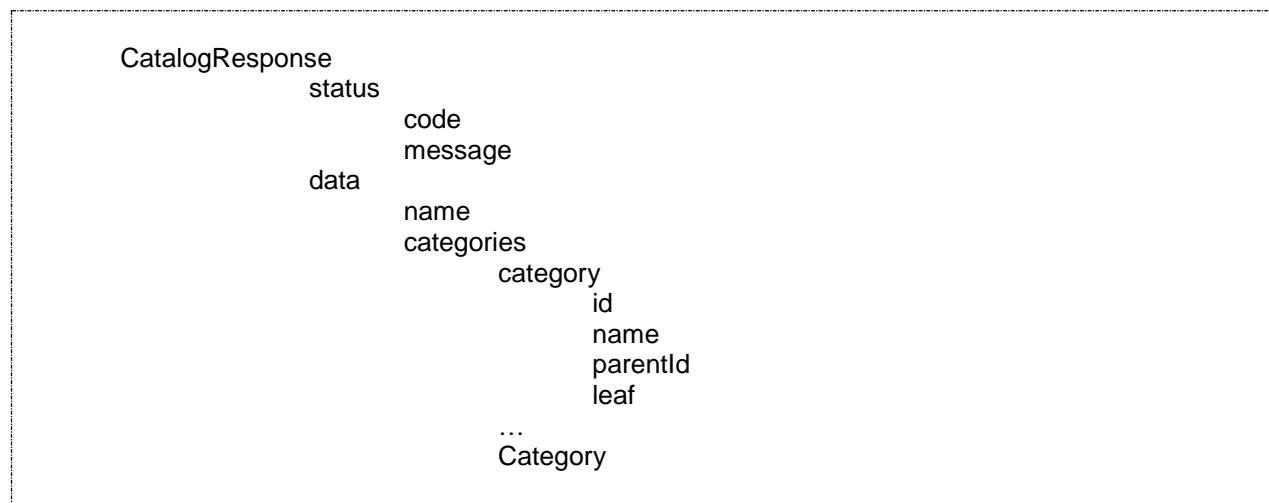
В качестве **входного параметра** метод принимает название каталога товаров.

Метод возвращает следующие данные:

Id	Уникальный идентификатор категории
Name	Название категории
parentId	Идентификатор родительской категории
Leaf	Признак <u>листовой категории</u> ¹

¹ Значение поля leaf является логическим и в различных языках программирования интерпретируется по-разному, так в 1С принимаемые значения будут ИСТИНА или ЛОЖЬ, а в C# и Java значения true или false соответственно.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/<catalogName>.xml?oauth_token=<token>

где вместо **<catalogName>** следует подставить название каталога, а вместо **<token>** временный код.

[Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С».](#)
[Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».](#)

2.2.3 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ КАТЕГОРИИ

По названию каталога и **уникальному идентификатору** листовой категории можно получить список товаров последней. Для этого используется метод **categoryAction**.

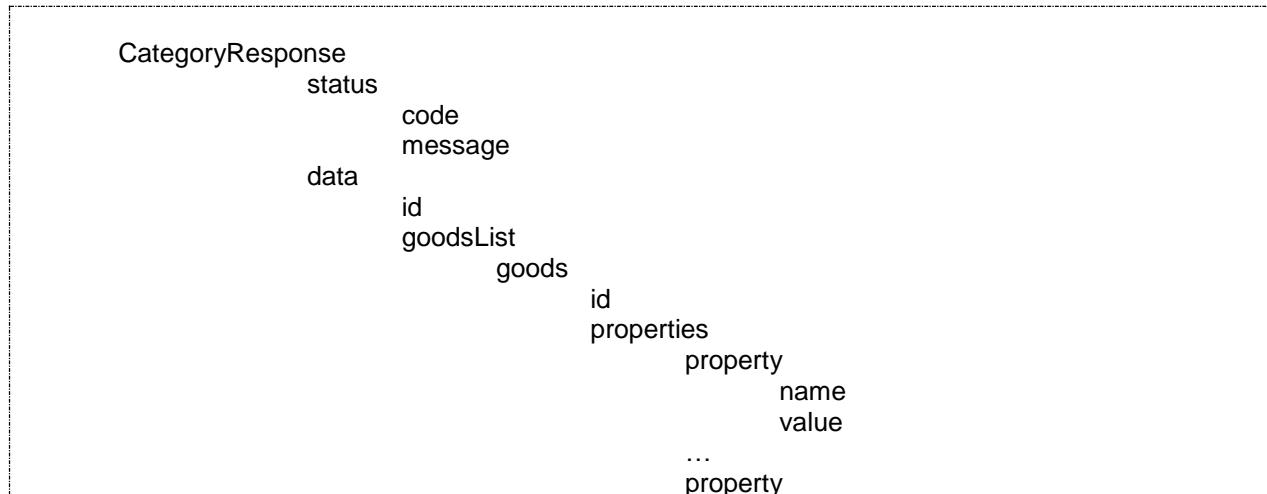
В качестве входных параметров принимаются следующие данные:

catalogName	Название каталога
categoryId	Идентификатор листовой категории

Метод возвращает следующие данные:

id	Уникальный идентификатор товара
properties	Список свойств товара. Содержит информацию о количестве, ценах и пр.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



Описанная структура данных содержит **пары(property) значений name-value**, в которых содержатся названия свойств и их значение соответственно. Здесь и далее, для описания разнородных объектов будет использоваться подобная конструкция.

*Примечание. Соответствие названий складов: **курская = бауманская, калужская = профсоюзная, лобненская=бусиновская**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/<catalogName>/<categoryId>.xml?oauth_token=<token>

где вместо **<catalogName>** следует подставить название каталога, вместо **<catalogId>** идентификатор категории товара, вместо **<token>** временный код

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.4 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ КАТЕГОРИИ ВМЕСТЕ С УДАЛЕННЫМИ ТОВАРАМИ

Метод аналогичен методу из пункта [ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ КАТЕГОРИИ](#), но с возможностью просмотра удаленных из каталога товаров, которые ранее были доступны в продаже. Название метода categoryWithDelAction.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/versions/2/<catalogName>/<categoryId>.xml?oauth_token=<token>&showDeleted=<showDeleted>

где вместо **<catalogName>** следует подставить название каталога, вместо **<catalogId>** идентификатор категории товара, вместо **<token>** временный код, вместо **<showDeleted>** - отображать удаленные товары. **1** (да) или **0** (нет).

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.5 ПОЛУЧЕНИЕ ОПИСАНИЯ ТОВАРА

Для получения списка характеристик товара (описания) по идентификатору товара используется **метод goodsAction**.

В качестве входных параметров принимаются следующие данные:

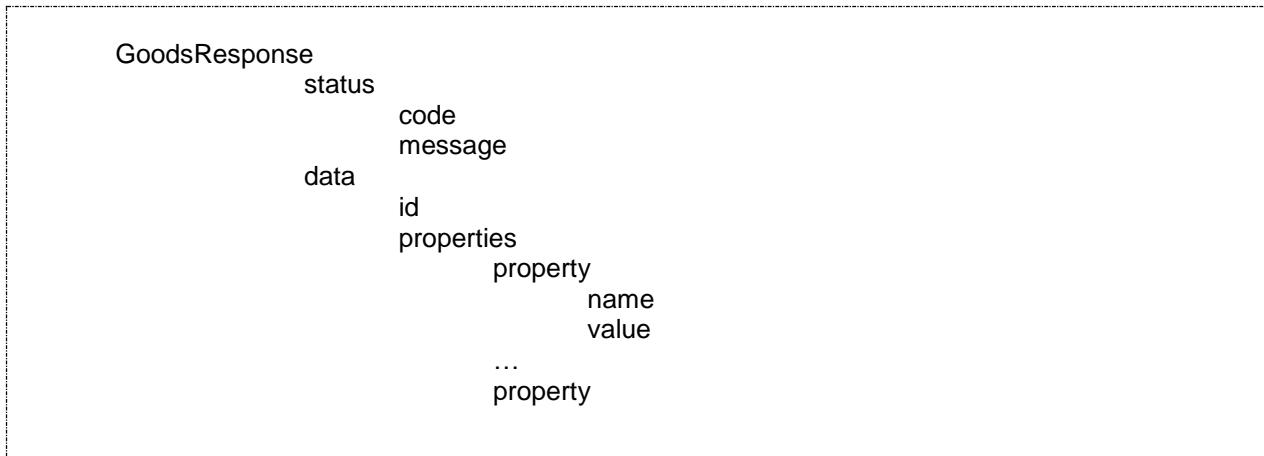
catalogName	Название каталога товаров
-------------	---------------------------

categoryId	Идентификатор категории товаров
goodsId	Идентификатор товара

Метод возвращает следующие данные:

id	Уникальный идентификатор товара
properties	Список свойств товара. Содержит информацию о характеристиках товара.

ВОЗВАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с **get запросом** по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/<catalogName>/<categoryId>/<goodsId>.xml?oauth_token=<token>

где вместо **<catalogName>** следует подставить название каталога,

вместо **<catalogId>** идентификатор категории товара,

вместо **<goodsId>** идентификатор товара,

а вместо **<token>** временный код.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.6 ПОЛУЧЕНИЕ ОПИСАНИЯ ТОВАРА ПО АРТИКУЛУ

Метод аналогичен получению описания товара из вышеописанного пункта, но требует только один параметр на вход.

Для получения списка характеристик товара (описания) по идентификатору товара используется **метод goodsDescriptionByUidAction**.

В качестве входных параметров принимаются следующие данные:

goodsId	Идентификатор товара
---------	----------------------

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с **get запросом** по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/goodsDescriptionByUid/<goodsId>.xml?oauth_token=<token>

вместо **<goodsId>** - идентификатор товара.

2.2.7 ПОЛУЧЕНИЕ ОПИСАНИЙ ТОВАРОВ ПО КАТЕГОРИИ

Для получения списка характеристик товаров (описаний) по категории используется **метод goodsDescriptionByCategoryAction**. В связи с тем, что описания товаров крупных категорий состоят из большого количества информации, на один запрос могут быть получены только характеристики 300 товаров. Поле **<id>** категории вернет количество «страниц» с описаниями товаров. Например:

```
<id>1357395 [Page 3 of 4]</id>
```

Для первого запроса по определенной категории рекомендуется указывать параметр **pageNumber = 1**.

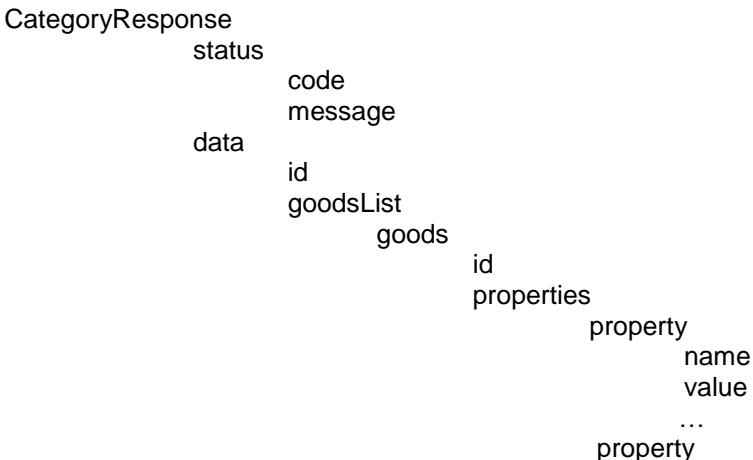
В качестве входных параметров принимаются следующие данные:

categoryId	Идентификатор категории товаров
pageNumber	Номер страницы

Метод возвращает следующие данные:

id	Уникальный идентификатор товара
properties	Список свойств товара. Содержит информацию о характеристиках товара.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с **get запросом** по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/goodsDescriptionByCategory/<categoryId>/<pageNumber>.xml?oauth_token=<token>

где вместо **<categoryId>** нужно указать идентификатор категории товара, вместо **<pageNumber>** «страницу» описаний (pageNumber = 1 для первого запроса), а вместо **<token>** временный код.

2.2.8 ПОЛУЧЕНИЕ ИЗОБРАЖЕНИЙ ТОВАРА

Для получения информации о файлах с изображениями товара и их местонахождении по идентификатору товара используется метод **goodsImagesAction**.

В качестве входного параметра принимается **goodsId - Идентификатор товара**.

Метод возвращает следующие данные:

Id	Номер изображения в списке
properties	Список свойств изображения. Содержит информацию о размере, дате обновления и расположении ² файла с изображением и пр.

² Для получения файлов с изображениями товаров, их необходимо скачать по указанным адресам (значение адреса файла хранится в свойстве с названием «Url»)

Использование логотипа Вашей компании в изображениях товаров

Для использования картинок товаров с Вашим логотипом в режиме онлайн предварительно нужно:

- Нарисовать логотип, который будет стоять на картинке на вашем сайте. Логотип нужен в формате GIF. Размер 130x40 точек.
 - Выслать логотип *своему менеджеру* в нашей компании, с указанием названия организации и номера кода клиента, присвоенного вашей организацией в нашей системе. Узнать контактные данные своего менеджера и код клиента можно в нашей b2b системе [nldealer online \(https://web-nld.netlab.ru \)](https://web-nld.netlab.ru) в разделе Сервис → Информация о клиенте.
 - Мы получим ваш логотип, сделаем привязку к картинкам, и Вы сможете пользоваться картинками товаров с Вашим логотипом.
- Получить доступ к картинкам с Вашим логотипом можно следующим образом. В текущем XML-файле <http://www.netlab.ru/products/PriceXML.zip> для картинок указан, например, следующий URL: <http://www.netlab.ru/ISAPI/TestISAPI.dll?3299&0>
- Если последний символ "0" заменить, на код XXXXXXXX (высыпается менеджером после получения Вашего логотипа), то будет картинка с Вашим логотипом.

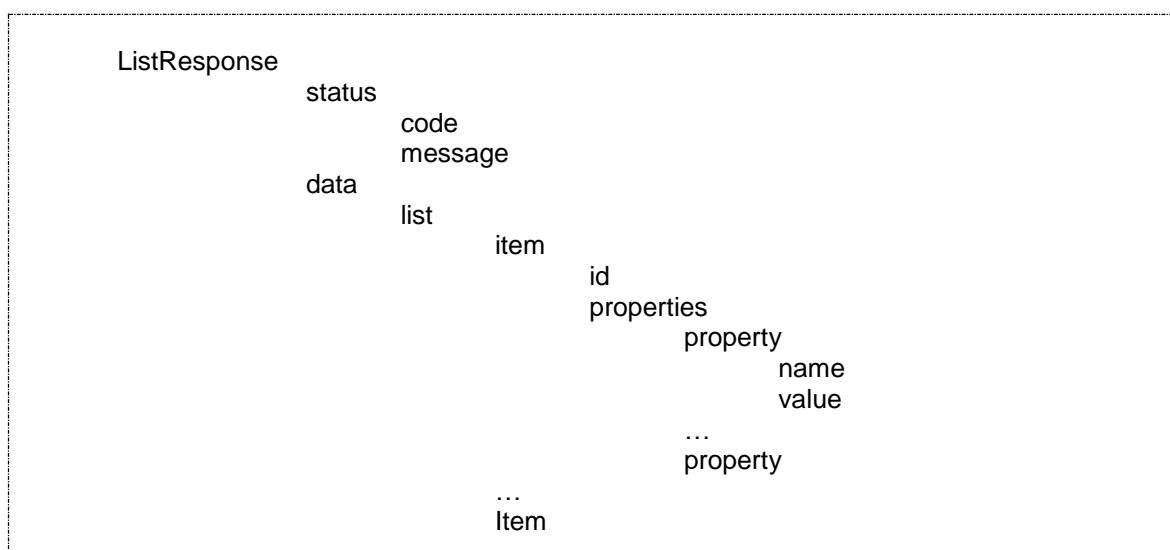
Пример из price.xml для товара с артикулом 164397 и для кода, присланного Вашим менеджером – 999555.

```
<offer id="64397" OrderBy="140729">
    <uid>164397</uid>
    <url>http://www.netlab.ru/descr.asp?id=64397</url>
    <priceR>32.9</priceR>
    <priceB>29.3</priceB>
    <priceC>27.9</priceC>
    <priceD>27.6</priceD>
    <priceE>27.3</priceE>
    <priceF>27.0</priceF>
    <priceRRP>0.0</priceRRP>
    <currencyId>USD</currencyId>
    <categoryId>64393</categoryId>
    <picture>http://www.netlab.ru/ISAPI/TestISAPI.dll?18253&0</picture>
...
...
</offer>
```

Ссылка будет такая:

<http://www.netlab.ru/ISAPI/TestISAPI.dll?18253&999555>.

ВОЗВАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/goodsImages/<goodsId>.xml?oauth_token=<token>

где вместо **<goodsId>** следует подставить идентификатор товара

а вместо **<token>** временный код

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP». Дополнительный пример кода 1С для управляемых форм для работы с изображениями.

2.2.9 ПОЛУЧЕНИЕ ИЗОБРАЖЕНИЙ ТОВАРОВ ПО КАТЕГОРИИ

Для получения информации о файлах с изображениями товаров и их местонахождении по категории товара используется метод **goodsImagesByCategoryAction**.

В качестве входного параметра принимается **categoryId - Идентификатор категории**.

Метод возвращает следующие данные:

Id	Номер изображения в списке
properties	Список свойств изображения. Содержит информацию о дате обновления и расположении ² файла с изображением

² Для получения файлов с изображениями товаров, их необходимо скачать по указанным адресам (значение адреса файла хранится в свойстве с названием «Url»)

ВОЗВАЩАЕМАЯ СТРУКТУРА ДАННЫХ:

```

CategoryResponse
  status
    code
    message
  data
    id
    goodsList
      goods
        id
        properties
          property
            name
            value
          ...
          property
  ...

```

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/goodsImagesByCategory/<categoryId>.xml?oauth_token=<token>

где вместо **<categoryId>** следует подставить идентификатор товара

а вместо **<token>** временный код

2.2.10 ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ЦЕНАХ И КОЛИЧЕСТВЕ ТОВАРА ПО КОДУ ТОВАРА

Для получения информации о наличии и ценах товара по идентификатору товара используется метод **goodsByUidAction**.

В качестве входного параметра принимается **goodsId** - Идентификатор товара. Возвращаемая информация идентична методу **categoryAction**.

Возвращаемая структура данных аналогична методу **goodsAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/goodsByUid/<goodsId>.xml?oauth_token=<token>

где вместо **<goodsId>** следует подставить идентификатор товара

а вместо **<token>** временный код

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.11 ПОЛУЧЕНИЕ СПИСКА НОВЫХ ТОВАРОВ

Для получения списка всех новых товаров используется метод **newGoodsAction**.

Метод не принимает входных параметров.

Возвращаемая информация, в том числе структура данных, идентична методу **categoryAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/newGoods.xml?oauth_token=<token>

где вместо < token > следует подставить временный код

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С».

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.12 ПОЛУЧЕНИЕ СПИСКА УЦЕНЕННЫХ ТОВАРОВ

Для получения списка всех уцененных товаров используется метод **usedGoodsAction**.

Метод не принимает входных параметров.

Метод не принимает входных параметров. Возвращаемая информация, в том числе структура данных, идентична методу **categoryAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/usedGoods.xml?oauth_token=<token>

где вместо < token > следует подставить временный код

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С».

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.13 ПОИСК ТОВАРА ПО PART NUMBER

Для получения информации о товаре по известному part number используется метод **goodsByPartnumber**.

В качестве входного параметра принимается **part number** товара.

Возвращаемая структура данных аналогична методу **categoryAction** (получение списка товаров категории) только для одного товара.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/goodsByPartnumber/{partnumber}.xml?oauth_token=<token>

где вместо **<token>** следует подставить временный код вместо **<partnumber>** - part number товара.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.14 ПОИСК ДАТЫ НАКЛАДНОЙ ПО НОМЕРУ СТИКЕРА

Для получения даты накладной по номеру стикера используется метод **getDateByStickerAction**.

В качестве входного параметра принимается номер стикера.

Метод возвращает список свойств, содержащий дату накладной, если таковая имеется.

Возвращаемая структура данных аналогична методу **goodsImagesAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/dateBySticker.xml?oauth_token=<token>&sticker=<stickerNumber>

где вместо **< token >** следует подставить временный код

вместо **<stickerNumber>** - номер стикера.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.15 ПОИСК ДАТЫ НАКЛАДНОЙ ПО СЕРИЙНОМУ НОМЕРУ ТОВАРА.

Для получения даты накладной по серийному номеру товара используется метод **getDateBySerialNumberAction**.

В качестве входных параметров принимается серийный номер товара.

Метод возвращает список свойств, содержащий дату накладной, если таковая имеется.

Возвращаемая структура данных аналогична методу **goodsImagesAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА.

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/dateBySerialNumber.xml?oauth_token=<token>&serialnumber=<serialNumber>

где вместо **<token>** следует подставить временный код
вместо **<serialNumber>** - серийный номер товара.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С».
Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.16 ПОЛУЧЕНИЕ ПРОЧЕЙ ИНФОРМАЦИИ (КУРСЫ ВАЛЮТ)

Для получения дополнительной информации, необходимой для работы, используется метод **getInfoAction**.

В настоящий момент возвращаемые данные содержат информацию о курсах валют, используемых в нашей компании. В дальнейшем информация может пополняться.

Метод не принимает входных параметров.

Возвращаемая структура данных аналогична методу **goodsImagesAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/info.xml?oauth_token=<token>

где вместо **<token>** следует подставить временный код.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С».
Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.2.17 ПОИСК ТОВАРОВ ПО КЛЮЧЕВЫМ СЛОВАМ

Для получения списка товаров с основными характеристиками по ключевым словам используется метод **getGoodsSearch**.

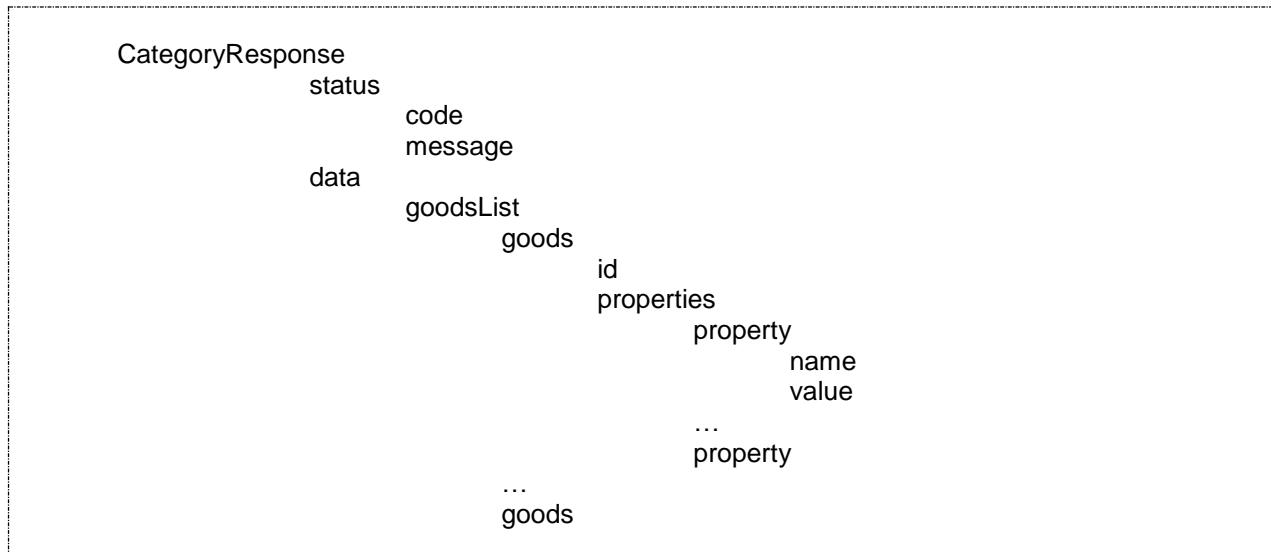
В качестве входного параметра принимаются до 5 слов, разделенные между собой знаком «::» (двоеточие).

Пример: « apple:retina:белый »

Метод возвращает список товаров, которые были найдены в результате поиска по ключевым словам. Количество возвращенных товаров не может превышать 500 позиций. Для более точного поиска используйте больше ключевых слов.

Возвращаемая структура данных аналогична методу **goodsImagesAction**.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/catalogsZip/getGoodsSearch/{descr}.json?oauth_token=<token>

, где вместо `< token >` следует подставить временный код

вместо `{descr}` – ключевые слова через знак «:» .

2.3 МОДУЛЬ «ДОКУМЕНТЫ»

Данный модуль включает в себя функциональность необходимую для работы с документами.

2.3.0 ОПИСАНИЕ МОДУЛЯ

Методы этого модуля описываются следующими URL:

<http://services.netlab.ru/DocumentsServiceZip?wsdl> – дескриптор, методы которого возвращают ответ в сжатом виде (gzip).

<http://services.netlab.ru/DocumentsService?wsdl> – дескриптор, методы которого возвращают ответ в не сжатом виде.

<http://services.netlab.ru/rest/documentsZip/> * - RESTful методы с ответом в сжатом виде. <http://services.netlab.ru/rest/documents/> * - RESTful методы с ответом в не сжатом виде.

Т.к. объем возвращаемых данных у методов этого модуля небольшой, то выбор использовать или нет сжатие, необходимо сделать после практического тестирования. Дело в том, что сжатие на сервере может занимать больше времени,

чем вы сэкономите на пересылке (например, 5Кб и 500 байт). В то время как для модуля «Каталог» рекомендуется использовать сжатие (там речь идет на мегабайты данных и сжатие оправдано).

2.3.1 ПОЛУЧЕНИЕ СПИСКА ЮРИДИЧЕСКИХ ЛИЦ

Создаваемые клиентом нашей компании документы привязываются к юридическому лицу (организации). Один клиент может быть представлен одним и более юридическим лицом. Каждой организации ставится в соответствие уникальный идентификатор. В дальнейшем, при формировании документов (резервов, счетов и т. д.), потребуется указывать идентификатор организации.

Для получения списка организаций представляющих клиента используется метод **getOrganizationsAction**.

Метод не принимает входных параметров.

Метод возвращает следующие данные:

id	Код клиента.
properties	Список свойств организаций. Содержит идентификаторы и названия организаций.

Возвращаемая структура данных аналогична методу **goodsImagesAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/organizations.xml?oauth_token=<token>

где вместо < token > следует подставить временный код

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.2 ПОЛУЧЕНИЕ СПИСКА ТИПОВ ДОСТУПНЫХ ДОКУМЕНТОВ

Документы, представленные в документообороте компании, разбиты на типы. Все доступные для клиентов типы документов представлены в нашей **b2b системе NLDealer online**, например: резервы, счета, отгрузки...

При работе с сервисами, для указания типа документа, используется соответствующий данному типу идентификатор.

Для получения полного списка типов документов с их идентификаторами используется метод **getDocumentTypesAction**.

Метод не принимает входных параметров.

Возвращаемая структура данных аналогична методу [goodsImagesAction](#).

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/documentTypes.xml?oauth_token=<token>

где вместо < token > следует подставить временный код

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.3 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ

Для получения списка документов используется **метод listDocuments**.

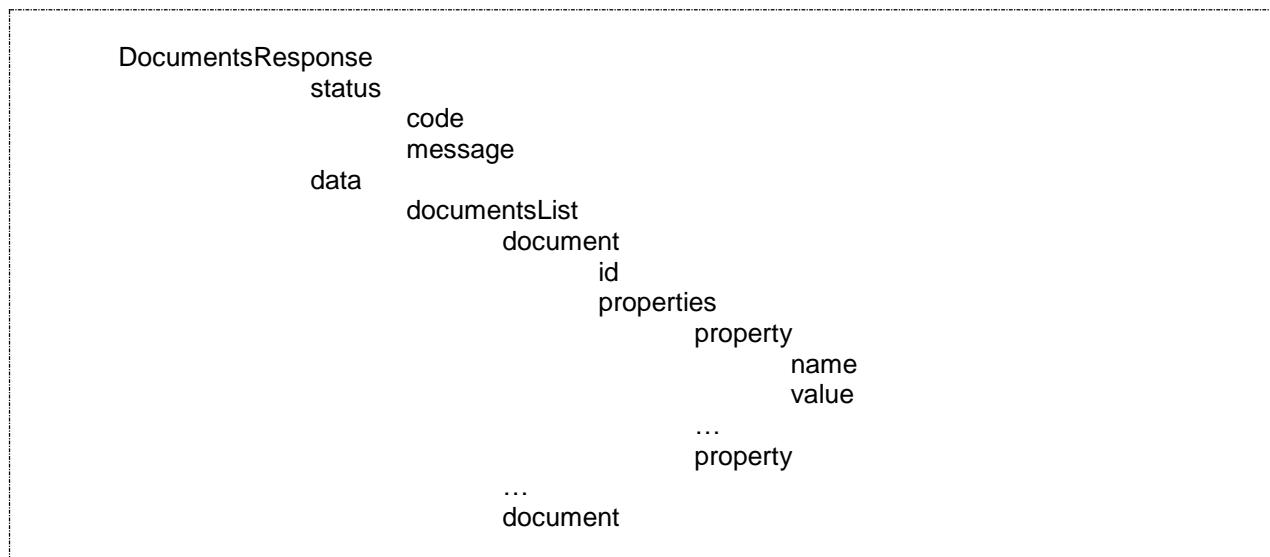
Получить список документов определенного типа можно за указанный интервал времени (Исключение составляют «резервы»).

В качестве входных параметров принимаются следующие данные:

documentTypeId	Тип документов
beginDate	Начало интервала времени. ³
endDate	Конец интервала. ³

³Дата указывается в формате «YYYYMMDD», где YYYY-год, MM-месяц, DD-день. Пример 20150207 – 7 февраля 2015 г.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/list.xml?oauth_token=<token>&documentTypeId=<typeld>&beginDate=<beginInterval>&endDate=<endInterval>

где вместо `<token>` следует подставить временный код,

вместо `<typeld>` - тип документа,

`<beginInterval>` - начальная дата интервала,

`<endInterval>` - конечная дата интервала.

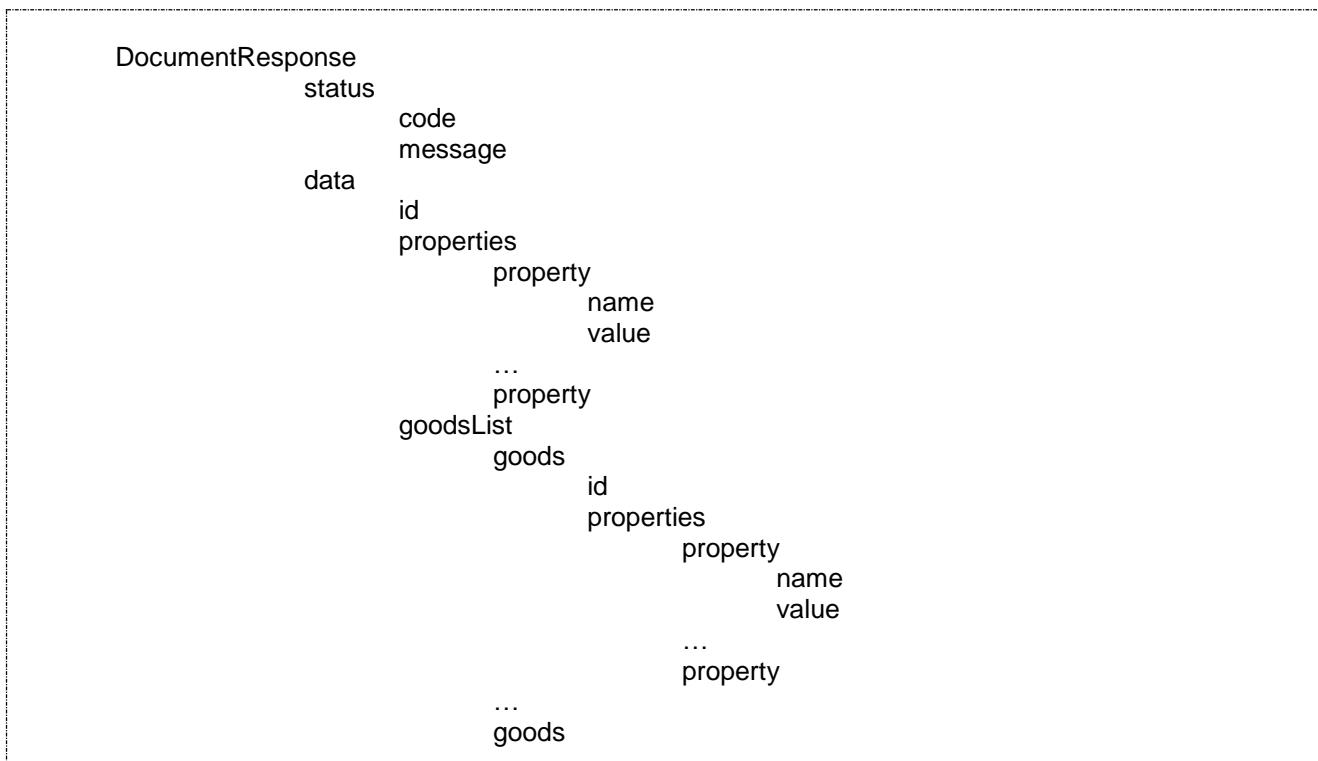
Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.4 ПОЛУЧЕНИЕ СПИСКА ТОВАРОВ ДОКУМЕНТА

Для получения списка позиций (в частном случае товаров) и прочей информации о документе, такой как тип, дата, организация и т.п. используется метод `readAction`.

Метод принимает в качестве входного параметра номер документа.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с `get` запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/get.xml?oauth_token=<token>&documentId=<Id>

где вместо `<token>` следует подставить временный код,

вместо `<id>` - номер документа.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.5 СОЗДАНИЕ ДОКУМЕНТОВ

Клиенты компании могут самостоятельно создавать некоторые типы документов, такие как счета и резервы.

Для создания документа используется метод `createAction`.

В качестве входных параметров принимаются следующие данные:

<code>documentTypeId</code>	Тип документа
<code>organizationId</code>	Идентификатор организации
<code>date</code>	Дата ³
<code>comment</code>	Комментарий
<code>goodsString</code>	Список товаров ⁴

³ Дата указывается в формате «YYYYMMDD», где YYYY-год, MM-месяц, DD-день. Пример 20150207 – 7 февраля 2015 г.
Максимальный срок документа может быть ограничен (для резерва 2 дня, если товары в наличии на складе)

⁴ Список товаров представляется в виде строки, содержащей артикулы товаров и их количество. Информация, относящаяся к одному товару, разделена запятой, а к различным товарам – точкой с запятой. Пример: «1254130,1;1128362,3;» - 1 единица товара с артикулом 1254130 и 3 единицы с артикулом 1128362.

Возвращаемая структура данных аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/new.xml?oauth_token=<token>&documentTypeId=<documentType>&organizationId=<org>&date=<date>&comment=<comment>&goodsString=<goods>

где вместо <**token**> следует подставить временный код,

вместо <**documentType**> - тип документа,

вместо <**org**> - идентификатор организации,

вместо <**date**> - дата документа ^{[3](#)}

вместо <**comment**> - комментарий,

вместо <**goods**> - Список товаров ^{[4](#)}

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С».

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.6 РЕДАКТИРОВАНИЕ ДОКУМЕНТОВ

Для редактирования документа используются методы **updateAction** и **changeAction**.

Метод **updateAction** служит для полного обновления (перезаписи) информации в документе.

Метод **changeAction** предназначен для частичного редактирования документа без передачи излишней (всего содержимого документа) информации. С его помощью можно осуществлять такие логические операции как:

- Вставка или удаление строк (товарных позиций) документа.
- Изменение количества присутствующей в документе товарной позиций (как в сторону увеличения, так и уменьшения).

Рассмотрим метод **updateAction**.

В качестве входных параметров принимаются следующие данные:

documentTypeId	Тип документа
----------------	---------------

documentId	Номер документа
organizationId	Идентификатор организации
date	Дата ⁴
comment	Комментарий
goodsString	Список товаров ⁵

Метод **changeAction** принимает следующие входные параметры:

documentId	Номер документа
documentTypeId	Тип документа
goodsString	Список изменений по товарным позициям ⁵

Возвращаемая структура данных для обоих методов аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить методы из браузера можно обратившись с get запросом по следующему URL-адресу:

updateAction:

http://services.netlab.ru/rest/documents/update.xml?oauth_token=<token>&documentType=Id=<documentType>&documentId=<docId>&organizationId=<org>&date=<date>&comment=<comment>&goodsString=<goods>

changeAction:

http://services.netlab.ru/rest/documents/change.xml?oauth_token=<token>&documentType=Id=<documentType>&documentId=<docId>&goodsString=<goodsChanges>

где вместо **<token>** следует подставить временный код,

вместо **<documentType>** - тип документа,

вместо **<docId>** - номер документа,

вместо **<org>** - идентификатор организации,

вместо **<date>** - дата документа ³

вместо **<comment>** - комментарий,

вместо **<goods>** - Список товаров ⁴

вместо **<goodsChanges>** - Список изменений по товарам. Формат аналогичен **<goods>**.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.7 УДАЛЕНИЕ ДОКУМЕНТОВ

Для удаления документов используется метод **deleteAction**.

В качестве входных параметров принимаются следующие данные:

documentTypeId	Тип документа
documentId	Номер документа

Возвращаемая структура данных аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/delete.xml?oauth_token=<token>&documentType=*&documentId=<docId>

где вместо **<token>** следует подставить временный код,

вместо **<documentType>** - тип документа,
вместо **<docId>** - номер документа.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С». Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.8 СОЗДАНИЕ НОВОГО РЕЗЕРВА

Для создания нового резерва используется метод **createReserve**.

В качестве входных параметров принимаются следующие данные:

organizationId	Идентификатор организации
date	Дата 3
goodString	Список товаров 5
reservePrice	Резервировать цену. 0 – нет, 1 - да. Можно не заполнять (по умолчанию 0)

Возвращаемая структура данных аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/newReserve.xml?oauth_token=<token>&organizationId=<organization>&date=<date>&comment=<comment>&goodsString=<goods>&reservePrice=<reservePrice>

где вместо <**token**> следует подставить временный код,

вместо <**organization**> - идентификатор организации,

вместо <**date**> - дата документа ^{[3](#)},

вместо <**goods**> - Список товаров ^{[4](#)}

вместо <**comment**> - комментарий

вместо <**reservePrice**> - 0 или 1. 0 – не резервировать цену, 1 – резервировать цену.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «1С».

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.9 ЗАКАЗ ОТСУТСТВУЮЩИХ ТОВАРОВ

Для уточнения информации по отсутствующим товарам и дальнейшего их заказа используется метод **orderGoods**. В зависимости от того, какой статус будет у заказного товара, будет выполнено одно из доступных действий (уточнение в течение 1 рабочего дня, в течение 2 рабочих часов, статус «Привезти» и т.д.)

Примечание: Если поле «**orderStatus**» пустое, то либо товар есть/появился на складе, либо недоступен к заказу.

В качестве входных параметров принимаются следующие данные:

documentId	Номер документа
goodsUid	Артикул товара

Возвращаемая структура данных аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/orderGoods.json?oauth_token=<token>&documentId=<documentId>&goodsUid=<goodsUid>

где вместо <**token**> следует подставить временный код,

вместо <**documentId**> - номер документа,

вместо <**goodsUid**> - артикул товара.

2.3.10 ЗАПРОС СПИСКА ТОВАРОВ В РЕМОНТЕ

Для запроса нового резерва используется метод **repairGoodsList**.

В качестве входных параметров принимаются следующие данные:

organizationId	Идентификатор организации
date	Дата ⁴
goodString	Список товаров ⁵
reservePrice	Резервировать цену. 0 – нет, 1 - да. Можно не заполнять (по умолчанию 0)

Возвращаемая структура данных

```

DocumentResponse
status
code
message
data
list
item
id
properties
property
name
value
...
data

```

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/repairGoodsList.xml?oauth_token=<token>

где вместо **<token>** следует подставить временный код.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.11 ЗАПРОС ДАННЫХ ДОКУМЕНТА ОТГРУЗКИ

Для запроса данных документа отгрузки используется метод **readShipment**.

В качестве входных параметров принимаются следующие данные:

organizationId	Идентификатор организации
documentId	Идентификатор документа

Возвращаемая структура данных аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/readShipment.xml?oauth_token=<token>&documentId=<document>

где вместо **<token>** следует подставить временный код, вместо **<document>** - идентификатор документа.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.12 РАЗБИЕНИЕ РЕЗЕРВА НА ДВА ДОКУМЕНТА

Для разделения резерва на два отдельных документа используется метод **splitReserve**.

В качестве входных параметров принимаются следующие данные:

documentId	Идентификатор документа
goodsId	Артикулы товаров через символ ;

Возвращаемая структура данных аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/splitReserve.xml?oauth_token=<token>&documentId=<document>&goodsString=<goodsId>

где вместо **<token>** следует подставить временный код, вместо **<documentId>** - идентификатор документа, вместо **<goodsId>** - артикулы товаров через точку с запятой ; .

Метод возвращает данные двух методов.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.3.13 ОБЪЕДИНЕНИЕ РЕЗЕРВОВ В ОДИН ДОКУМЕНТ

Для разделения резерва на два отдельных документа используется метод **mergeReserves**. При объединении резерва второй документ удаляется, и все его товары перемещаются в первый документ.

В качестве входных параметров принимаются следующие данные:

document1Id	Идентификатор документа 1
document2Id	Идентификатор документа 2

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/mergeReserves.xml?oauth_token=<token>&document1Id=<document1Id>&document2Id=<document2Id>

где вместо <token> следует подставить временный код,
вместо <document1Id> - идентификатор первого документа,
вместо <document2Id> - идентификатор второго документа.

Метод возвращает статус выполнения операции.

2.3.14 ПРОДЛЕНИЕ РЕЗЕРВА

Для продления резерва используется метод **prolongReserve**. Продлевать резерв можно на ограниченный срок (в случае превышения возможного срока резервирования, программа автоматически выставит максимально отдаленную дату).

Примечание: каждый товар в резерве имеет свою дату удаления. Если она пройдет, то несмотря ни на что, товар будет удален из резерва автоматически.

В качестве входных параметров принимаются следующие данные:

documentId	Номер документа
date	Дата ⁴

Возвращаемая структура метода аналогична методу **readAction**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/documents/prolongReserve.xml?oauth_token=<token>&documentId=<documentId>&date=<date>

где вместо <token> следует подставить временный код,
вместо <documentId> - номер документа,
вместо < date > - дата документа ³

2.4 МОДУЛЬ «ДОСТАВКА»

Модуль содержит методы для работы с документами доставки.

2.4.0 ОПИСАНИЕ МОДУЛЯ

Методы этого модуля описываются следующими URL:

<http://services.netlab.ru/DeliveryServiceZip?wsdl> – дескриптор, методы которого возвращают ответ в сжатом виде (gzip).

<http://services.netlab.ru/DeliveryService?wsdl> – дескриптор, методы которого возвращают ответ в не сжатом виде.

<http://services.netlab.ru/rest/deliveryZip/> * - RESTful методы с ответом в сжатом виде.

<http://services.netlab.ru/rest/delivery/> * - RESTful методы с ответом в не сжатом виде.

2.4.1 ПОЛУЧЕНИЕ СПИСКА ДОКУМЕНТОВ ДОСТАВКИ

Для получения списка документов доставки используется метод **метод listDocuments**.

Получить список документов доставки необходимо указать интервал времени.

В качестве входных параметров принимаются следующие данные:

beginDate	Начало интервала времени. ³
endDate	Конец интервала. ³

³Дата указывается в формате «YYYYMMDD», где YYYY-год, MM-месяц, DD-день. Пример 20150207 – 7 февраля 2015 г.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:

```

DocumentsResponse
  status
    code
    message
  data
    documentsList
      document
        id
        properties
          property
            name
            value
          ...
          property
        ...
      document
    ...
  ...

```

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/list.xml?oauth_token=<token>&beginDate=<beginInterval>&endDate=<endInterval>

где вместо **<token>** следует подставить временный код,

<beginInterval> - начальная дата интервала,

<endInterval> - конечная дата интервала.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.4.2 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ СЧЕТОВ ДЛЯ ОФОРМЛЕНИЯ ДОСТАВКИ

Чтобы получить список документов, на основании которых можно оформить документ доставки предусмотрен метод **availableDocuments**.

В качестве входных параметров принимаются следующие данные:

beginDate	Начало интервала времени.
endDate	Конец интервала.

Возвращаемая структура данных аналогична методу **listDocuments** из вышеописанного пункта.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/docs.xml?oauth_token=<token>&beginDate=<beginInterval>&endDate=<endInterval>

где вместо **<token>** следует подставить временный код,

<beginInterval> - начальная дата интервала,

<endInterval> - конечная дата интервала.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.4.3 ПОЛУЧЕНИЕ ПОДРОБНОЙ ИНФОРМАЦИИ ДОКУМЕНТА ДОСТАВКИ

Для получения подробной информации по документу доставки используется метод **details**.

В качестве входных параметров принимаются следующие данные:

packId	Номер упаковки (например, 987456)
--------	-----------------------------------

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:

```

DetailsResponse
  status
    code
    message
  data
    properties
      property
        name
        value
      ...
      property
      ...
    properties
    goodsList
      properties
        property
          name
          value
      ...
      property
      ...
    properties
  data

```

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/details.xml?oauth_token=<token>&packId=<packId>

где вместо <**token**> следует подставить временный код,

<**packId**> - номер упаковки.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

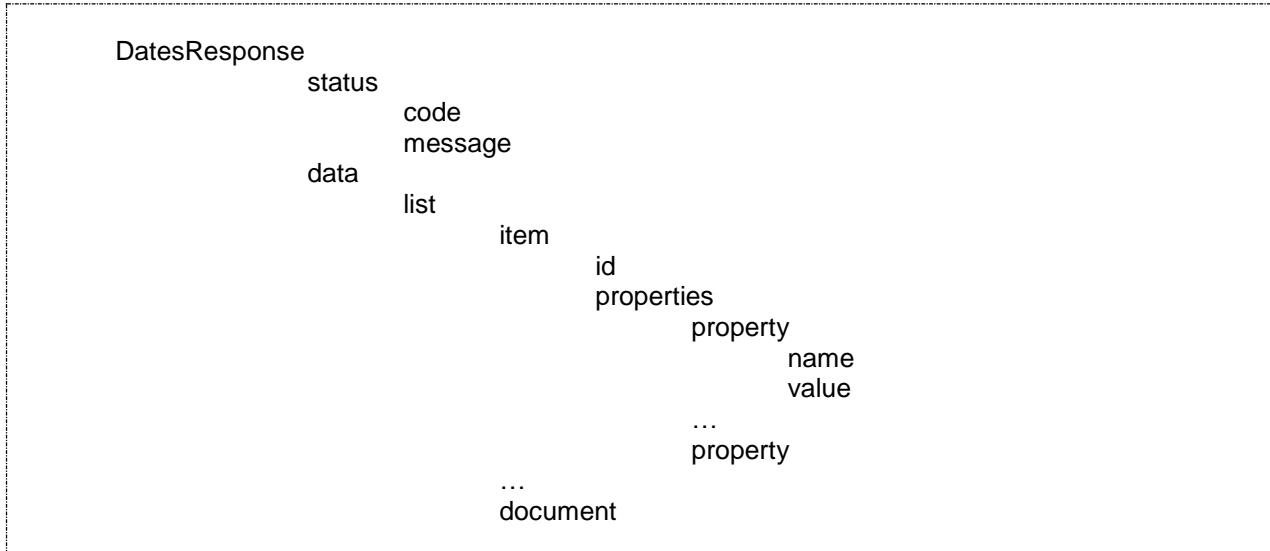
2.4.4 ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ ДАТ ДОСТАВКИ

Для получения списка доступных дат доставки используется метод **dates**.

В качестве входных параметров принимаются следующие данные:

beginDate	Начало интервала времени.
endDate	Конец интервала.

ВОЗВРАЩАЕМАЯ СТРУКТУРА ДАННЫХ:



ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/dates.xml?oauth_token=<token>&beginInterval=<beginInterval>&endDate=<endInterval>

где вместо <token> следует подставить временный код,

<beginInterval> - начальная дата интервала,

<endInterval> - конечная дата интервала.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.4.5 ПОЛУЧЕНИЕ ВОЗМОЖНЫХ ТИПОВ ДОСТАВКИ

Список возможных типов доставки возвращается **методом deliveryTypes**.

Метод не требует входных параметров (кроме временного кода).

Возвращаемая структура данных аналогична методу **dates**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/deliveryTypes.xml?oauth_token=<token>

где вместо **<token>** следует подставить временный код.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.4.6 ПОЛУЧЕНИЕ СПИСКА ВАРИАНТОВ ОПЛАТЫ ДОСТАВКИ

Для получения списка вариантов оплаты доставки используется **метод paymentTypes**.

Метод не требует входных параметров (кроме временного кода).

Возвращаемая структура данных аналогична методу **dates**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/paymentTypes.xml?oauth_token=<token>

где вместо **<token>** следует подставить временный код.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.4.7 ПОЛУЧЕНИЕ СПИСКА ТАРИФОВ ДОСТАВКИ

Для получения списка вариантов оплаты доставки используется **метод rates**.

В качестве входных параметров принимаются следующие данные:

documentIds	ID документов через символ ;
deliveryDate	Дата доставки
deliveryType	Тип доставки

Возвращаемая структура данных аналогична методу **dates**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/rates.xml?oauth_token=<token>&documentIds=<documentIds>&deliveryDate=<deliveryDate>&deliveryType=<deliveryType>

где вместо **<token>** следует подставить временный код,

<documentIds> - номера документов через точку с запятой, на основании которых оформляется доставка,

<deliveryDate> - дата доставки,

<deliveryType> - тип доставки.

Перейти к примеру запроса и обработки результатов ответа сервера на языке «PHP».

2.4.8 СОХРАНЕНИЕ ДОКУМЕНТА ДОСТАВКИ

Для сохранения документ доставки используется **метод save**.

В качестве входных параметров принимаются следующие данные:

id	
documentIds	ID документов через символ ;
deliveryDate	Дата доставки
deliveryType	Тип доставки
ratelId	
clientId	
typeCode	
paymentTypeId	
paymentDocumentId	

Возвращаемая структура данных аналогична **методу details**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/save.xml?oauth_token=<token>&id=<id>&documentIds=<documentIds>&deliveryDate=<deliveryDate>&deliveryType=<deliveryType>&rateId=<rateId>&clientId=<clientId>&typeCode=<typeCode>&paymentTypeId=<paymentTypeId>&paymentDocumentId=<paymentDocumentId>

где вместо **<token>** следует подставить временный код,

<id> - id документа доставки (не обязательный параметр при первичном сохранении),

<documentIds> - номера документов через точку с запятой, на основании которых оформляется доставка,

<deliveryDate> - дата доставки,

<deliveryType> - тип доставки,

<rateId> - id тарифа доставки,

<clientId> - id клиента доставки (номер транспортной компании и т.п.),

<typeCode> - код тарифа доставки,

<paymentTypeId> - выбор типа оплаты доставки: «В акт», «В накладную», «С баланса»,

<paymentDocumentId> - id документа для оплаты доставки (в случае выбора «В акт» или «В накладную»). Не требуется при выборе «Оплата с баланса».

2.4.9 ОТГРУЗКА ПО ДОКУМЕНТУ ДОСТАВКИ

Для подтверждения отгрузки документа используется **метод finalize**. После выполнения метода, документ будет недоступен для редактирования или удаления.

В качестве входных параметров принимаются следующие данные:

deliveryDocumentId	Id документа доставки
--------------------	-----------------------

Возвращаемая структура данных аналогична **методу details**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/finalize.xml?oauth_token=<token>&deliveryDocumentId=<documentId>

где вместо **<token>** следует подставить временный код,

<documentId> - номер документа доставки.

2.4.10 УДАЛЕНИЕ ДОКУМЕНТА ДОСТАВКИ

Для подтверждения отгрузки документа используется **метод delete**. Удалить можно сохраненный, но не подтвержденный документ доставки.

В качестве входных параметров принимаются следующие данные:

deliveryDocumentId	Id документа доставки
--------------------	-----------------------

Возвращаемая структура данных аналогична **методу details**.

ПРИМЕР ОБРАЩЕНИЯ К МЕТОДУ ИЗ БРАУЗЕРА

Выполнить метод из браузера можно обратившись с get запросом по следующему URL-адресу:

http://services.netlab.ru/rest/delivery/delete.xml?oauth_token=<token>&deliveryDocumentId=<documentId>

где вместо **<token>** следует подставить временный код,

<documentId> - номер документа доставки.

3 ПРИМЕРЫ СКРИПТОВ ДЛЯ «1С ПРЕДПРИЯТИЕ 8.2»

Ниже приводится пример настройки программы «1С Предприятие 8.2» для работы с сервисами.

Трем модулям, описанным выше Аутентификация, Каталог, Документы соответствуют следующие **файлы-дескрипторы**:

- <http://services.netlab.ru/AuthenticationService?wsdl>
- <http://services.netlab.ru/CatalogService?wsdl>
- <http://services.netlab.ru/CatalogServiceZip?wsdl>
- <http://services.netlab.ru/DocumentsService?wsdl>
- <http://services.netlab.ru/DocumentsServiceZip?wsdl>

Описание механизмов интеграции «1С Предприятие» посредством веб-сервисов можно найти тут:

- http://www.v8.1c.ru/overview/Term_000000273.htm#1.

3.1 ПРИМЕР НАСТРОЙКИ «1С ПРЕДПРИЯТИЕ 8.2» ДЛЯ РАБОТЫ С ВЕБ-СЕРВИСАМИ. СОЗДАНИЕ ССЫЛОК НА СЕРВИСЫ

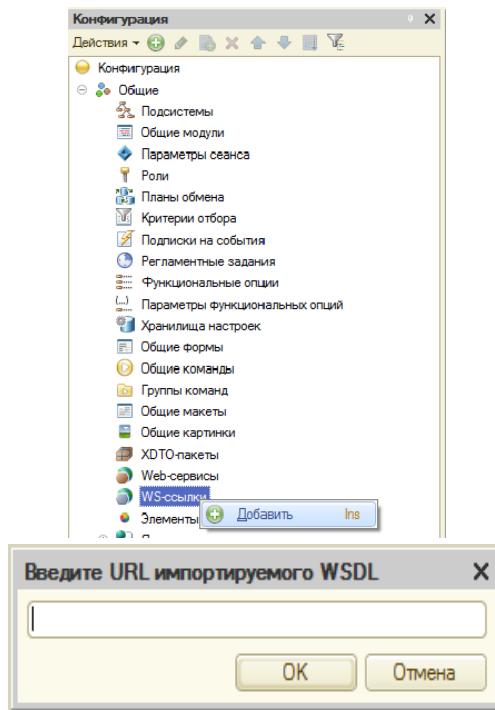
Работа с сервисами в программе «1С Предприятие» осуществляется посредством объектов типа **WS-ссылка**. WS-ссылка представляет собой WSDL описание веб-сервиса, импортированное из указанного источника.

Подробное описание представлено тут:

- http://v8.1c.ru/overview/Term_000000164.htm

Ниже приводится **последовательность действий**, которую необходимо осуществить для создания WS-ссылок.

- Запускаем программу в **режиме конфигуратора**.
- Открываем пункт «**Общие**» в дереве конфигурации.
- Выбираем «**Добавить**» в контекстном меню пункта «**WS-ссылки**».

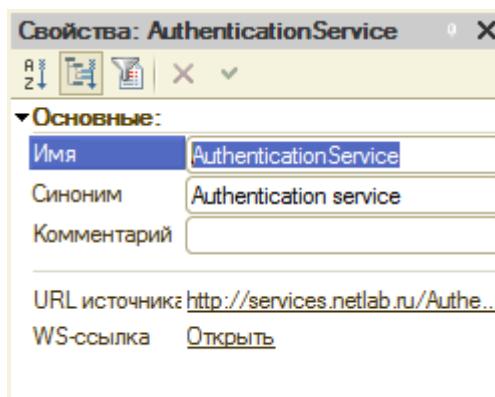


В появившемся окне вводим **адрес WSDL файла**:

- <http://services.netlab.ru/AuthenticationService?wsdl>

и нажимаем «OK».

Открываем свойства созданной ссылки и **вводим данные**, как указано ниже.

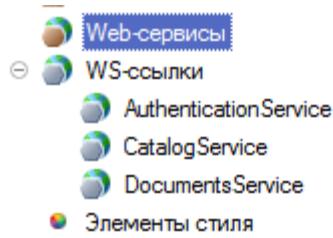


и нажимаем на зеленую галочку.

Повторяем действия 1-3 для остальных WSDL файлов:

WSDL файл описания	Название WS-ссылки
http://services.netlab.ru/DocumentsService?wsdl	DocumentsService
http://services.netlab.ru/CatalogService?wsdl	CatalogService

В итоге должно получиться так:



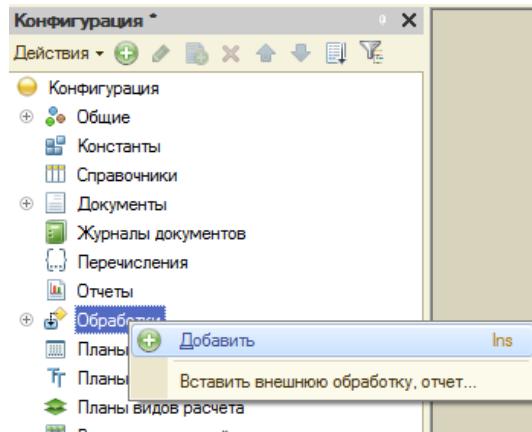
3.2 СОЗДАНИЕ ФОРМЫ ОБРАБОТКИ ДЛЯ ТЕСТИРОВАНИЯ

Создание формы обработки не является обязательным условием для использования веб-сервисов и приводится только как часть демонстрации.

Пример «качественно» сделанной формы обработки можно найти тут:

- <http://habrahabr.ru/post/148658/> .

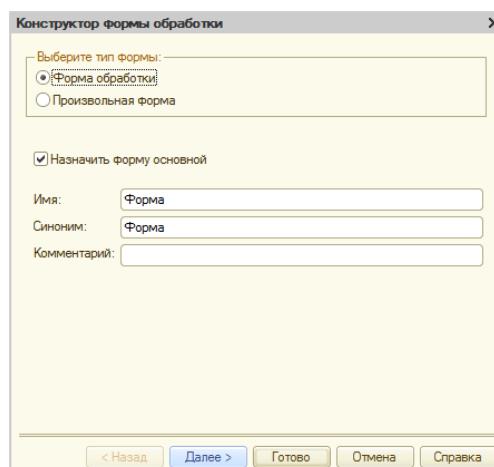
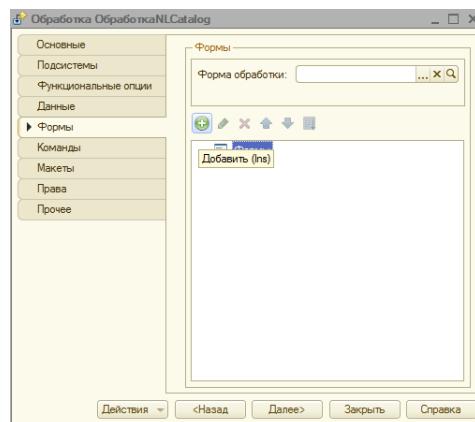
Создаем новую обработку.



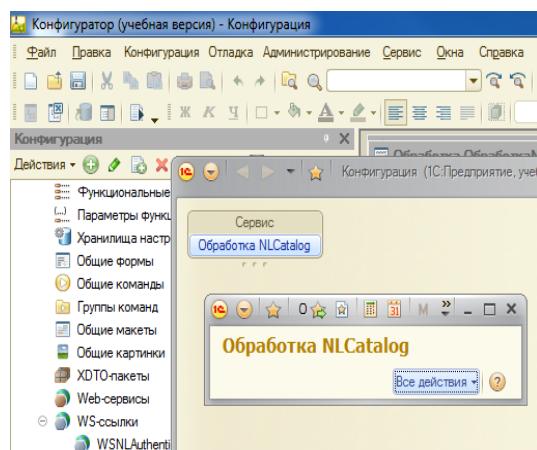
Заполняем информацию.

Имя:	ОбработкаNLCatalog
Синоним:	Обработка NLCatalog
Комментарий:	
Расширенное представление:	
Пояснение:	

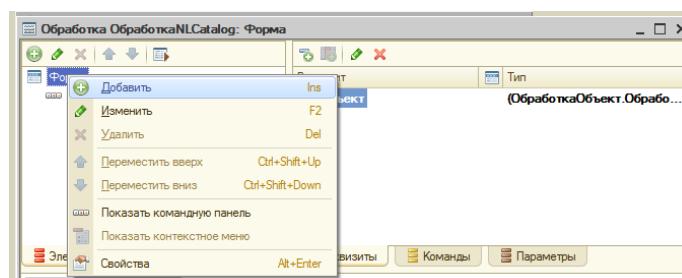
Создаем форму.

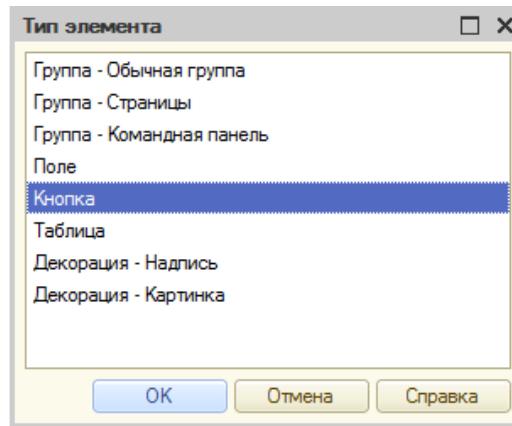


Для проверки запускаем программу, и открываем форму.

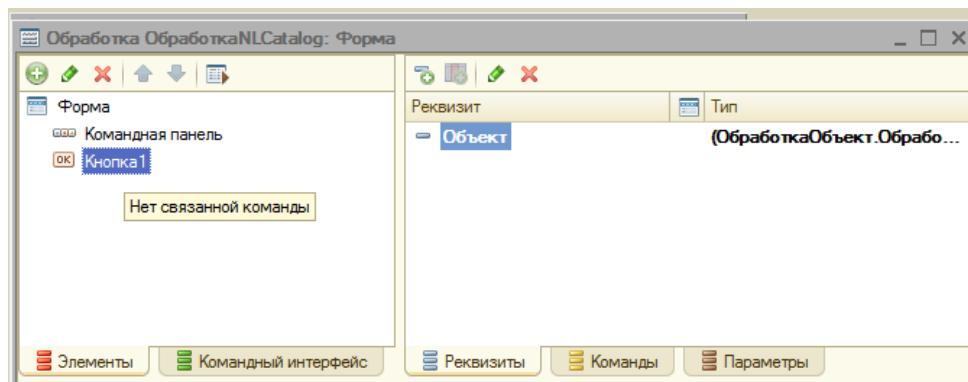


На форме обработки создаем кнопку, при нажатии на которую будут вызываться все имеющиеся на данный момент процедуры сервисов.

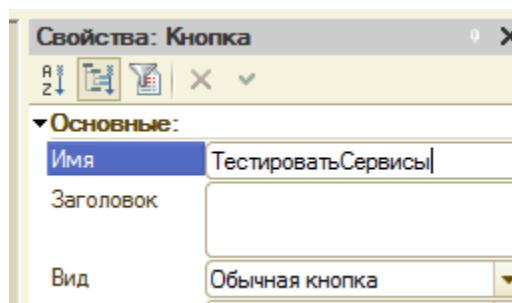




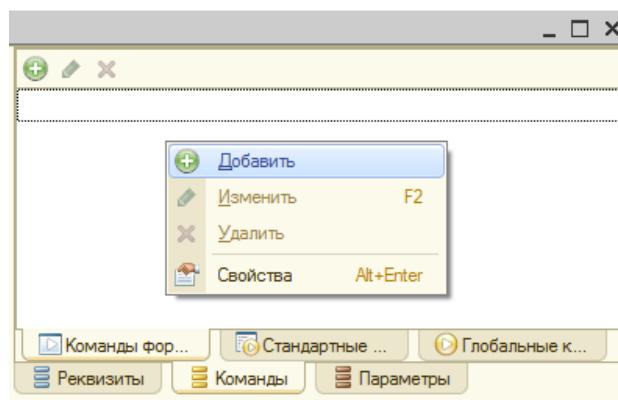
Делаем двойной клик по кнопке.



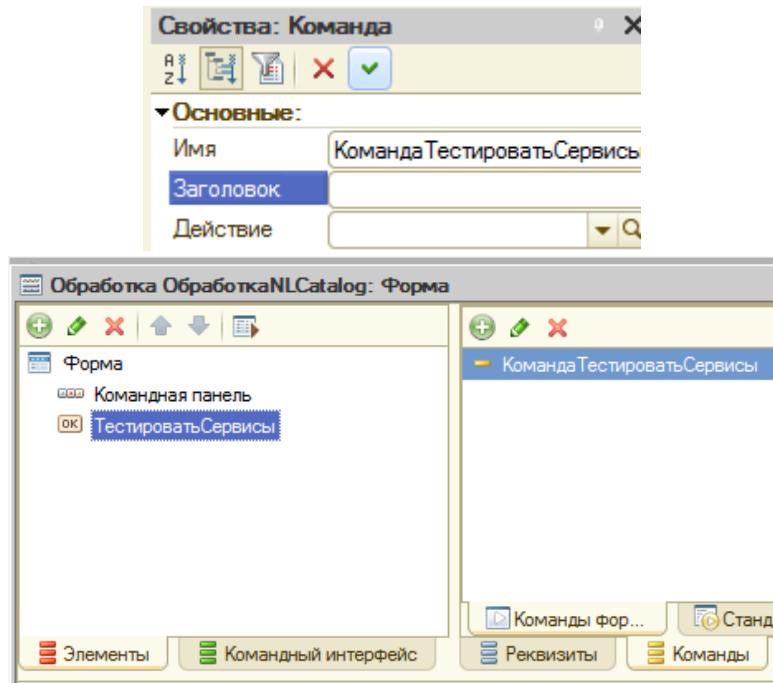
Вводим название кнопки.



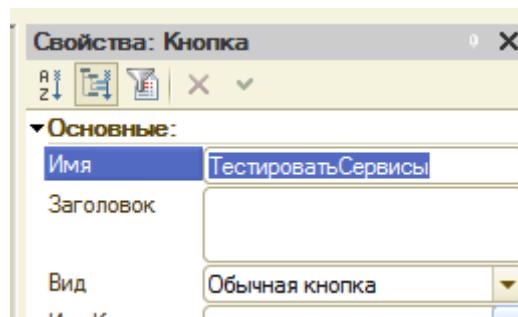
Переключаемся в табу «Команды», «Команды формы» в контекстном меню выбираем **добавить**.



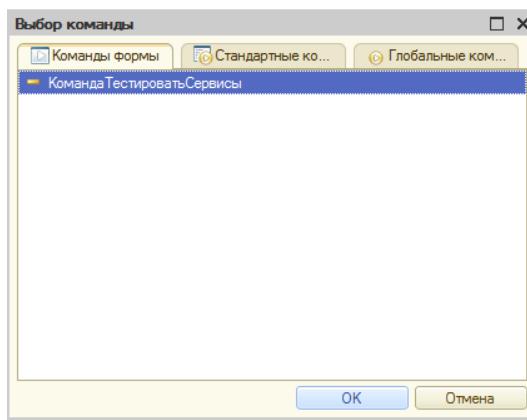
Делаем двойной клик по **названию кнопки**.



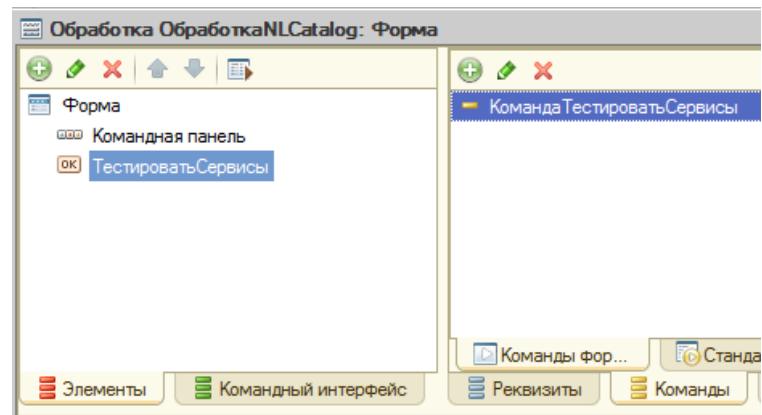
Жмем на кнопку «...» справа от ИмяКоманды.



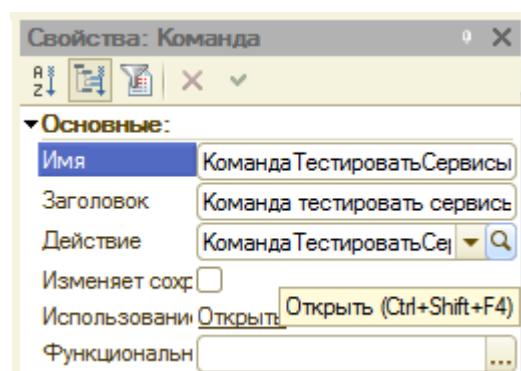
Выбираем команду, жмем «OK».



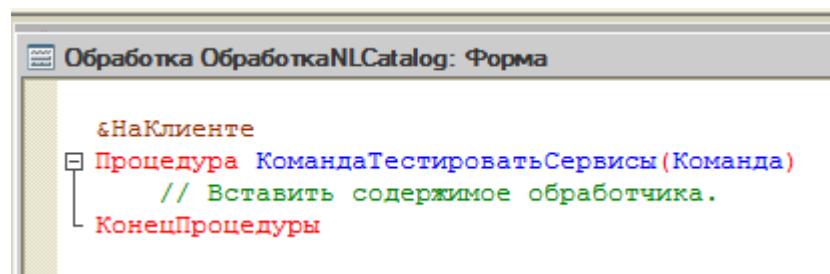
Делаем двойной клик по «КомандаТестироватьСервисы».



Жмем на кнопку с лупой справа от «Действие».



Открываем модуль с кодом.



Все готово для написания кода, приведенного в следующем пункте.

ПРИМЕРЫ СКРИПТОВ ДЛЯ ТЕСТИРОВАНИЯ С ПОМОЩЬЮ ОБРАБОТКИ

```

&НаКлиенте
Процедура Команда Тестировать Новые Скрипты(Команда)
    //Получение временного кода(токена), для дальнейшей работы
    username = "myUsername";
    password = "myPassword";
    token = authentication(username,password);
    Если token = NULL Тогда
        Сообщить("Токен не получен. Тест прерван.", СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    //Тестирование каталога и вспомогательного функционала
    testCatalog(token);
    //Тестирование документов
    testDocuments(token);
КонецПроцедуры

Процедура testCatalog(token)

```

```

//1. получение списка каталогов товаров
listOfCatalogs = getCatalogs(token);
Если listOfCatalogs = NULL Тогда
    Сообщить("Список каталогов не получен. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//вывод сообщений с названиями каталогов
//loopByArray(listOfCatalogs);

//2. Получение списка категорий товаров каталога
//Название каталога товаров (один из "Прайс-лист", "Новинки",...)
selectedCatalog = listOfCatalogs[0];
listOfCategories = getCategories(token, selectedCatalog);
Если listOfCategories = NULL Тогда
    Сообщить("Список категорий для каталога "+selectedCatalog+" не получен. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//Выбор листовых категорий( то есть тех в которых располагаются товары)
leafCategories = selectAllLeafCategories(listOfCategories);
//вывод сообщений с названиями категорий и их уникальными идентификаторами.
//loopCategories(leafCategories);

//3. Получение списка товаров категории
leafCategoryId = leafCategories[0][1];//выбираем id одной из листовых категорий
leafCategoryName = leafCategories[0][0];//название категории
goodsPropertyNames = Новый Массив();//список свойств товаров при получении списка товаров категории
goodsPropertyNames.Добавить("id");//id всегда будет первым свойством, остальные произвольно.
listOfGoods = getGoods(token, selectedCatalog, leafCategoryId, goodsPropertyNames);
Если listOfGoods = NULL Тогда
    Сообщить("Список товаров для категории "+leafCategoryName+" не получен. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//Перебор всех свойств последнего товара, полученного в списке
index = listOfGoods.Количество()-1;//можно выбрать любой товар из списка 0...listOfGoods.Количество()-1
goodsProperties = listOfGoods[index];
//loopAllProperties(goodsProperties,goodsPropertyNames);

//4. Получение описания товара
goodsDescriptionNames = Новый Массив();//список названий характеристик товара при получении описания товара.
goodsId = listOfGoods[0][0];//id у первого товара в списке
listOfProperties =
getDescription(token, selectedCatalog, leafCategoryId, goodsId, goodsDescriptionNames);
Если listOfProperties = NULL Тогда
    Сообщить("Список характеристик для товара "+goodsId+" не получен. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//loopAllProperties(listOfProperties,goodsDescriptionNames);

//5. Получение информации об изображениях товара
imagePropertyNames = Новый Массив();
imageInfo = getImageInfo(token, goodsId, imagePropertyNames);
Если imageInfo = NULL Тогда
    Сообщить("Информация о изображении для товара "+goodsId+" не получена. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;

```

```

//loopAllProperties(imageInfo[0],imagePropertyNames);

//6. Получение информации о ценах и количестве товара по коду товара
goodsPropertyNames2 = Новый Массив();
goods = getGoodsById(token,goodsId,goodsPropertyNames2);
Если goods = NULL Тогда
    Сообщить("Информация о товаре с ид:"+goodsId+" не получена. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//loopAllProperties(goods,goodsPropertyNames);

//7. Получение списка новых товаров.
newGoodsPropertyNames = Новый Массив();
newGoodsPropertyNames.Добавить("id");//id всегда будет первым свойством, остальные произвольно.
listOfNewGoods = getNewGoods(token,newGoodsPropertyNames);
Если listOfNewGoods = NULL Тогда
    Сообщить("Список новых товаров не получен.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//Перебор всех свойств последнего товара, полученного в списке
index = listOfNewGoods.Количество()-1;
goodsProperties = listOfNewGoods[index];
//loopAllProperties(goodsProperties,newGoodsPropertyNames);

//8. Получение списка уцененных товаров
usedGoodsPropertyNames = Новый Массив();
goodsPropertyNames.Добавить("id");//id всегда будет первым свойством, остальные произвольно.
listOfUsedGoods = getUsedGoods(token,usedGoodsPropertyNames);
Если listOfUsedGoods = NULL Тогда
    Сообщить("Список уцененных товаров не получен. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//Перебор всех свойств последнего товара, полученного в списке
index = listOfUsedGoods.Количество()-1;
goodsProperties = listOfUsedGoods[index];
//loopAllProperties(goodsProperties,usedGoodsPropertyNames);

//9. Получение даты накладной по номеру стикера

stickerPropertyNames = Новый Массив();
sticker = "1234";
stickerData = getDateBySticker(token,sticker,stickerPropertyNames);
Если stickerData = NULL Тогда
    Сообщить("Дата стикера не получена. Тест прерван.", СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//loopAllProperties(stickerData[0],stickerPropertyNames);

//10 Получение даты документа по серийному номеру
serialPropertyNames = Новый Массив();
serialNumber = "1234";
serialNumberData = getDateBySerialNumber(token,serialNumber,serialPropertyNames);
Если serialNumberData = NULL Тогда
    Сообщить("Дата по серийному номеру не получена.", СтатусСообщения.Важное);
    //Возврат; - тут не прерывать, так как если не нашли, то его и нет, это не обязательно ошибка.
Иначе
    //loopAllProperties(serialNumberData[0],serialPropertyNames);

```

```

КонецЕсли;

//11 Получение дополнительной информации (курс валюты) о сеансе
infoNames = Новый Массив();
info = getInfo(token,infoNames);
Если info = NULL Тогда
    Сообщить("Информация не получена. Тест прерван.");
СтатусСообщения.Важное);
    Возврат;
КонецЕсли;
//loopAllProperties(info[0],infoNames);

Сообщить("Конец теста каталога.", СтатусСообщения.Важное);
КонецПроцедуры

Процедура testDocuments(token)
    //Документы:
    //Получение типов документов
    docTypesNames = Новый Массив();
    docTypesNames.Добавить("id");//идентификатор, используемый для получения списка
документов
    docTypesNames.Добавить("name");// название типа документов (резерв, счет)
    documentTypes = getDocumentTypes(token,docTypesNames);
    Если documentTypes = NULL Тогда
        Сообщить("Список типов документов не получен. Тест прерван.");
СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    //loopAllProperties(documentTypes[0],docTypesNames);
    //Получение списка организаций
    orgNames = Новый Массив();
    orgNames.Добавить("id");//идентификатор, используемый для создания документов
    orgNames.Добавить("название");// название организации (резерв, счет,...)
    organizationsData = getOrganizations(token,orgNames);
    Если organizationsData = NULL Тогда
        Сообщить("Список организаций не получен. Тест прерван.");
СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    //loopAllProperties(organizationsData[0],orgNames);
    //Получение списка документов по типу

    propertyNames = новый Массив();
    propertyNames.Добавить("id");
    documentTypeld = 0;//0==резервы, допустимые значения надо брать из documentTypes
(см ранее), в данный момент 0..6
    beginDate = "20140101";
    endDate = "20150101";
    documentListData =
getDocumentsList(token,documentTypeld,beginDate,endDate,propertyNames);
    Если documentListData = NULL Тогда
        Сообщить("Список документов не получен. Тест прерван.");
СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    //loopAllProperties(documentListData[0],propertyNames);

    //Создание резерва
    propertyNames1 = новый Массив();//названия свойств,значения которых вернуться
после создания документа
    propertyNames1.Добавить("id");
    goodsPropertyNames = новый Массив();
    documentTypeld = "0";
    organizaionId = organizationsData[0][0];//id первой организации;           date = "20150228";

```

```

comment = "Этот резерв создан в тестовых целях";
goods = "1291752,1;1291753,2;";
documentCreationResult =
createDocument(token,documentTypeld,organizaionId,date,comment,goods,goodsPropertyNames,p
ropertyNames1);
    Если documentCreationResult = NULL Тогда
        Сообщить("Не получилось создать документ с типом "+documentTypeld+. Тест
прерван.", СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    Если documentCreationResult[0].Количество() > 0 Тогда
        goodsInfo = documentCreationResult[0][0]; //первый товар из документа, второй -
[0][1]
        //loopAllProperties(goodsInfo,goodsPropertyNames);
    КонецЕсли;
    documentInfo = documentCreationResult[1];
    //loopAllProperties(documentInfo,propertyNames1);

    //Чтение деталей резерва
    createdDocumentId = documentInfo[0];//id ранее созданного документа
    propertyNames1 = новый Массив();
    goodsPropertyNames = новый Массив();
    documentReadResult =
getDocument(token,createdDocumentId,goodsPropertyNames,propertyNames1);
    Если documentReadResult = NULL Тогда
        Сообщить("Не получилось прочесть детали документа с идентификатором
"+createdDocumentId+. Тест прерван.", СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    Если documentReadResult[0].Количество() > 0 Тогда
        goodsInfo = documentReadResult[0][0]; //первый товар из документа, второй -
[0][1]
        //loopAllProperties(goodsInfo,goodsPropertyNames);
    КонецЕсли;
    documentInfo = documentReadResult[1];
    //loopAllProperties(documentInfo,propertyNames1);

    //Редактирование резерва
    propertyNames1 = новый Массив();//названия свойств,значения которых вернуться
после создания документа
    propertyNames1.Добавить("id");
    goodsPropertyNames = новый Массив();
    documentTypeld = "0";
    organizaionId = organizationsData[0][0];//id первой организации;      date = "20150228";
    comment = "Этот резерв создан в тестовых целях";
    goods = "1291752,1;1291753,3";
    documentUpdateResult =
updateDocument(token,createdDocumentId,documentTypeld,organizaionId,date,comment,goods,go
odsPropertyNames,propertyNames1);
    Если documentUpdateResult = NULL Тогда
        Сообщить("Не получилось отредактировать документ с типом
"+documentTypeld+. Тест прерван.", СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    Если documentUpdateResult[0].Количество() > 0 Тогда
        goodsInfo = documentUpdateResult[0][0]; //первый товар из документа, второй -
[0][1]
        //loopAllProperties(goodsInfo,goodsPropertyNames);
    КонецЕсли;
    documentInfo = documentUpdateResult[1];
    //loopAllProperties(documentInfo,propertyNames1);

    //Удаление резерва
    propertyNames1 = новый Массив();//названия свойств,значения которых вернуться

```

```

после создания документа
propertyNames1.Добавить("id");
goodsPropertyNames = новый Массив();

documentDeleteResult =
deleteDocument(token,createdDocumentId,documentTypeId,goodsPropertyNames,propertyNames1
);
Если documentDeleteResult = NULL Тогда
Сообщить("Не получилось удалить документ с идентификатором "+ createdDocumentId +" типом "+documentTypeId+". Тест прерван.", СтатусСообщения.Важное);
Возврат;
КонецЕсли;

//Создание счета
propertyNames1 = новый Массив(); //названия свойств,значения которых вернуться
после создания документа
propertyNames1.Добавить("id");
goodsPropertyNames = новый Массив();
documentTypeId = "2"; //счета==2, резервы==0,
organizaionId = "101076";
date = "20150228";
comment = "Этот счет создан в тестовых целях";
goods = "1291752,1;1291753,2";
documentCreationResult =
createDocument(token,documentTypeId,organizaionId,date,comment,goods,goodsPropertyNames,propertyNames1);
Если documentCreationResult = NULL Тогда
Сообщить("Не получилось создать документ с типом "+documentTypeId+". Тест прерван.", СтатусСообщения.Важное);
Возврат;
КонецЕсли;
Если documentCreationResult[0].Количество() > 0 Тогда
goodsInfo = documentCreationResult[0][0]; //первый товар из документа, второй -
[0][1]
//loopAllProperties(goodsInfo,goodsPropertyNames);
КонецЕсли;
documentInfo = documentCreationResult[1];
//loopAllProperties(documentInfo,propertyNames1);

//Чтение деталей счета
createdDocumentId = documentInfo[0];//id ранее созданного документа
propertyNames1 = новый Массив();
goodsPropertyNames = новый Массив();
documentReadResult =
getDocument(token,createdDocumentId,goodsPropertyNames,propertyNames1);
Если documentReadResult = NULL Тогда
Сообщить("Не получилось прочесть детали документа с идентификатором "+createdDocumentId+". Тест прерван.", СтатусСообщения.Важное);
Возврат;
КонецЕсли;
Если documentReadResult[0].Количество() > 0 Тогда
goodsInfo = documentReadResult[0][0]; //первый товар из документа, второй -
[0][1]
//loopAllProperties(goodsInfo,goodsPropertyNames);
КонецЕсли;
documentInfo = documentReadResult[1];

//Редактирование счета
propertyNames1 = новый Массив(); //названия свойств,значения которых вернуться
после создания документа
propertyNames1.Добавить("id");
goodsPropertyNames = новый Массив();
documentTypeId = "2";
organizaionId = "101076";

```

```

date = "20150228";
comment = "Этот счет создан(отредактирован) в тестовых целях";
goods = "1291752,1;1291753,3";
documentUpdateResult =
updateDocument(token,createdDocumentId,documentTypeld,organizaionId,date,comment,goods,go
odsPropertyNames,propertyNames1);
    Если documentUpdateResult = NULL Тогда
        Сообщить("Не получилось отредактировать документ с типом
"+documentTypeld+. Тест прерван.", СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;
    Если documentUpdateResult[0].Количество() > 0 Тогда
        goodsInfo = documentUpdateResult[0][0]; //первый товар из документа, второй -
[0][1]
        //loopAllProperties(goodsInfo,goodsPropertyNames);
    КонецЕсли;
    documentInfo = documentUpdateResult[1];
    //loopAllProperties(documentInfo,propertyNames1);
    //Удаление счета
    propertyNames1 = новый Массив();//названия свойств,значения которых вернуться после
создания документа
    propertyNames1.Добавить("id");
    goodsPropertyNames = новый Массив();

    documentDeleteResult =
deleteDocument(token,createdDocumentId,documentTypeld,goodsPropertyNames,propertyNames1
);
    Если documentDeleteResult = NULL Тогда
        Сообщить("Не получилось удалить документ с идентификатором "+
createdDocumentId +" типом "+documentTypeld+. Тест прерван.", СтатусСообщения.Важное);
        Возврат;
    КонецЕсли;

    Сообщить("Конец теста документов.", СтатусСообщения.Важное);
КонецПроцедуры

//--Служебные функции для работы с сервисами-----
Функция initializeAuthenticationService()
    Возврат WSCсылки.AuthenticationService.СоздатьWSПрокси("http://ws.web.netlab.com/",
"AuthenticationWSService", "AuthenticationWSPort");
КонецФункции

Функция initializeCatalogService()

Возврат
WSCсылки.CatalogService.СоздатьWSПрокси("http://ws.web.netlab.com/", "CatalogWSService",
"CatalogWSPort");
КонецФункции

Функция initializeDocumentService()
    Возврат
WSCсылки.DocumentsService.СоздатьWSПрокси("http://ws.web.netlab.com/", "DocumentsWSServ
ice", "DocumentsWSPort");
КонецФункции

Функция isError(response)
    Если response.status.code <> "200" Тогда
        Сообщить("Ошибка при выполнении запроса
код:"+response.status.code+"сообщение:"+response.status.message,
СтатусСообщения.Важное);
        Возврат Истина;
    Иначе
        Возврат Ложь;
    КонецЕсли;

```

КонецФункции

```
Функция parseGoodsResponse(response,names)
    Если isError(response) Тогда
        Возврат NULL;
    КонецЕсли;
    results = Новый Массив();
    addProperties(response.data.properties,results,names);
    Если results.Количество() < 1 Тогда
        Сообщить("Список пуст.", СтатусСообщения.Важное);
        Возврат NULL;
    Иначе
        Сообщить("Получено "+results.Количество()+" пар ключ-значение.",
СтатусСообщения.Важное);
    КонецЕсли;
    Возврат results;
КонецФункции
```

```
Функция parseEntityResponse(response,names)
    Если isError(response) Тогда
        Возврат NULL;
    КонецЕсли;
    results = Новый Массив();
    Для Каждого item Из response.data.list.item Цикл
        imageItem = Новый Массив();
        addProperties(item.properties,imageItem,names); //Копирование всех свойств
товара (цены, кол-во,...)
        results.Добавить(imageItem);
    КонецЦикла;
    Если results.Количество() < 1 Тогда
        Сообщить("Список пуст.", СтатусСообщения.Важное);
        Возврат NULL;
    Иначе
        Сообщить("Получено "+results.Количество()+" пар ключ-значение.",
СтатусСообщения.Важное);
    КонецЕсли;
    Возврат results;
КонецФункции
```

```
Функция parseCategoryResponse(response,names)
    Если isError(response) Тогда
        Возврат NULL;
    КонецЕсли;
    results = Новый Массив();
    Для Каждого element Из response.data.goodsList.goods Цикл
//Сообщить("Категория[Ид:"+element.id, СтатусСообщения.Важное);
    goods = Новый Массив();
    addProperties(element.properties,goods,names); //Копирование всех свойств
товара (цены, кол-во,...)
    results.Добавить(goods);
    КонецЦикла;
    Если results.Количество() < 1 Тогда
        Сообщить("Список товаров пуст.", СтатусСообщения.Важное);
        Возврат NULL;
    Иначе
        Сообщить("Получено "+results.Количество()+" товаров.",
СтатусСообщения.Важное);
    КонецЕсли;
    Возврат results;
КонецФункции
```

```
Процедура addProperties(source,dest,names)
    Для Каждого property Из source.property Цикл
```

```

        Если property.name <> NULL Тогда
            index = getIndex(names,property.name); //вычислить индекс
        свойства в массиве
            dest.Вставить(index,property.value); //добавить значение свойства
        на указанную позицию(позиции значений равны позициям названий свойств в глобальном
        массиве)
            КонецЕсли;
        КонецЦикла;
    КонецПроцедуры

Функция parseDocumentsResponse(response,names)
    Если isError(response) Тогда
        Возврат NULL;
    КонецЕсли;
    results = Новый Массив();
    Для Каждого document Из response.data.documentsList.document Цикл
        documentData = Новый Массив();
        documentData.Добавить(document.id);
        addProperties(document.properties,documentData,names);
        results.Добавить(documentData);
    КонецЦикла;
    Если results.Количество() < 1 Тогда
        Сообщить("Список документов пуст.", СтатусСообщения.Важное);
        Возврат NULL;
    Иначе
        Сообщить("Получено "+results.Количество()+" документов.");
    СтатусСообщения.Важное);
    КонецЕсли;
    Возврат results;
КонецФункции

Функция parseDocumentResponse(response,goodsNames,propertyNames)
    Если isError(response) Тогда
        Возврат NULL;
    КонецЕсли;
    //товары, хранящиеся в документе и их свойства.
    goodsList = Новый Массив();
    Попытка
        Для Каждого goods Из response.data.goodsList.goods Цикл
            goodsData = Новый Массив();
            goodsData.Добавить(goods.id);
            addProperties(goods.properties,goodsData,goodsNames);
            goodsList.Добавить(goodsData);
        КонецЦикла;
    Исключение
    КонецПопытки;
    //свойства заголовка документа, такие как его тип, дата.
    propertyList = Новый Массив();

Попытка
    addProperties(response.data.properties,propertyList,propertyNames);
    Исключение
    КонецПопытки;
    results = новый Массив();
    results.Добавить(goodsList);
    results.Добавить(propertyList);
    Возврат results;
КонецФункции
//--Получение данных от сервера-----
Функция authentication(username,password)
    service = initializeAuthenticationService();
    tokenResponse = service.authenticate(username,password);

Если isError(tokenResponse) Тогда

```

```

        Возврат NULL;
    Иначе
        Сообщить("Токен получен:"+tokenResponse.data.token, СтатусСообщения.Важное);
        Сообщить("Действителен до:"+tokenResponse.data.expired_in,
СтатусСообщения.Важное);
        Возврат tokenResponse.data.token;
    КонецЕсли;
КонецФункции

Функция getCatalogs(token)
    service = initializeCatalogService();
    catalogsResponse = service.catalogsAction(token);
    Если isError(catalogsResponse) Тогда
        Возврат NULL;
    Иначе
        array = Новый Массив();
        size = 0;
        Для Каждого element Из catalogsResponse.data.catalogsList.catalog Цикл
            //Сообщить("Каталог:"+element.name, СтатусСообщения.Важное);
            array.Добавить(size,element.name);
            size = size + 1;
        КонецЦикла;
        Если array.Количество() < 1 Тогда
            Сообщить("Список каталогов пуст.", СтатусСообщения.Важное);
            Возврат NULL;
        Иначе
            Сообщить("Получено "+array.Количество()+" каталога(ов)",
СтатусСообщения.Важное);
            КонецЕсли;
            Возврат array;
        КонецЕсли;
КонецФункции

Функция getCategories(token,catalogName)
    service = initializeCatalogService();
    catalogResponse = service.catalogAction(token,catalogName);
    Если isError(catalogResponse) Тогда
        Возврат NULL;
    Иначе
        results = Новый Массив();
        size = 0;
        Для Каждого element Из catalogResponse.data.categories.category Цикл
            //Сообщить("Категория[Название:"+element.name+", ИД:"+element.id+",",
ИД родителя:"+element.parentId+, Листовая:"+element.leaf+"]", СтатусСообщения.Важное);
            category = Новый Массив();
            category.Добавить(element.name);
            category.Добавить(element.id);
            category.Добавить(element.parentId);
            category.Добавить(element.leaf);
            results.Добавить(category);
        КонецЦикла;
        Если results.Количество() < 1 Тогда
            Сообщить("Список категорий пуст.", СтатусСообщения.Важное);
            Возврат NULL;
        Иначе
            Сообщить("Получено "+results.Количество()+" категорий.",
СтатусСообщения.Важное);
            КонецЕсли;
            Возврат results;
        КонецЕсли;
КонецФункции

Функция getGoods(token,catalogName,categoryId,goodsPropertyNames)
    service = initializeCatalogService();

```

```

categoryResponse = service.categoryAction(token,catalogName,categoryId);
Возврат parseCategoryResponse(categoryResponse,goodsPropertyNames);
КонецФункции

Функция getDescription(token,catalogName,categoryId,goodsId,goodsDescriptionNames)
    service = initializeCatalogService();
    goodsResponse = service.goodsAction(token,catalogName,categoryId,goodsId);
    Возврат parseGoodsResponse(goodsResponse,goodsDescriptionNames);
КонецФункции

Функция getImageInfo(token,goodsId,imagePropertyNames)
    service = initializeCatalogService();
    entityListResponse = service.goodsImagesAction(token,goodsId);
    Возврат parseEntityResponse(entityListResponse,imagePropertyNames);
КонецФункции

Функция getGoodsByld(token,goodsld,goodsPropertyNames)
    service = initializeCatalogService();
    goodsResponse = service.goodsByUidAction(token,goodsld);
    Возврат parseGoodsResponse(goodsResponse,goodsPropertyNames);
КонецФункции

Функция getInfo(token,infoNames)
    service = initializeCatalogService();
    entityListResponse = service.getInfoAction(token);
    Возврат parseEntityResponse(entityListResponse,infoNames);
КонецФункции

Функция getDateBySticker(token, sticker, stickerPropertyNames)
    service = initializeCatalogService();
    entityListResponse = service.getDateByStickerAction(token,sticker);
    Возврат parseEntityResponse(entityListResponse,stickerPropertyNames);
КонецФункции

Функция getDateBySerialNumber(token, serialNumber, serialNumberNames)
    service = initializeCatalogService();
    entityListResponse = service.getDateBySerialNumberAction(token,serialNumber);
    Возврат parseEntityResponse(entityListResponse,serialNumberNames);
КонецФункции

Функция getNewGoods(token,goodsPropertyNames)
    service = initializeCatalogService();
    categoryResponse = service.newGoodsAction(token);
    Возврат parseCategoryResponse(categoryResponse,goodsPropertyNames);
КонецФункции

Функция getUsedGoods(token,goodsPropertyNames)
    service = initializeCatalogService();
    categoryResponse = service.usedGoodsAction(token);
    Возврат parseCategoryResponse(categoryResponse,goodsPropertyNames);
КонецФункции

Функция getDocumentTypes(token,propertyNames)
    service = initializeDocumentService();
    entityListResponse = service.getDocumentTypesAction(token);
    Возврат parseEntityResponse(entityListResponse,propertyNames);
КонецФункции

Функция getOrganizations(token,propertyNames)
    service = initializeDocumentService();
    entityListResponse = service.getOrganizationsAction(token);
    Возврат parseEntityResponse(entityListResponse,propertyNames);
КонецФункции

```

```

Функция getDocumentsList(token,documentTypeld,beginDate,endDate,propertyNames)
    service = initializeDocumentService();
    documentsResponse = service.listDocuments(token,documentTypeld,beginDate,endDate);
    Возврат parseDocumentsResponse(documentsResponse,propertyNames);
КонецФункции

Функция
createDocument(token,documentTypeld,organizaionId,date,comment,goods,goodsNames,propertyN
ames)
    service = initializeDocumentService();
    documentResponse =
service.createAction(token,documentTypeld,organizaionId,date,comment,goods);
    Возврат parseDocumentResponse(documentResponse,goodsNames,propertyNames);
КонецФункции

Функция getDocument(token,documentId,goodsNames,propertyNames)
    service = initializeDocumentService();
    documentResponse = service.readAction(token,documentId);
    Возврат parseDocumentResponse(documentResponse,goodsNames,propertyNames);
КонецФункции

Функция updateDocument(token,documentId,documentTypeld,organizaionId,date,comment,goods,goodsNam
es,propertyNames)
    service = initializeDocumentService();
    documentResponse =
service.updateAction(token,documentId,documentTypeld,organizaionId,date,comment,goods);
    Возврат parseDocumentResponse(documentResponse,goodsNames,propertyNames);
КонецФункции

Функция deleteDocument(token,documentId,documentTypeld,goodsNames,propertyNames)
    service = initializeDocumentService();

documentResponse = service.deleteAction(token,documentId,documentTypeld);
    Возврат parseDocumentResponse(documentResponse,goodsNames,propertyNames);
КонецФункции

//--Логические операции с данными-----
Функция selectAllLeafCategories(listOfCategories)
    results = Новый Массив();
    Для index = 0 по listOfCategories.Количество()-1 Цикл
        Если listOfCategories[index][3] = Истина Тогда
            results.Добавить(listOfCategories[index]);
        КонецЕсли;
    КонецЦикла;
    Сообщить("Выделено "+results.Количество()+" листовых категорий.");
    СтатусСообщения.Важное);
    Возврат results;
КонецФункции

Функция getIndex(array,value)
    //вычисляет индекс в массиве, если значения нет, то добавляет.
    Если array.Количество() > 0 Тогда
        Для index = 0 По array.Количество()-1 Цикл
            Если array[index] = value Тогда
                Возврат index;
            КонецЕсли;
        КонецЦикла;
    КонецЕсли;
    array.Добавить(value);
    //Сообщить("propertyNames["++(array.Количество()-1)+"]="+value,
    СтатусСообщения.Важное);
    Возврат (array.Количество()-1);
КонецФункции

```

```

Процедура loopAllProperties(values,names)
    //Пробегает по всем свойствам товара и ставит им в соответствие название из списка
    названий свойств
        Для index = 0 По (values.Количество()-1) Цикл
            Если index <= (names.Количество()-1) Тогда
                Сообщить(names[index]+ "=" +values[index], СтатусСообщения.Важное);
            КонецЕсли;
        КонецЦикла;
    КонецПроцедуры

Процедура loopByArray(array)
    Для index = 0 По (array.Количество()-1) Цикл
        Сообщить(array[index], СтатусСообщения.Важное);
    КонецЦикла;
    КонецПроцедуры

Процедура loopCategories(array)
    Для index = 0 По (array.Количество()-1) Цикл
        Сообщить(array[index][0]+",id:" +array[index][1], СтатусСообщения.Важное);
    КонецЦикла;
    КонецПроцедуры

```

Образец кода 1с для управляемых форм для работы с изображениями

Дополнительный пример реализации получения (обработки) изображений товаров, который использует один из наших клиентов.

образец кода (для управляемых форм)

```

ТаблицаСвойствИзображения = Обработки.СетеваяЛаборатория.getImageInfo(Параметры.Токен,
Параметры.НоменклатураКод);
Адрес = ТаблицаСвойствИзображения.Найти("Url", "Свойство");
Если НЕ Адрес = Неопределено Тогда
    АдресСтруктура =
        Обработки.СетеваяЛаборатория.ПолучитьСтруктуруАдресаКартинки(Адрес.Значение);
    Если Не АдресСтруктура = Неопределено Тогда
        Соединение = Новый HTTPСоединение(АдресСтруктура.Сервер);
        Заголовки = Новый Соответствие;
        Заголовки.Вставить("host", АдресСтруктура.Сервер);
        Запрос = Новый HTTPЗапрос(АдресСтруктура.Путь, Заголовки);
        Ответ = Соединение.Получить(Запрос);
    Если Ответ.КодСостояния = 200 Тогда // Данные получены, обрабатываем их
        Картинка = Новый Картинка(Ответ.ПолучитьТелоКакДвоичныеДанные());
        Изображение = ПоместитьВоВременноеХранилище(Картинка, УникальныйИдентификатор);
    КонецЕсли;
    КонецЕсли;
    КонецЕсли;

```

Функция ПолучитьСтруктуруАдресаКартинки(Адрес) Экспорт

```

АдресСтруктура = Новый Структура;
АдресСтруктура.Вставить("Сервер", "netlab.ru");
Код = Найти(Адрес, "/ISAPI");

```

Если Код > о Тогда
АдресСтруктура.Вставить("Путь", Сред(Адрес, Код));
Возврат АдресСтруктура;
КонецЕсли;
Возврат Неопределено;

КонецФункции

4. ПРИМЕРЫ КОДА НА PHP И JAVASCRIPT

Для просмотра данных ответа сервера написана простая html страница с полями для ввода данных и кнопками, соответствующими вызываемому методу. Создав html страницу, php и js файлы, вы сможете посмотреть структуру ответов сервера и протестировать необходимый функционал веб-сервисов.

Пояснение к коду.

Для загрузки компонентов на html странице используется javascript код. По клику на кнопку вызывается onclick() функция с обращением к php. В результате выполнения php кода возвращается ответ сервера в переменной \$responseStr. Данные этой переменной показываются в конце html страницы.

Готовые файлы можно скачать по ссылке <http://www.netlab.ru/nldealer/docs/phpSample.zip>

Архив содержит html страницу, javascript и php скрипты.

4.1 HTML код

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Example Services</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script src="resources/js/lib/jquery-1.6.2.min.js" type="text/javascript"></script>
    <script src="resources/js/scriptsServices.js" ></script>
</head>

<style class="tableClass" type="text/css">
    table {border: 2px solid gray; margin: 5px 0 5px 0}
    tr, td {border: 0; }
</style>

<body>
<a id="topHrefId" ></a>
<div id="buttonsExample">
    <table class="tableClass">
        <tr>
            <td><button id="getTokenBtnId" title="Аутентификация"
onclick="authenticate(this)">authenticate</button>(username, password)</td>
        </tr>
        <tr>
            <td><input id="usernameId" style="background-color: white"
placeholder="clientName"><label for="usernameId">username</label></td>
        </tr>
        <tr>
            <td><input id="passwordId" style="background-color: white"
placeholder="password"><label for="passwordId">password</label></td>
        </tr>
        <tr>
            <td>Token:<span id="tokenOutputId"></span></td>
        </tr>
    </table>
<table class="tableClass">
    <tr>
        <td><button id="getCatalogsActionBtnId" title="Список каталогов товаров"
onclick="catalogsAction(this)">catalogsAction</button>(token)</td>
    </tr>
```

```

<tr>
    <td><button id="getCatalogActionBtnId" title="Список категорий товаров
каталога" onclick="catalogAction(this)">catalogAction</button>(token)</td>
</tr>
</table>

<table class="tableClass">
    <tr>
        <td><input id="catalogNameId" value="Прайс-лист" style="background-color:
white"><label for="catalogNameId">catalogName</label></td>
    </tr>
    <tr>
        <td><input id="categoryId" style="background-color: white"
placeholder="123456"><label for="categoryId">categoryId</label></td>
    </tr>
    <tr>
        <td><input id="goodsId" style="background-color: white"
placeholder="1234567"><label for="goodsId">goodsId</label></td>
    </tr>
    <tr>
        <td><input id="showDeletedId" style="background-color: white"
placeholder="1 или 0"><label for="showDeletedId">showDeleted</label></td>
    </tr>
    <tr>
        <td><input id="versionId" style="background-color: white"
placeholder="1"><label for="versionId">versionId</label></td>
    </tr>
    <tr>
        <td><input id="partNumberId" style="background-color: white"
placeholder="STDT8000200"><label for="partNumberId">partNumber</label></td>
    </tr>
    <tr>
        <td><input id="pageNumberId" style="background-color: white"
placeholder="page Number"><label for="pageNumberId">pageNumber</label></td>
    </tr>
    <tr>
        <td><button id="getCategoryActionBtnId" title="Список товаров категории"
onclick="categoryAction(this)">categoryAction</button>(catalogName, categoryId)</td>
    </tr>
    <tr>
        <td><button id="getCategoryWithDelActionBtnId"
title="Список товаров категории + удаленные товары"
onclick="categoryWithDelAction(this)">categoryWithDelAction</button>(catalogName, categoryId, showDeleted)
    </td>
    </tr>
    <tr>
        <td><button id="getAllGoodsActionBtnId" title="Получение всех товаров (+
удаленных)" onclick="allGoodsAction(this)">getAllGoodsAction</button>(versionId, showDeleted) *Длительный
метод*</td>
    </tr>
    <tr>
        <td><button id="getGoodsActionBtnId" title="Описание товара по id"
onclick="goodsAction(this)">goodsAction</button>(catalogName, catalogId, goodsId,
token)</td>
    </tr>
    <tr>
        <td><button id="getGoodsByUidActionBtnId" title="Информация о ценах и
количестве товара по коду товара"
onclick="goodsByUidAction(this)">goodsByUidAction</button>(goodsId, token)</td>
    </tr>
    <tr>
        <td><button id="getNewGoodsActionBtnId" title="Список новых товаров"
onclick="newGoodsAction(this)">newGoodsAction</button>(token)</td>
    </tr>

```

```

        </tr>
        <tr>
            <td><button id="getUsedGoodsActionBtnId" title="Список учененных товаров"
onclick="usedGoodsAction(this)">usedGoodsAction</button>(token)</td>
        </tr>
        <tr>
            <td> <button id="getGoodsByPartNumberActionBtnId" title="Получить
информацию о товаре по PartNumber"
onclick="goodsByPartNumber(this)">getGoodsByPartNumberAction</button>(token,
partNumber)</td>
            </tr>
            <tr>
                <td> <button id="goodsDescriptionByCategoryActionBtnId" title="Получение
описания товара по категории"
onclick="getDescriptionByCategory(this)">goodsDescriptionByCategoryAction
                </button>
                (categoryId, pageNumber, token)
            </td>
        </tr>
        <tr>
            <td>
                <button id="goodsImagesByCategoryActionBtnId" title="Получение
изображений товара по категории"
                    onclick="getImagesByCategory(this)">goodsImagesByCategoryAction
                </button>
                (categoryId, token)
            </td>
        </tr>
    </table>
    <table class="tableClass">
        <tr>
            <td><button id="getDateByStickerActionBtnId" title="Поиск даты накладной по
номеру стикера"
onclick="dateByStickerAction(this)">getDateByStickerAction</button>(token,
stickerNumber)</td>
            </tr>
            <tr>
                <td><input id="stickerNumberId" style="background-color: white"><label
for="stickerNumberId">stickerNumber</label></td>
            </tr>
    </table>
    <table class="tableClass">
        <tr>
            <td><button id="getDateBySerialNumberActionBtnId" title="Поиск даты
накладной по номеру SN товара"
onclick="dateBySerialNumberAction(this)">getDateBySerialNumberAction</button>(token) (se
rialNumber)</td>
            </tr>
            <tr>
                <td><input id="serialNumberId" style="background-color: white"
placeholder="E9M0AB059476"><label for="serialNumberId">serialNumber</label></td>
            </tr>
    </table>
    <table class="tableClass">
        <tr>
            <td><button id="getInfoActionBtnId" title="Курсы валют"
onclick="infoAction(this)">getInfoAction</button>(token)</td>
            </tr>
    </table>
    <table class="tableClass">
        <tr>
            <td> <button id="getOrganizationsActionBtnId" title="Список юридических

```

```

лиц" onclick="organizationAction(this)">getOrganizationsAction</button>(token)</td>
    </tr>
</table>
<table class="tableClass">
    <tr>
        <td> <button id="getDocumentTypesActionBtnId" title="Список доступных типов
документов"
onclick="documentTypesAction(this)">getDocumentTypesAction</button>(token)</td>
    </tr>
</table>
<table class="tableClass">
    <tr>
        <td> <button id="listDocumentsBtnId" title="Список документов"
onclick="listDocuments(this)">listDocuments</button>(token, typeId, beginDate,
endDate)</td>
    </tr>
    <tr>
        <td><input id="typeId" style="background-color: white"
placeholder="0"><label for="typeId">typeId</label></td>
    </tr>
    <tr>
        <td><input id="beginDatePickerId" style="background-color: white"
placeholder="YYYYMMDD" ><label for="beginDatePickerId">beginDate</label></td>
    </tr>
    <tr>
        <td><input id="endDatePickerId" style="background-color: white"
placeholder="YYYYMMDD" ><label for="endDatePickerId">endDateId</label></td>
    </tr>
</table>
<table class="tableClass">
    <tr>
        <td> <button id="readActionBtnId" title="Список товаров документа"
onclick="readAction(this)">readAction</button>(token, documentId)</td>
    </tr>
    <tr>
        <td><input id="documentId" style="background-color: white"
placeholder="123456789Ц"><label for="documentId">documentId</label></td>
    </tr>
</table>
<table class="tableClass">
    <tr>
        <td> <button id="createActionBtnId" title="Создание документов"
onclick="createAction(this)">createAction</button>(token, typeId, organizationId, date,
comment, goodsString)</td>
    </tr>
    <tr>
        <td><button id="updateActionBtnId" title="Редактирование документа"
onclick="updateAction(this)">updateAction</button>(token, typeId, documentId,
organizationId, date, comment, goodsString)</td>
    </tr>
    <tr>
        <td><button id="changeActionBtnId" title="Редактирование документа"
onclick="changeAction(this)">changeAction</button>(token, typeId, documentId,
goodsString)</td>
    </tr>
    <tr>
        <td><button id="deleteActionBtnId" title="Удаление документов"
onclick="deleteAction(this)">deleteAction</button>(token, typeId, documentId)</td>
    </tr>
    <tr>
        <td><button id="createReserveBtnId" title="Создание нового резерва"
onclick="createReserveAction(this)">createReserve</button>(token, organizationId, date,
comment, goodsString, reservePrice)</td>
    </tr>
    <tr>

```

```

        <td> <button id="repairGoodsListBtnId" title="Запрос списка товаров в
ремонте" onclick="repairGoodsList(this)">repairGoodsList</button>(token)</td>
</tr>
<tr>
    <td> <button id="readShipmentBtnId" title="Данные документа отгрузки"
onclick="readShipment(this)">readShipment</button>(token, documentId)</td>
</tr>
<tr>
    <td> <button id="splitReserveActionBtnId" title="Выполнить разбиение
резерва">
        onclick="splitReserve(this)">splitReserve</button>(token,
documentId, goodsId)</td>
    </tr>
    <tr>
        <td> <button id="mergeReservesActionBtnId" title="Выполнить слияние
резервов">
            onclick="mergeReserves(this)">mergeReserves</button>(token,
document1Id, document2Id)</td>
        </tr>
        <tr>
            <td> <button id="prolongReserveActionBtnId" title="Выполнить продление
резерва">
                onclick="prolongReserve(this)">prolongReserve</button>(token,
documentId, date)</td>
            </tr>
            <tr>
                <td>
                    <button id="orderGoodsActionBtnId" title="Заказ остатковующих товаров"
                        onclick="orderGoods(this)">orderGoods
                    </button>
                    (token, documentId, goodsUid)
                </td>
            </tr>
            <tr>
                <td><input id="typeId2" style="background-color: white"
placeholder="0"><label for="typeId2">typeId</label></td>
            </tr>
            <tr>
                <td><input id="commentId" style="background-color: white"
placeholder="комментарий"><label for="commentId">comment</label></td>
            </tr>
            <tr>
                <td><input id="goodsStringId" style="background-color: white"
placeholder="1234567,1 (артикул, кол-во);"><label for="goodsStringId">goodsString.
Пример:1234567,1;5555566,3;7777777,50;</label></td>
            </tr>
            <tr>
                <td><input id="reservePriceId" style="background-color: white"
placeholder="0"><label for="reservePriceId">reservePrice</label></td>
            </tr>
            <tr>
                <td><input id="organizationId" style="background-color: white"
placeholder="123456"><label for="organizationId">organizationId</label></td>
            </tr>
            <tr>
                <td><input id="documentId2" style="background-color: white"
placeholder="123456789Ц"><label for="documentId2">documentId</label></td>
            </tr>
            <tr>
                <td><input id="document1Id" style="background-color: white"
placeholder="123456789Ц"><label for="document1Id">document1Id</label></td>
            </tr>
            <tr>
                <td><input id="document2Id" style="background-color: white"
placeholder="123456789Ц"><label for="document2Id">document2Id</label></td>
            </tr>

```

```

</tr>
<tr>
    <td><input id="goodsSplitId" style="background-color: white"
placeholder="1194608;1290640;"><label for="goodsSplitId">goodsId</label></td>
</tr>
<tr>
    <td><input id="dateId" style="background-color: white"
placeholder="YYYYMMDD"><label for="dateId">date</label></td>
</tr>
<tr>
    <td><input id="goodsUid" style="background-color: white"
placeholder="Артикул"><label for="goodsUid">goodsUid</label></td>
</tr>
</table>





```

```

<tr>
    <td><input id="endIntervalId" style="background-color: white"
placeholder="YYYYMMDD"><label for="endIntervalId">endInterval</label></td>
</tr>
<tr>
    <td><input id="packId" style="background-color: white"
placeholder="986520"><label for="packId">packId</label></td>
</tr>
    <td><input id="documentDeliveryId" style="background-color: white"
placeholder="123456789Ц;123456789Ц;"><label
for="documentDeliveryId">documentIds</label></td>
</tr>
    <td><input id="deliveryDateId" style="background-color: white"
placeholder="YYYYMMDD"><label for="deliveryDateId">deliveryDate</label></td>
</tr>
    <td><input id="deliveryTypeId" style="background-color: white"
placeholder="Доставка"><label for="deliveryTypeId">deliveryType</label></td>
</tr>
</table>






```

4.2 JavaScript код

Каждая функция соответствует одному методу веб-сервиса.

`scriptsServices.js`

```

var token;
var url = 'service.php';
function fillPageWithData(data) {
    var responseInfo = document.getElementById('responseId');
    responseInfo.innerHTML = data;
}

function authenticate(param) {
    var buttonId = param.id;
    var username = document.getElementById('usernameId').value;
    var password = document.getElementById('passwordId').value;
    $.get(url + "?btnId=" + buttonId + "&username=" + username + "&password=" + password,
function(data) {
    var responseInfo = document.getElementById('tokenOutputId');
    responseInfo.innerHTML = data;
    token = data;
});
}

function catalogsAction(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function catalogAction(param) {
    var buttonId = param.id;
    var categoryId = document.getElementById('catalogNameId').value;
    $.get(url + "?btnId=" + buttonId + "&categoryId=" + categoryId
        + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function categoryAction(param) {
    var buttonId = param.id;
    var catalogName = document.getElementById('catalogNameId').value;
    var categoryId = document.getElementById('categoryId').value;
    $.get(url + "?btnId=" + buttonId + "&catalogName=" + catalogName +
"&categoryId=" + categoryId
        + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function categoryWithDelAction(param) {
    var buttonId = param.id;
    var catalogName = document.getElementById('catalogNameId').value;
    var categoryId = document.getElementById('categoryId').value;
    var showDeleted = document.getElementById('showDeletedId').value;
    $.get(url + "?btnId=" + buttonId + "&catalogName=" + catalogName +
"&categoryId=" + categoryId
        + "&showDeleted=" + showDeleted + "&oauth_token=" + token , function(data) {
        fillPageWithData(data);
    });
}

function allGoodsAction(param) {
    var buttonId = param.id;
    var versionId = document.getElementById('versionId').value;
    var showDeleted = document.getElementById('showDeletedId').value;
    $.get(url + "?btnId=" + buttonId + "&versionId=" + versionId +

```

```

"&showDeleted=" + showDeleted + "&oauth_token=" + token, function(data) {
    fillWithData(data);
});

function goodsAction(param) {
    var buttonId = param.id;
    var catalogName = document.getElementById('catalogNameId').value;
    var categoryId = document.getElementById('categoryId').value;
    var goodsId = document.getElementById('goodsId').value;
    $.get(url + "?btnId=" + buttonId + "&catalogName=" + catalogName +
"&categoryId=" + categoryId +
    "&goodsId=" + goodsId + "&oauth_token=" + token, function(data) {
        fillWithData(data);
    });
}

function goodsByUidAction(param) {
    var buttonId = param.id;
    var goodsId = document.getElementById('goodsId').value;
    $.get(url + "?btnId=" + buttonId + "&goodsId=" + goodsId +
    "&oauth_token=" + token, function(data) {
        fillWithData(data);
    });
}

function newGoodsAction(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillWithData(data);
    });
}

function usedGoodsAction(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillWithData(data);
    });
}

function goodsByPartNumber(param) {
    var buttonId = param.id;
    var partNumber = document.getElementById('partNumberId').value;
    $.get(url + "?btnId=" + buttonId + "&partNumber=" + partNumber + "&oauth_token=" +
token, function(data) {
        fillWithData(data);
    });
}

function dateByStickerAction(param) {
    var buttonId = param.id;
    var stickerNumber = document.getElementById('stickerNumberId').value;
    $.get(url + "?btnId=" + buttonId + "&stickerNumber=" + stickerNumber +
"&oauth_token=" + token, function(data) {
        fillWithData(data);
    });
}

function dateBySerialNumberAction(param) {
    var buttonId = param.id;
    var serialNumber = document.getElementById('serialNumberId').value;
    $.get(url + "?btnId=" + buttonId + "&serialNumber=" + serialNumber + "&oauth_token=" +
token, function(data) {
        fillWithData(data);
    });
}

function infoAction(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {

```

```

        fillPageWithData(data);
    });
}

function organizationAction(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function documentTypesAction(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function listDocuments(param) {
    var buttonId = param.id;
    var documentTypeId = document.getElementById("typeId").value;
    var beginDate = document.getElementById("beginDateId").value;
    var endDate = document.getElementById("endDateId").value;
    $.get(url + "?btnId=" + buttonId + "&documentTypeId=" + documentTypeId + "&beginDate=" +
beginDate + "&endDate=" + endDate
        + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function readAction(param) {
    var buttonId = param.id;
    var documentId = document.getElementById("documentId").value;
    $.get(url + "?btnId=" + buttonId + "&documentId=" + documentId + "&oauth_token=" +
token, function(data) {
        fillPageWithData(data);
    });
}

function createAction(param) {
    var buttonId = param.id;
    var documentTypeId = document.getElementById("typeId2").value;
    var organizationId = document.getElementById("organizationId").value;
    var date = document.getElementById("dateId").value;
    var comment = document.getElementById("commentId").value;
    var goodsString = document.getElementById("goodsStringId").value;
    $.get(url + "?btnId=" + buttonId + "&documentTypeId=" + documentTypeId +
"&organizationId=" + organizationId
        + "&date=" + date + "&comment=" + comment
        + "&goodsString=" + goodsString + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function updateAction(param) {
    var buttonId = param.id;
    var documentTypeId = document.getElementById("typeId2").value;
    var documentId = document.getElementById("documentId2").value;
    var organizationId = document.getElementById("organizationId").value;
    var date = document.getElementById("dateId").value;
    var comment = document.getElementById("commentId").value;
    var goodsString = document.getElementById("goodsStringId").value;
    $.get(url + "?btnId=" + buttonId + "&documentTypeId=" + documentTypeId + "&documentId=" +
documentId
        + "&organizationId=" + organizationId
        + "&date=" + date + "&comment=" + comment
        + "&goodsString=" + goodsString + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function changeAction(param) {

```

```

var buttonId = param.id;
var documentTypeId = document.getElementById("typeId2").value;
var documentId = document.getElementById("documentId2").value;
var goodsString = document.getElementById("goodsStringId").value;
$.get(url + "?btnId=" + buttonId + "&documentTypeId=" + documentTypeId + "&documentId=" +
+ documentId
    + "&goodsString=" + goodsString + "&oauth_token=" + token, function(data) {
    fillPageWithData(data);
});
}
function deleteAction(param) {
    var buttonId = param.id;
    var documentTypeId = document.getElementById("typeId2").value;
    var documentId = document.getElementById("documentId2").value;
    $.get(url + "?btnId=" + buttonId + "&documentTypeId=" + documentTypeId + "&documentId=" +
+ documentId
    + "&oauth_token=" + token, function(data) {
    fillPageWithData(data);
});
}
function createReserveAction(param) {
    var buttonId = param.id;
    var organizationId = document.getElementById("organizationId").value;
    var dateId = document.getElementById("dateId").value;
    var comment = document.getElementById("commentId").value;
    var goodsString = document.getElementById("goodsStringId").value;
    var reservePrice = document.getElementById("reservePriceId").value;
    $.get(url + "?btnId=" + buttonId + "&documentId=" + documentId
        + "&organizationId=" + organizationId
        + "&date=" + dateId + "&comment=" + comment + "&goodsString=" + goodsString
        + "&reservePrice=" + reservePrice + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
});
}
function repairGoodsList(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
});
}
function readShipment(param) {
    var buttonId = param.id;
    var organizationId = document.getElementById("organizationId").value;
    var documentId = document.getElementById("documentId2").value;
    $.get(url + "?btnId=" + buttonId + "&organizationId=" + organizationId +
"&documentId=" + documentId + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
});
}
function splitReserve(param) {
    var buttonId = param.id;
    var documentId = document.getElementById('documentId2').value;
    var goodsString = document.getElementById('goodsSplitId').value;
    $.get(url + "?btnId=" + buttonId + "&documentId=" + documentId
        + "&goodsString=" + goodsString + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
});
}
function mergeReserves(param) {
    var buttonId = param.id;
    var document1Id = document.getElementById('document1Id').value;
    var document2Id = document.getElementById('document2Id').value;
    $.get(url + "?btnId=" + buttonId + "&document1Id=" + document1Id
        + "&document2Id=" + document2Id + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
});
}

```

```

    });
}

function getListDeliveryDocuments(param) {
    var buttonId = param.id;
    var beginInterval = document.getElementById('beginIntervalId').value;
    var endInterval = document.getElementById('endIntervalId').value;
    $.get(url + "?btnId=" + buttonId + "&beginDate=" + beginInterval
        + "&endDate=" + endInterval + "&oauth_token=" + token, function(data) {
        + fillPageWithData(data);
    });
}

function getAvailableDeliveryDocuments(param) {
    var buttonId = param.id;
    var beginInterval = document.getElementById('beginIntervalId').value;
    var endInterval = document.getElementById('endIntervalId').value;
    $.get(url + "?btnId=" + buttonId + "&beginDate=" + beginInterval
        + "&endDate=" + endInterval + "&oauth_token=" + token, function(data) {
        + fillPageWithData(data);
    });
}

function getDeliveryDetails(param) {
    var buttonId = param.id;
    var packId = document.getElementById('packId').value;
    $.get(url + "?btnId=" + buttonId + "&packId=" + packId
        + "&oauth_token=" + token, function(data) {
        + fillPageWithData(data);
    });
}

function getDeliveryDates(param) {
    var buttonId = param.id;
    var beginInterval = document.getElementById('beginIntervalId').value;
    var endInterval = document.getElementById('endIntervalId').value;
    $.get(url + "?btnId=" + buttonId + "&beginDate=" + beginInterval
        + "&endDate=" + endInterval + "&oauth_token=" + token, function(data) {
        + fillPageWithData(data);
    });
}

function getDeliveryTypes(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function getDeliveryPaymentTypes(param) {
    var buttonId = param.id;
    $.get(url + "?btnId=" + buttonId + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function getDeliveryRates(param) {
    var buttonId = param.id;
    var documentIds = document.getElementById('documentDeliveryId').value;
    var deliveryDate = document.getElementById('deliveryDateId').value;
    var deliveryType = document.getElementById('deliveryTypeId').value;
    $.get(url + "?btnId=" + buttonId + "&documentIds=" + documentIds
        + "&deliveryDate=" + deliveryDate + "&deliveryType=" + deliveryType +
    "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}

function goodsImagesAction(param) {
    var buttonId = param.id;
    var goodsId = document.getElementById("goodsImageId").value;
    var clientImageId = document.getElementById("clientImageId").value;
    var pictureUrl = "<img class='image' "

```

```

src='http://www.netlab.ru/ISAPI/TestISAPI18.dll?' + clientImageId+"&" + goodsId +
"'/>";
    var responseInfo = document.getElementById('imgResponseId');
    responseInfo.innerHTML = pictureUrl;
}
function prolongReserve(param) {
    var buttonId = param.id;
    var documentId = document.getElementById('documentId2').value;
    var date = document.getElementById("dateId").value;
    $.get(url + "?btnId=" + buttonId + "&documentId=" + documentId
        + "&date=" + date + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}
function orderGoods(param) {
    var buttonId = param.id;
    var documentId = document.getElementById('documentId2').value;
    var goodsUid = document.getElementById('goodsUid').value;
    $.get(url + "?btnId=" + buttonId + "&documentId=" + documentId
        + "&goodsUid=" + goodsUid + "&oauth_token=" + token, function (data) {
        fillPageWithData(data);
    });
}
function getDescriptionByCategory(param) {
    var buttonId = param.id;
    var categoryId = document.getElementById('categoryId').value;
    var pageNumber = document.getElementById('pageNumberId').value;
    $.get(url + "?btnId=" + buttonId + "&categoryId=" + categoryId +
"&pageNumber=" + pageNumber
        + "&oauth_token=" + token, function(data) {
        fillPageWithData(data);
    });
}
function getImagesByCategory(param) {
    var buttonId = param.id;
    var categoryId = document.getElementById('categoryId').value;
    $.get(url + "?btnId=" + buttonId + "&categoryId=" + categoryId + "&oauth_token=" +
token, function(data) {
        fillPageWithData(data);
    });
}

```

4.3 PHP код

Пояснение к коду.

Доступ к веб сервисам осуществляется по ссылкам вида

<http://services.netlab.ru/rest/authentication/token.xml?username=<username>&password=<password>>

Вызов php функции возвращает переменную **\$responseStr**. Данные записываются в переменную вызовом вроде

```
$responseStr = $service->readAction($token, $documentId);
```

В функцию service передаются параметры с html формы. Далее используется класс UrlBuilder для построения ссылки, указанной выше. Возвращаемые данные с сервера записываются в переменную **\$responseStr**.

service.php

*Если у вас возникли проблемы с кодировкой русских символов, то раскомментируйте эти строчки в начале php файла:

```
//mb_internal_encoding("UTF-8");
//mb_http_output("UTF-8");
//ob_start("mb_output_handler");

<?php

//mb_internal_encoding ("UTF-8");
//mb_http_output ("UTF-8");
//ob_start ("mb_output_handler");

//constants
class Config {

// USERNAME AND PASSWORD
    public $USERNAME = "";
    public $PASSWORD = "";
// USERNAME AND PASSWORD

    public $ROOT_URL = "http://localhost:9081/rest/";
    // public $ROOT_URL = "http://services.netlab.ru/rest/";
    public $AUTHENTICATION_URL = "authentication/token";
    public $CATALOGS_URL = "catalogs/list";
    public $CATEGORIES_URL = "catalogs/";
    public $DOCUMENTS_URL = "documents/";
    public $DELIVERY_URL = "delivery/";
    public $RESPONSE_FORMAT = ".json?";
    public $REQUEST_DELAY_PERIOD = 20;
    public $GOODS_BY_UID_URL = "goodsByUid";
    public $NEW_GOODS_URL = "newGoods";
    public $USED_GOODS_URL = "usedGoods";
    public $DATE_BY_STICKER_URL = "dateBySticker";
    public $MERGE_RESERVES_URL = "mergeReserves";
    public $SPLIT_RESERVE_URL = "splitReserve";
    public $GOODS_BY_PARTNUMBER_URL = "goodsByPartnumber";
    public $DATE_SERIAL_NUMBER_URL = "dateBySerialNumber";
    public $INFO_URL = "info";
    public $IMAGE_URL = "http://www.netlab.ru/ISAPI/TestISAPI.dll?";
    public $VERSION_URL = "catalogs/versions/2/";
    public $ALLGOODS_URL = "catalogs/allGoods/versions";

    public $ORGANIZATIONS_URL = "organizations";
    public $DOCUMENTS_TYPES_URL = "documentTypes";
    public $LIST_URL = "list";
    public $DOCS_URL = "docs";
    public $DETAILS_URL = "details";
    public $DATES_URL = "dates";
    public $GET_URL = "get";
    public $NEW_URL = "new";
    public $UPDATE_URL = "update";
    public $CHANGE_URL = "change";
    public $DELETE_URL = "delete";
    public $NEW_RESERVE_URL = "newReserve";
    public $REPAIR_GOODS_LIST_URL = "repairGoodsList";
    public $READ_SHIPMENT_URL = "readShipment";
    public $DELIVERY_TYPES_URL = "deliveryTypes";
    public $PAYMENT_TYPES_URL = "paymentTypes";
    public $RATES_URL = "rates";

    public $PROLONG_RESERVE_URL = "prolongReserve";
    public $ORDER_GOODS_URL = "orderGoods";
```

```

public $DESC_BY_CATEGORY = "goodsDescriptionByCategory";
public $ZIP_CATEGORIES_URL = "catalogsZip/";
public $GOODS_IMAGES_CAT = "goodsImagesByCategory";
}

$config = new Config();
//service initialization
class HttpClient {
    public function get($url) {
        try {
            return file_get_contents($url);
        } catch (Exception $e) {
            return '';
        }
    }
}

$httpClient = new HttpClient();
class UrlBuilder {
    private $encoding = "UTF-8";
    private $slash = "/";
    private $config;
    private $amp = "&";
    /** catalog */
    private $usernameParam = "username=";
    private $passwordParam = "&password=";
    private $tokenParam = "oauth_token=";
    private $stickerParam = "&sticker=";
    private $serialNumberParam = "&serialnumber=";
    private $showDeletedParam = "&showDeleted=";
    /** documents */
    private $documentTypeIdParam = "&documentTypeId=";
    private $beginDateParam = "&beginDate=";
    private $endDateParam = "&endDate=";
    private $documentIdParam = "&documentId=";
    private $organizationIdParam = "&organizationId=";
    private $dateParam = "&date=";
    private $document1IdParam = "&document1Id=";
    private $document2IdParam = "&document2Id=";
    private $commentParam = "&comment=";
    private $goodsStringParam = "&goodsString=";
    private $reservePriceParam = "&reservePrice=";

    private $goodsUid= "&goodsUid=";
    private $pageNumber = "&pageNumber=";

    /** delivery */
    private $packIdParam = "&packId=";
    private $documentIdsParam = "&documentIds=";
    private $deliveryDateParam = "&deliveryDate=";
    private $deliveryTypeParam = "&deliveryType=";

    public function __construct($config) {
        $this->config = $config;
    }
}

// http://services.netlab.ru/rest/authentication/token.xml?username=<username>&password=<password>
public function getAuthenticationUrl($username, $password) {
    return $this->config->ROOT_URL . $this->config->AUTHENTICATION_URL . $this->config->RESPONSE_FORMAT . $this->usernameParam
        . $username . $this->passwordParam . $password;
}

// http://services.netlab.ru/rest/catalogs/list.xml?oauth_token=<token>
public function getCatalogsActionUrl($token) {
    return $this->config->ROOT_URL . $this->config->CATALOGS_URL . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}

```

```

    }
// http://services.netlab.ru/rest/catalogs/<catalogName>.xml?oauth_token=<token>
public function getCatalogActionUrl($token, $catalogName) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($catalogName)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
//
http://services.netlab.ru/rest/catalogs/<catalogName>/<categoryID>.xml?oauth_token=<tok
en>
public function getCategoryActionUrl($token, $catalogName, $categoryId) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($catalogName)
    . $this->slash . $categoryId . $this->config->RESPONSE_FORMAT . $this->tokenParam
. $token;
}
//
http://services.netlab.ru/rest/catalogs/versions/2/<catalogName>/<categoryID>.xml?oauth_
token=<token>&showDeleted=<showDeleted>
public function getCategoryWithDelActionUrl($token, $catalogName, $categoryId,
$showDeleted) {
    return $this->config->ROOT_URL . $this->config->VERSION_URL . $this-
>encode($catalogName)
    . $this->slash . $categoryId . $this->config->RESPONSE_FORMAT . $this->tokenParam
. $token . $this->showDeletedParam . $showDeleted;
}
//
http://services.netlab.ru/rest/catalogs/allGoods/versions/<versionId>.xml?oauth_token=T
OKEN&showDeleted=0
public function getAllGoodsActionUrl($token, $versionId, $showDeleted) {
    return $this->config->ROOT_URL . $this->config->ALLGOODS_URL . $this->slash .
$versionId . $this->config->RESPONSE_FORMAT
    . $this->tokenParam . $token . $this->showDeletedParam . $showDeleted;
}
//
http://services.netlab.ru/rest/catalogs/<catalogName>/<categoryID>/<goodsId>.xml?oauth_
token=<token>
public function getGoodsActionUrl($token, $catalogName, $categoryId, $goodsId) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($catalogName)
    . $this->slash . $categoryId . $this->slash . $goodsId . $this->config-
>RESPONSE_FORMAT . $this->tokenParam . $token;
}
//
http://services.netlab.ru/rest/catalogs/goodsByUid/<goodsId>.xml?oauth_token=<token>
public function getGoodsByUidActionUrl($token, $goodsId) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($this->config->GOODS_BY_UID_URL)
    . $this->slash . $goodsId . $this->config->RESPONSE_FORMAT . $this->tokenParam .
$token;
}
//
http://services.netlab.ru/rest/catalogs/newGoods.xml?oauth_token=<token>
public function getNewGoodsActionUrl($token) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($this->config->NEW_GOODS_URL) . $this->config->RESPONSE_FORMAT
    . $this->tokenParam . $token;
}
//
http://services.netlab.ru/rest/catalogs/usedGoods.xml?oauth_token=<token>
public function getUsedGoodsActionUrl($token) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($this->config->USED_GOODS_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
//
http://services.netlab.ru/rest/catalogs/dateBySticker.xml?oauth_token=<token>&sticker=<
stickerNumber>

```

```

public function getDateByStickerActionUrl($token, $stickerNumber) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($this->config->DATE_BY_STICKER_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>stickerParam . $stickerNumber;
}
// http://services.netlab.ru/rest/catalogs/goodsByPartnumber/<partNumber>.xml?oauth_token=<token>
public function getGoodsByPartNumberActionUrl($token, $partNumber) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($this->config->GOODS_BY_PARTNUMBER_URL)
    . $this->slash . $partNumber . $this->config->RESPONSE_FORMAT . $this->tokenParam
. $token;
}
// http://services.netlab.ru/rest/catalogs/dateBySerialNumber.xml?oauth_token=<token>&seri
alnumber=<serialNumber>
public function getDateBySerialNumberActionUrl($token, $serialNumber) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($this->config->DATE_SERIAL_NUMBER_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>serialNumberParam . $serialNumber;
}
// http://services.netlab.ru/rest/catalogs/info.xml?oauth_token=<token>
public function getInfoActionUrl($token) {
    return $this->config->ROOT_URL . $this->config->CATEGORIES_URL . $this-
>encode($this->config->INFO_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
// http://services.netlab.ru/rest/catalogsZip/goodsDescriptionByCategory/<categoryId>/<pag
eNumber>.xml?oauth_token=<token>
public function getDescriptionByCategoryActionUrl($token, $categoryId, $pageNumber)
{
    return $this->config->ROOT_URL . $this->config->ZIP_CATEGORIES_URL . $this-
>encode($this->config->DESC_BY_CATEGORY)
    . $this->slash . $categoryId . $this->slash . $pageNumber . $this->config-
>RESPONSE_FORMAT . $this->tokenParam . $token;
}
// http://services.netlab.ru/rest/catalogsZip/goodsImagesByCategory/<categoryId>.xml?oauth_
token=<token>
public function getImagesByCategoryActionUrl($token, $categoryId) {
    return $this->config->ROOT_URL . $this->config->ZIP_CATEGORIES_URL . $this-
>encode($this->config->GOODS_IMAGES_CAT)
    . $this->slash . $categoryId . $this->config->RESPONSE_FORMAT . $this-
>tokenParam . $token;
}

/** Documents services */
// http://services.netlab.ru/rest/documents/organizations.xml?oauth_token=<token>
public function getOrganizationsActionUrl($token) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->ORGANIZATIONS_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
// http://services.netlab.ru/rest/documents/documentTypes.xml?oauth_token=<token>
public function getDocumentTypesActionUrl($token) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->DOCUMENTS_TYPES_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
// http://services.netlab.ru/rest/documents/list.xml?oauth_token=<token>&documentTypeId=<t
oken>

```

```

    $typeId>&beginDate=<beginInterval>&endDate=<endInterval>
    public function listDocumentsUrl($token, $typeId, $beginDate, $endDate) {
        return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->LIST_URL)
        . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentTypeIdParam . $typeId . $this->beginDateParam . $beginDate . $this-
>endDateParam . $endDate;
    }
// http://services.netlab.ru/rest/documents/get.xml?oauth_token=<token>&documentId=<Id>
    public function readActionUrl($token, $documentId) {
        return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->GET_URL)
        . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentIdParam . $documentId;
    }
//
http://services.netlab.ru/rest/documents/new.xml?oauth_token=<token>&documentTypeId=<do
cumentType>&organizationId=<org>&date=<date>&comment=<comment>&goodsString=<goods>
    public function createActionUrl($token, $typeId, $organizationId, $date, $comment,
$goodsString) {
        return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->NEW_URL)
        . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentTypeIdParam . $typeId . $this->organizationIdParam . $organizationId . $this-
>dateParam . $date
        . $this->commentParam . $comment . $this->goodsStringParam . $goodsString;
    }
//
http://services.netlab.ru/rest/documents/update.xml?oauth_token=<token>&documentTypeId=
<documentType>&documentId=<docId>&organizationId=<org>&date=<date>&comment=<comment>&go
odsString=<goods>
    public function updateActionUrl($token, $typeId, $documentId, $organizationId,
$date, $comment, $goodsString) {
        return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->UPDATE_URL)
        . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentTypeIdParam . $typeId . $this->documentIdParam . $documentId
        . $this->organizationIdParam . $organizationId . $this->dateParam . $date
        . $this->commentParam . $comment . $this->goodsStringParam . $goodsString;
    }
//
http://services.netlab.ru/rest/documents/change.xml?oauth_token=<token>&documentTypeId=
<documentType>&documentId=<docId>&goodsString=<goodsChanges>
    public function changeActionUrl($token, $typeId, $documentId, $goodsString) {
        return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->CHANGE_URL)
        . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentTypeIdParam . $typeId . $this->documentIdParam . $documentId . $this-
>goodsStringParam . $goodsString;
    }
//
http://services.netlab.ru/rest/documents/delete.xml?oauth_token=<token>&documentTypeId=
<documentType>&documentId=<docId>
    public function deleteActionUrl($token, $typeId, $documentId) {
        return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->DELETE_URL)
        . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentTypeIdParam . $typeId . $this->documentIdParam . $documentId;
    }
//
http://services.netlab.ru/rest/documents/newReserve.xml?oauth_token=<token>&organizatio
nId=<organization>&date=<date>&comment=<comment>&goodsString=<goods>&reservePrice=<rese
rvePrice>
    public function createReserveUrl($token, $organizationId, $date, $comment,
$goodsString, $reservePrice) {

```

```

    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->NEW_RESERVE_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>organizationIdParam . $organizationId
    . $this->dateParam . $date . $this->commentParam . $comment . $this-
>goodsStringParam . $goodsString. $this->reservePriceParam . $reservePrice;
}
// http://services.netlab.ru/rest/documents/repairGoodsList.xml?oauth_token=<token>
public function repairGoodsListUrl($token) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->REPAIR_GOODS_LIST_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
//
http://services.netlab.ru/rest/documents/readShipment.xml?oauth_token=<token>&documentI
d=<document>
public function readShipmentUrl($token, $documentId) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->READ_SHIPMENT_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentIdParam . $documentId;
}
//
http://services.netlab.ru/rest/documents/mergeReserves.xml?oauth_token=<token>&document
1Id=<document1Id>&document2Id=<document2Id>
public function mergeReservesActionUrl($token, $document1Id, $document2Id) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->MERGE_RESERVES_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>document1IdParam . $document1Id . $this->document2IdParam . $document2Id;
}
//
http://services.netlab.ru/rest/documents/prolongReserve.xml?oauth_token=<token>&documen
tId=<documentId>&date=<date>
public function prolongReserveActionUrl($token, $documentId, $date){
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->PROLONG_RESERVE_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentIdParam . $documentId . $this->dateParam . $date;
}
//
http://services.netlab.ru/rest/documents/orderGoods.xml?oauth_token=<token>&documentId=
<documentId>&goodsUid=<goodsUid>
public function orderGoodsActionUrl($token, $documentId, $goodsUid) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL
        . $this->encode($this->config->ORDER_GOODS_URL) . $this->config-
>RESPONSE_FORMAT . $this->tokenParam . $token
        . $this->documentIdParam . $documentId . $this->goodsUid . $goodsUid;
}
//
http://services.netlab.ru/rest/documents/splitReserve.xml?oauth_token=<token>&documentI
d=<document>&goodsString=<goodsId>
public function splitReserveActionUrl($token, $documentId, $goodsString) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->SPLIT_RESERVE_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentIdParam . $documentId . $this->goodsStringParam . $goodsString;
}
// http://www.netlab.ru/ISAPI/TestISAPI.dll?18253&999555
public function goodsImagesActionUrl($clientImageId, $goodsId) {
    return $this->config->IMAGE_URL . $clientImageId . $this->amp . $goodsId;
}
//
http://services.netlab.ru/rest/documents/readShipment.xml?oauth_token=<token>&documentI
d=<document>

```

```

public function readShipmentUrl($token, $documentId) {
    return $this->config->ROOT_URL . $this->config->DOCUMENTS_URL . $this-
>encode($this->config->READ_SHIPMENT_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentIdParam . $documentId;
}
/** Delivery services */
//
http://services.netlab.ru/rest/delivery/list.xml?oauth_token=<token>&beginDate=<beginIn-
terval>&endDate=<endInterval>
public function getListDeliveryDocumentsUrl($token, $beginDate, $endDate) {
    return $this->config->ROOT_URL . $this->config->DELIVERY_URL . $this-
>encode($this->config->LIST_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>beginDateParam . $beginDate . $this->endDateParam . $endDate;
}
//
http://services.netlab.ru/rest/delivery/docs.xml?oauth_token=<token>&beginDate=<beginIn-
terval>&endDate=<endInterval>
public function getAvailableDeliveryDocumentsUrl($token, $beginDate, $endDate) {
    return $this->config->ROOT_URL . $this->config->DELIVERY_URL . $this-
>encode($this->config->DOCS_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>beginDateParam . $beginDate . $this->endDateParam . $endDate;
}
//
http://services.netlab.ru/rest/delivery/details.xml?oauth_token=<token>&packId=<packId>
public function getDeliveryDetailsUrl($token, $packId) {
    return $this->config->ROOT_URL . $this->config->DELIVERY_URL . $this-
>encode($this->config->DETAILS_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>packIdParam . $packId ;
}
//
http://services.netlab.ru/rest/delivery/dates.xml?oauth_token=<token>&beginDate=<beginIn-
terval>&endDate=<endInterval>
public function getDeliveryDatesUrl($token, $beginDate, $endDate) {
    return $this->config->ROOT_URL . $this->config->DELIVERY_URL . $this-
>encode($this->config->DATES_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>beginDateParam . $beginDate . $this->endDateParam . $endDate;
}
// http://services.netlab.ru/rest/delivery/deliveryTypes.xml?oauth_token=<token>
public function getDeliveryTypesUrl($token) {
    return $this->config->ROOT_URL . $this->config->DELIVERY_URL . $this-
>encode($this->config->DELIVERY_TYPES_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
// http://services.netlab.ru/rest/delivery/paymentTypes.xml?oauth_token=<token>
public function getDeliveryPaymentTypesUrl($token) {
    return $this->config->ROOT_URL . $this->config->DELIVERY_URL . $this-
>encode($this->config->PAYMENT_TYPES_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token;
}
//
http://services.netlab.ru/rest/delivery/rates.xml?oauth_token=<token>&documentIds=<docu-
mentIds>&deliveryDate=<deliveryDate>&deliveryType=<deliveryType>
public function getDeliveryRatesUrl($token, $documentIds, $deliveryDate,
$deliveryType) {
    return $this->config->ROOT_URL . $this->config->DELIVERY_URL . $this-
>encode($this->config->RATES_URL)
    . $this->config->RESPONSE_FORMAT . $this->tokenParam . $token . $this-
>documentIdsParam . $documentIds
    . $this->deliveryDateParam . $deliveryDate . $this->deliveryTypeParam .
$deliveryType;
}

```

```

}

private function encode($param) {
    try {
        $result = urlencode ($param);
        return $result;
    } catch (Exception $ex) {
        return null;
    }
}

$urlBuilder = new UrlBuilder($config);
class TokenParser {

    private $quote = '"';
    private $tokenTag = 'token":';

    public function parse($response) {
        if ($response == null) {
            return null;
        }
        $index = strpos($response,$this->tokenTag);
        if ($index == false) {
            return null;
        }
        $index = $index + strlen($this->tokenTag);
        $index2 = strpos($response,$this->quote, $index);
        if ($index2 == false) {
            return null;
        }
        return substr($response,$index, $index-$index2+1);
    }
}

class Service {
    private $config;
    private $urlBuilder;
    private $client;
    private $parser;

    public function __construct($config,$urlBuilder,$client) {
        $this->urlBuilder = $urlBuilder;
        $this->config = $config;
        $this->client = $client;
        $this->parser = new TokenParser();
    }

    private function sleep() {
        $time = round(microtime(true) * 1000) + $this->config->REQUEST_DELAY_PERIOD;
        while ($time > round(microtime(true) * 1000)) {
            try {
                usleep($this->config->REQUEST_DELAY_PERIOD*1000 + 1);
            } catch (Exception $ex) {
            }
        }
    }

    public function getToken($username, $password) {
        $this->sleep();
        $url = $this->urlBuilder->getAuthenticationUrl($username, $password);
        $response = $this->client->get($url);
        return $this->parser->parse($response);
    }

    /**
     * Catalog services */
    public function getCatalogsAction($token) {
        $this->sleep();
        $url = $this->urlBuilder->getCatalogsActionUrl($token);
    }
}

```

```

        return $this->client->get($url);
    }

    public function getCatalogAction($token, $catalogName) {
        $this->sleep();
        $url = $this->urlBuilder->getCatalogActionUrl($token, $catalogName);
        return $this->client->get($url);
    }

    public function getCategoryAction($token, $catalogName, $categoryId) {
        $this->sleep();
        $url = $this->urlBuilder->getCategoryActionUrl($token, $catalogName,
$categoryId);
        return $this->client->get($url);
    }

    public function getCategoryWithDelAction($token, $catalogName, $categoryId,
$showDeleted) {
        $this->sleep();
        $url = $this->urlBuilder->getCategoryWithDelActionUrl($token, $catalogName,
$categoryId, $showDeleted);
        return $this->client->get($url);
    }

    public function getAllGoodsAction($token, $versionId, $showDeleted) {
        $this->sleep();
        $url = $this->urlBuilder->getAllGoodsActionUrl($token, $versionId, $showDeleted);
        return $this->client->get($url);
    }

    public function getGoodsAction($token, $catalogName, $categoryId, $goodsId) {
        $this->sleep();
        $url = $this->urlBuilder->getGoodsActionUrl($token, $catalogName, $categoryId,
$goodsId);
        return $this->client->get($url);
    }

    public function getGoodsByUidAction($token, $goodsId) {
        $this->sleep();
        $url = $this->urlBuilder->getGoodsByUidActionUrl($token, $goodsId);
        return $this->client->get($url);
    }

    public function getNewGoodsAction($token) {
        $this->sleep();
        $url = $this->urlBuilder->getNewGoodsActionUrl($token);
        return $this->client->get($url);
    }

    public function getUsedGoodsAction($token) {
        $this->sleep();
        $url = $this->urlBuilder->getUsedGoodsActionUrl($token);
        return $this->client->get($url);
    }

    public function getDateByStickerAction($token, $stickerNumber) {
        $this->sleep();
        $url = $this->urlBuilder->getDateByStickerActionUrl($token, $stickerNumber);
        return $this->client->get($url);
    }

    public function getGoodsByPartNumberAction($token, $partNumber) {
        $this->sleep();
        $url = $this->urlBuilder->getGoodsByPartNumberActionUrl($token, $partNumber);
        return $this->client->get($url);
    }

    public function getDateBySerialNumberAction($token, $serialNumber) {
        $this->sleep();
        $url = $this->urlBuilder->getDateBySerialNumberActionUrl($token, $serialNumber);
        return $this->client->get($url);
    }

    public function getInfoAction($token) {
        $this->sleep();
        $url = $this->urlBuilder->getInfoActionUrl($token);
        return $this->client->get($url);
    }
}

```

```

}

public function getDescriptionByCategoryAction($token, $categoryId, $pageNumber) {
    $this->sleep();
    $url = $this->urlBuilder->getDescriptionByCategoryActionUrl($token,
$categoryId, $pageNumber);
    return $this->client->get($url);
}

public function getImagesByCategoryAction($token, $categoryId) {
    $this->sleep();
    $url = $this->urlBuilder->getImagesByCategoryActionUrl($token, $categoryId);
    return $this->client->get($url);
}

/** Documents services */
public function getOrganizationsAction($token) {
    $this->sleep();
    $url = $this->urlBuilder->getOrganizationsActionUrl($token);
    return $this->client->get($url);
}

public function getDocumentTypesAction($token) {
    $this->sleep();
    $url = $this->urlBuilder->getDocumentTypesActionUrl($token);
    return $this->client->get($url);
}

public function listDocuments($token, $typeId, $beginDate, $endDate) {
    $this->sleep();
    $url = $this->urlBuilder->listDocumentsUrl($token, $typeId, $beginDate,
$endDate);
    return $this->client->get($url);
}

public function readAction($token, $documentId) {
    $this->sleep();
    $url = $this->urlBuilder->readActionUrl($token, $documentId);
    return $this->client->get($url);
}

public function createAction($token, $typeId, $organizationId, $date, $comment,
$goodsString) {
    $this->sleep();
    $url = $this->urlBuilder->createActionUrl($token, $typeId, $organizationId,
$date, $comment, $goodsString);
    return $this->client->get($url);
}

public function updateAction($token, $typeId, $documentId, $organizationId, $date,
$comment, $goodsString) {
    $this->sleep();
    $url = $this->urlBuilder->updateActionUrl($token, $typeId, $documentId,
$organizationId, $date, $comment, $goodsString);
    return $this->client->get($url);
}

public function changeAction($token, $typeId, $documentId, $goodsString) {
    $this->sleep();
    $url = $this->urlBuilder->changeActionUrl($token, $typeId, $documentId,
$goodsString);
    return $this->client->get($url);
}

public function deleteAction($token, $typeId, $documentId) {
    $this->sleep();
    $url = $this->urlBuilder->deleteActionUrl($token, $typeId, $documentId);
    return $this->client->get($url);
}

public function createReserve($token, $organizationId, $date, $comment,
$goodsString, $reservePrice) {
    $this->sleep();
    $url = $this->urlBuilder->createReserveUrl($token, $organizationId, $date,
$comment, $goodsString, $reservePrice);
}

```

```

    return $this->client->get($url);
}

public function repairGoodsList($token) {
    $this->sleep();
    $url = $this->urlBuilder->repairGoodsListUrl($token);
    return $this->client->get($url);
}

public function readShipment($token, $documentId) {
    $this->sleep();
    $url = $this->urlBuilder->readShipmentUrl($token, $documentId);
    return $this->client->get($url);
}

public function mergeReserves($token, $document1Id, $document2Id) {
    $this->sleep();
    $url = $this->urlBuilder->mergeReservesActionUrl($token, $document1Id,
$document2Id);
    return $this->client->get($url);
}

public function prolongReserve($token, $documentId, $date) {
    $this->sleep();
    $url = $this->urlBuilder->prolongReserveActionUrl($token, $documentId, $date);
    return $this->client->get($url);
}

public function orderGoods($token, $documentId, $goodsUid) {
    $this->sleep();
    $url = $this->urlBuilder->orderGoodsActionUrl($token, $documentId, $goodsUid);
    return $this->client->get($url);
}

public function splitReserve($token, $documentId, $goodsString) {
    $this->sleep();
    $url = $this->urlBuilder->splitReserveActionUrl($token, $documentId,
$goodsString);
    return $this->client->get($url);
}

public function goodsImagesAction($clientImageId, $goodsId) {
    $this->sleep();
    $url = $this->urlBuilder->goodsImagesActionUrl($clientImageId, $goodsId);
    return $this->client->get($url);
}

/** Delivery services */
public function getListDeliveryDocuments($token, $beginDate, $endDate) {
    $this->sleep();
    $url = $this->urlBuilder->getListDeliveryDocumentsUrl($token, $beginDate,
$endDate);
    return $this->client->get($url);
}

public function getAvailableDeliveryDocuments($token, $beginDate, $endDate) {
    $this->sleep();
    $url = $this->urlBuilder->getAvailableDeliveryDocumentsUrl($token, $beginDate,
$endDate);
    return $this->client->get($url);
}

public function getDeliveryDetails($token, $packId) {
    $this->sleep();
    $url = $this->urlBuilder->getDeliveryDetailsUrl($token, $packId);
    return $this->client->get($url);
}

public function getDeliveryDates($token, $beginDate, $endDate) {
    $this->sleep();
    $url = $this->urlBuilder->getDeliveryDatesUrl($token, $beginDate, $endDate);
    return $this->client->get($url);
}

public function getDeliveryTypes($token) {
    $this->sleep();
}

```

```

    $url = $this->urlBuilder->getDeliveryTypesUrl($token);
    return $this->client->get($url);
}
public function getDeliveryPaymentTypes($token) {
    $this->sleep();
    $url = $this->urlBuilder->getDeliveryPaymentTypesUrl($token);
    return $this->client->get($url);
}
public function getDeliveryRates($token, $documentIds, $deliveryDate, $deliveryType)
{
    $this->sleep();
    $url = $this->urlBuilder->getDeliveryRatesUrl($token, $documentIds,
$deliveryDate, $deliveryType);
    return $this->client->get($url);
}
// public function
$service = new Service($config, $urlBuilder, $httpClient);
//get url params
$categoryParam = 'categoryId';
$catalogNameParam = 'catalogName';
$goodsIdParam = 'goodsId';
$goodsInfoByIdParam = 'goodsInfoById';
$btnIdParam = 'btnId';
$btnId = 'examplePageId';

/** catalog */
$usernameParam = "username";
$passwordParam = "password";
$tokenParam = "oauth_token";
$catalogNameParam = "catalogName";
$stickerNumberParam = "stickerNumber";
$partNumberParam = "partNumber";
$serialNumberParam = "serialNumber";
$versionIdParam = "versionId";
$showDeletedParam = "showDeleted";
$pageNumberParam = "pageNumber";

/** documents */
$documentTypeIdParam = "documentTypeId";
$beginDateParam = "beginDate";
$endDateParam = "endDate";
$documentIdParam = "documentId";
$organizationIdParam = "organizationId";
$dateParam = "date";
$commentParam = "comment";
$goodsStringParam = "goodsString";
$reservePriceParam = "reservePrice";
$document1IdParam = "document1Id";
$document2IdParam = "document2Id";
$clientImageIdParam = "clientImageId";
$goodsUidParam = "goodsUid";

/** delivery */
$documentIdsParam = "documentIds";
$packIdParam = "packId";
$deliveryDateParam = "deliveryDate";
$deliveryTypeParam = "deliveryType";

$emptyString = '';
$categoryId = null;
$catalogName = null;
$goodsId = null;
$partNumber = null;

```

```

$goodsInfoById = null;

$token = null;
$username = null;
$password = null;

$stickerNumber = null;
$serialNumber = null;
$versionId = null;
$showDeleted = null;
/** documents */
$documentTypeId = null;
$beginDate = null;
$endDate = null;
$documentId = null;
$organizationId = null;
$date = null;
$comment = null;
$goodsString = null;
$reservePrice = null;
$document1Id = null;
$document2Id = null;
$clientImageId = null;
$goodsUid = null;
$pageNumber = null;
/** delivery */
$documentIds = null;
$spackId = null;
$deliveryDate = null;
$deliveryType = null;

try {
    if (isset($_REQUEST[$tokenParam])) {
        $token = $_GET[$tokenParam];
    }
    if (isset($_REQUEST[$usernameParam])) {
        $username = $_GET[$usernameParam];
    }
    if (isset($_REQUEST[$passwordParam])) {
        $password = $_GET[$passwordParam];
    }
    if (isset($_REQUEST[$categoryIdParam])) {
        $categoryId = $_GET[$categoryIdParam];
    }
    if (isset($_REQUEST[$catalogNameParam])) {
        $catalogName = $_GET[$catalogNameParam];
    }
    if (isset($_REQUEST[$goodsIdParam])) {
        $goodsId = $_GET[$goodsIdParam];
    }
    if (isset($_REQUEST[$stickerNumberParam])) {
        $stickerNumber = $_GET[$stickerNumberParam];
    }
    if (isset($_REQUEST[$partNumberParam])) {
        $partNumber = $_GET[$partNumberParam];
    }
    if (isset($_REQUEST[$serialNumberParam])) {
        $serialNumber = $_GET[$serialNumberParam];
    }
    if (isset($_REQUEST[$goodsInfoByIdParam])) {
        $goodsInfoById = $_GET[$goodsInfoByIdParam];
    }
    if (isset($_REQUEST[$documentTypeIdParam])) {
        $documentTypeId = $_GET[$documentTypeIdParam];
    }
}

```

```

if (isset($_REQUEST[$beginDateParam])) {
    $beginDate = $_GET[$beginDateParam];
}
if (isset($_REQUEST[$endDateParam])) {
    $endDate = $_GET[$endDateParam];
}
if (isset($_REQUEST[$documentIdParam])) {
    $documentId = $_GET[$documentIdParam];
}
if (isset($_REQUEST[$btnIdParam])) {
    $btnId = $_GET[$btnIdParam];
}
if (isset($_REQUEST[$dateParam])) {
    $date = $_GET[$dateParam];
}
if (isset($_REQUEST[$organizationIdParam])) {
    $organizationId = $_GET[$organizationIdParam];
}
if (isset($_REQUEST[$reservePriceParam])) {
    $reservePrice = $_GET[$reservePriceParam];
}
if (isset($_REQUEST[$goodsStringParam])) {
    $goodsString = $_GET[$goodsStringParam];
}
if (isset($_REQUEST[$commentParam])) {
    $comment = $_GET[$commentParam];
}
if (isset($_REQUEST[$document1IdParam])) {
    $document1Id = $_GET[$document1IdParam];
}
if (isset($_REQUEST[$document2IdParam])) {
    $document2Id = $_GET[$document2IdParam];
}
if (isset($_REQUEST[$clientImageIdParam])) {
    $clientImageId = $_GET[$clientImageIdParam];
}
if (isset($_REQUEST[$versionIdParam])) {
    $versionId = $_GET[$versionIdParam];
}
if (isset($_REQUEST[$showDeletedParam])) {
    $showDeleted = $_GET[$showDeletedParam];
}
if (isset($_REQUEST[$documentIdsParam])) {
    $documentIds = $_GET[$documentIdsParam];
}
if (isset($_REQUEST[$packIdParam])) {
    $packId = $_GET[$packIdParam];
}
if (isset($_REQUEST[$deliveryDateParam])) {
    $deliveryDate = $_GET[$deliveryDateParam];
}
if (isset($_REQUEST[$deliveryTypeParam])) {
    $deliveryType = $_GET[$deliveryTypeParam];
}
if (isset($_REQUEST[$goodsUidParam])) {
    $goodsUid = $_GET[$goodsUidParam];
}
if (isset($_REQUEST[$pageNumberParam])) {
    $pageNumber = $_GET[$pageNumberParam];
}
} catch (Exception $e) {
}
//data loading
$responseStr = null;
try {

```

```

if ($btnId != null) {
    if ($btnId == 'getTokenBtnId') { //Аутентификация
        $responseStr = $service->getToken($username, $password);
    }
    else if ($btnId == 'getCatalogsActionBtnId') { //Список каталогов товаров
        $responseStr = $service->getCatalogsAction($token);
    }
    else if ($btnId == 'getCatalogActionBtnId') { //Список категорий товаров каталога
        $responseStr = $service->getCatalogAction($token, $catalogId);
    }
    else if ($btnId == 'getCategoryActionBtnId') { //Список товаров категории
        $responseStr = $service->getCategoryAction($token, $catalogName, $catalogId);
    }
    else if ($btnId == 'getCategoryWithDelActionBtnId') { //Список товаров категории
        $responseStr = $service->getCategoryWithDelAction($token, $catalogName,
$catalogId, $showDeleted);
    }
    else if ($btnId == 'getAllGoodsActionBtnId') { //Описание товара по id
        $responseStr = $service->getAllGoodsAction($token, $versionId, $showDeleted);
    }
    else if ($btnId == 'getGoodsActionBtnId') { //Описание товара по id
        $responseStr = $service->getGoodsAction($token, $catalogName, $catalogId,
$goodsId);
    }
    else if ($btnId == 'getGoodsByUidActionBtnId') { //Информация о ценах и
количестве товара по коду товара
        $responseStr = $service->getGoodsByUidAction($token, $goodsId);
    }
    else if ($btnId == 'getNewGoodsActionBtnId') { //Список новых товаров
        $responseStr = $service->getNewGoodsAction($token);
    }
    else if ($btnId == 'getUsedGoodsActionBtnId') { //Список уцененных товаров
        $responseStr = $service->getUsedGoodsAction($token);
    }
    else if ($btnId == 'getDateByStickerActionBtnId') { //Поиск даты накладной по
номеру стикера
        $responseStr = $service->getDateByStickerAction($token, $stickerNumber);
    }
    else if ($btnId == 'getGoodsByPartNumberActionBtnId') { //Поиск товара по
partnumber
        $responseStr = $service->getGoodsByPartNumberAction($token, $partNumber);
    }
    else if ($btnId == 'getDateBySerialNumberActionBtnId') { //Поиск даты накладной
по номеру SN товара
        $responseStr = $service->getDateBySerialNumberAction($token, $serialNumber);
    }
    else if ($btnId == 'getInfoActionBtnId') { //Курсы валют
        $responseStr = $service->getInfoAction($token);
    }
    else if ($btnId == 'goodsDescriptionByCategoryActionBtnId') { // Получение
описание товара по категориям
        $responseStr = $service->getDescriptionByCategoryAction($token,
$catalogId, $pageNumber);
    }
    else if ($btnId == 'goodsImagesByCategoryActionBtnId') { // Получение
изображений товара по категории
        $responseStr = $service->getImagesByCategoryAction($token, $catalogId);
    }

//documents
else if ($btnId == 'getOrganizationsActionBtnId') { //Список юридических лиц
    $responseStr = $service->getOrganizationsAction($token);
}
else if ($btnId == 'getDocumentTypesActionBtnId') { //Список доступных типов
}

```

ДОКУМЕНТОВ

```

    $responseStr = $service->getDocumentTypesAction($token);
}
else if ($btnId == 'listDocumentsBtnId') { //Список доступных документов
    $responseStr = $service->listDocuments($token, $documentTypeId, $beginDate,
$endDate);
}
else if ($btnId == 'readActionBtnId') { //Список товаров документа
    $responseStr = $service->readAction($token, $documentId);
}
else if ($btnId == 'createDocumentBtnId') {
    $responseStr = $service->createAction($token, $documentTypeId,
$organizationId, $date, $comment, $goodsString);
}
else if ($btnId == 'updateActionBtnId') { //Редактирование документа
    $responseStr = $service->updateAction($token, $documentTypeId, $documentId,
$organizationId, $date, $comment, $goodsString);
}
else if ($btnId == 'changeActionBtnId') { //Редактирование документа
    $responseStr = $service->changeAction($token, $documentTypeId, $documentId,
$goodsString);
}
else if ($btnId == 'deleteActionBtnId') { // Удаление документа
    $responseStr = $service->deleteAction($token, $documentTypeId, $documentId);
}
else if ($btnId == 'createReserveBtnId') { //Создание нового резерва
    $responseStr = $service->createReserve($token, $organizationId, $date,
$comment, $goodsString, $reservePrice);
}
else if ($btnId == 'repairGoodsListBtnId') { //Запрос списка товаров в ремонте
    $responseStr = $service->repairGoodsList($token);
}
else if ($btnId == 'readShipmentBtnId') { //Данные документа отгрузки
    $responseStr = $service->readShipment($token, $documentId);
}
else if ($btnId == 'mergeReservesActionBtnId') { // Слияние резервов
    $responseStr = $service->mergeReserves($token, $document1Id, $document2Id);
}
else if ($btnId == 'prolongReserveActionBtnId') { // Продление резерва
    $responseStr = $service->prolongReserve($token, $documentId, $date);
}
else if ($btnId == 'orderGoodsActionBtnId') { // Заказ отсутсвующих товаров
    $responseStr = $service->orderGoods($token, $documentId, $goodsUid);
}
else if ($btnId == 'splitReserveActionBtnId') { // Разбиение резерва
    $responseStr = $service->splitReserve($token, $documentId, $goodsString);
}
else if ($btnId == 'goodsImagesActionBtnId') { //Данные документа отгрузки
    $responseStr = $service->goodsImagesAction($clientImageId, $goodsId);
}
// delivery
else if ($btnId == 'getListDeliveryDocumentsActionBtnId') { // Получение списка
документов доставки
    $responseStr = $service->getListDeliveryDocuments($token, $beginDate,
$endDate);
}
else if ($btnId == 'getAvailableDeliveryDocumentsActionBtnId') { // Получение
списка доступных счетов для оформления доставки
    $responseStr = $service->getAvailableDeliveryDocuments($token, $beginDate,
$endDate);
}
else if ($btnId == 'getDeliveryDetailsActionBtnId') { // Получение подробной
информации документа доставки
    $responseStr = $service->getDeliveryDetails($token, $spackId);
}
}

```

```
    else if ($btnId == 'getDeliveryDatesActionBtnId') { // Получение списка доступных
дат доставки
        $responseStr = $service->getDeliveryDates($token, $beginDate, $endDate);
    }
    else if ($btnId == 'getDeliveryTypesActionBtnId') { // Получение возможных типов
доставки
        $responseStr = $service->getDeliveryTypes($token);
    }
    else if ($btnId == 'getDeliveryPaymentTypesActionBtnId') { // Получение списка
вариантов оплаты доставки
        $responseStr = $service->getDeliveryPaymentTypes($token);
    }
    else if ($btnId == 'getDeliveryRatesActionBtnId') { // Получение списка тарифов
доставки
        $responseStr = $service->getDeliveryRates($token, $documentIds, $deliveryDate,
$deliveryType);
    }
}
catch (Exception $ex) {
    return;
}
echo $responseStr;
?>
```

5. ПРИМЕРЫ ВЫВОДА МЕТОДОВ

В данном разделе представлены примеры работы методов

1. Модуль «АУТЕНТИФИКАЦИЯ»

метод authenticate

<http://services.netlab.ru/rest/authentication/token.json?username=<username>&password=<password>>

```
{} && {"tokenResponse": {"status": {"code": "200", "message": ""}, "data": {"token": "cb9e1cc2-fe34-4a38-a6fa-877daed809de", "expiredIn": "19.07.2016 15:07"}}}
```

2. МОДУЛЬ «КАТАЛОГ»

метод catalogsAction

http://services.netlab.ru/rest/catalogsZip/list.json?oauth_token=<token>

```
{} && {"catalogsResponse": {"status": {"code": "200", "message": ""}, "data": {"catalogs": [{"name": "Прайс-лист", "category": null}, {"name": "В наличии", "category": null}, {"name": "Распродажа", "category": null}, {"name": "Новинки", "category": null}]}}}
```

метод catalogAction

http://services.netlab.ru/rest/catalogsZip/<catalogName>.json?oauth_token=<token>

```
{} && {"catalogResponse": {"status": {"code": "200", "message": ""}, "data": {"name": "Прайс-лист", "category": [{"id": "110128", "name": "Ноутбуки LENOVO", "parentId": "110429", "leaf": true, "goods": null}, {"id": "110129", "name": "Ноутбуки HP", "parentId": "110429", "leaf": true, "goods": null}, {"id": "23", "name": "Офисная мебель, сейфы, шкафы", "parentId": "0", "leaf": false, "goods": null}, {"id": "25", "name": "Системы безопасности и видеонаблюдения", "parentId": "0", "leaf": false, "goods": null}]}}}
```

метод categoryAction

http://services.netlab.ru/rest/catalogsZip/<catalogName>/<categoryId>.json?oauth_token=<token>

```
{} && {"categoryResponse": {"status": {"code": "200", "message": ""}, "data": {"id": "1485609", "name": "Ноутбуки IRBIS", "parentId": "110429", "leaf": true, "goods": [{"id": "1485628", "properties": {"дата транзита": null, "количество на Калужской": 0, "количество на Курской": 0, "снят с производства": false, "НДС": "18%", "цена по категории N": 241.0000, "РРЦ": 2123.55, "гарантия": "1 год", "производитель": "Irbis", " удален": false, "цена по категории F": 221.0000, "цена по категории E": 223.0000, "цена по категории D": 224.0000, "количество в транзите": 0.00, "цена по категории С": 226.0000, "цена по категории В": 234.0000, "цена по категории А": 226.0100, "вес, кг": "2.5000", "количество на Лобненской": 5.0000, "объём, м^3": 0.0042, "id": 1485628, "название": "IRBIS NB116 [NB116] black 11.6\" {FHD IPS Atom Z8350/4Gb/32GB/W10}", "PN": "NB116 ", "удаленный склад": null}}, {"id": "1502770", "properties": {"дата транзита": null, "количество на Калужской": 0, "количество на Курской": 0, "снят с производства": false, "НДС": "18%", "цена по категории N": 176.5100, "гарантия": "1 год", "производитель": "Irbis", "удален": false, "цена по категории F": 159.0000, "цена по категории Е": 160.0000, "цена по категории D": 161.0000, "количество в транзите": 0.00, "цена по категории С": 162.0000, "цена по категории В": 168.1000, "цена по категории А": 162.6000, "вес, кг": "2.5000", "количество на
```

```

Лобненской":0.0000,"объём, м^3":0.0067,"id":1502770,"название":"IRBIS NB52 [NB52] white
14\" {HD Atom Z8350/2Gb/32GB/W10}","PN":"NB52","удаленный
склад":null}, {"id":1615185,"properties":{"дата транзита":null,"количество на
Калужской":0,"количество на Курской":0,"снят с производства":false,"НДС":"18%","цена по
категории N":229.0000,"гарантия":"1 год","производитель":"Irbis"," удален":false,"цена
по категории F":210.0000,"цена по категории Е":211.0000,"цена по категории
D":213.0000,"количество в транзите":0.00,"цена по категории С":214.0000,"цена по
категории В":222.0000,"цена по категории А":214.7600,"вес, кг":"9.1000","количество на
Лобненской":5.0000,"объём, м^3":0.0068,"id":1615185,"название":"IRBIS NB138 [nb138]
Dark blue 13.3\" {FHD IPS Cel N3350/3Gb/32GB/W10Pro}","PN":"nb138","удаленный
склад":null}}]}}}
```

метод categoryWithDelAction

http://services.netlab.ru/rest/catalogsZip/versions/2/<catalogName>/<categoryId>.json?oauth_token=<token>&showDeleted=<showDeleted>

```

{} && {"categoryResponse": {"status": {"code": "200", "message": ""}, "data":
{"id": "110129", "name": null, "parentId": null, "leaf": null, "goods": [{"id": 1213372, "properti
es": {"количество на Курской": 0, "цена по категории В": 1017.0000, "цена по категории
А": 1252.6704, "цена по категории D": 978.0000, "цена по категории С": 983.0000, "количество
на Калужской": 0, "цена по категории F": 969.0000, "цена по категории
Е": 974.0000, "id": 1213372, "удален": true, "PN": "B0N00AA", "количество на
Лобненской": 0, "количество в транзите": 0.00, "название": "B0N00AA HP Folio 13 i5-
2467M/4G/128G SSD/13.3\" HD/WiFi/BT/Cam/FPR/Win7 PRO", "цена по категории
N": 1114.0000}}, {"id": 1372131, "properties": {"количество на Курской": 0, "цена по
категории В": 1431.0000, "цена по категории А": 1338.7700, "цена по категории
D": 1374.0000, "цена по категории С": 1383.0000, "количество на Калужской": 0, "цена по
категории F": 1316.3900, "цена по категории Е": 1366.0000, "id": 1372131, "удален": true,
"PN": "K0H73ES", "количество на Лобненской": 0, "количество в транзите": 0.00, "название": "HP
EliteBook 850 Core i5-5200U 2.2GHz,15.6\" FHD LED AG Cam,8GB DDR3L(1),3G/1TB
5.4krpm,32Gb FlashCache [K0H73ES]", "цена по категории N": 1560.0000}}, {"id": 1397755, "properties": {"количество на Курской": 0, "цена по категории
В": 0.2600, "цена по категории А": 0.2800, "цена по категории D": 0.1800, "цена по категории
С": 0.2000, "количество на Калужской": 0, "цена по категории F": 0.1500, "цена по категории
Е": 0.1700, "id": 1397755, "удален": true, "PN": "05-4003-1", "количество на
Лобненской": 0, "количество в транзите": 0.00, "название": "РАЗЪЕМ F-разъем RG-6 GOLD REXANT
05-4003-1", "цена по категории N": 0.4600}}]}}}
```

метод goodsAction

http://services.netlab.ru/rest/catalogsZip/<catalogName>/<categoryId>/<goodsId>.json?oauth_token=<token>

```

{} && {"goodsResponse": {"status": {"code": "200", "message": ""}, "data": {"id": 1213372, "properties
": {"Объем жесткого диска SSD (ГБ)": "128", "Подсветка экрана ":"LED", "Поддержка замка
Kensington": "Нет", "Тип манипулятора ноутбука": "Multi-touch TouchPad", "Безопасность": "-",
"COM порт": "Нет", "Тип дисплея ":"Не сенсорный", "Беспроводной сигнал и
приемопередатчики": "-", "Поддержка карт памяти": "Есть", "Идентификация пользователя по
отпечаткам пальцев": "Нет", "HDMI": "Есть", "3G": "Нет", "ПРЕДУПРЕЖДЕНИЕ по ГО": "Гарантия на
ноутбук 12 месяцев, на аккумулятор - 12 месяцев (с даты отгрузки со склада
hp)", "Логистика": "-", "высота (см)": 34.5000, "Оперативная память
(МБ)": 4096, "gtdNumber": "10130200/290312/0004754/024", "Видеопамять": "Выделяемая из
оперативной", "Наушники / Микрофон": "Есть", "rusName": "[Ноутбук] B0N00AA HP Folio 13 i5-
2467M/4G/128G SSD/13.3\" HD/WiFi/BT/Cam/FPR/Win7 PRO", "Цвет крышки
ноутбука": "Серебристый", "Опции (приобретаются отдельно)": "(аксессуары HP) AU155AA,
AP355AA, BM868AA, BT796AA, AY052AA, AX727AA", "eSATA": "Нет", "pictureID": 91065, "LAN
(RJ45)": "Есть", "гарантия": "1 год", "Поверхность экрана": "Глянцевая", "Операционная
система": "Windows 7 Professional 64", "USB 3.0 (USB-C)
порты": "0", "DisplayPort": "Нет", "Оptический привод": "Нет", "Встроенный
```

микрофон": "Есть", "Описание расцветки": "Серебристый, Черный", "Процессор": "-", "Разрешение матрицы ноутбука": "1366 x 768", "USB 3.1 (USB-C) порты": "0", "инструкция": "http://h10025.www1.hp.com/ewfrf/wc/contacthp?lc=en&cc=uk&product=5210926&lang=en&cc=uk", "miniHDMI": "Нет", "Процессор ноутбука": "Intel Core i5", "Время работы от аккумулятора (часов)": "9.5", "Встроенные динамики": "Есть", "Слот расширения": "Нет", "mini D-Sub": "Нет", "Чипсет": "Intel HM65 Express", "USB 2.0 порты": "1", "Wi-Fi": "Есть", "Типы поддерживаемых карт": "SD, MMC", "Производитель": "Hp", "длина(см)": "46.2000", "Объем жесткого диска iSSD (ГБ)": "0", "Объем жесткого диска (ГБ)": "0", "WEB-камера": "Есть", "Размеры упаковки": "46.5 x 34.5 x 8 см", "Цвет подставки под запястья": "Серебристый", "Горячая линия производителя": "(495) 777-32-84 (Москва), 8-800-500-92-60 (бесплатный звонок из любого региона России). Пн -пт: с 9:00 до 18:00", "Формат накопителя ноутбука": "2.5\"", "Внешние источники информации": "--", "USB 3.0 порты": "1", "Размер L3 кэша (МБ)": "3", "Отделка подставки под запястья": "Текстурированный матовый алюминий", "Основные характеристики": "--", "Объем дискретной видеопамяти (МБ)": "0", "Видео": "--", "Питание": "--", "Коммуникации": "--", "Тип видеокарты": "Встроенная", "Количество ядер": "2", "LTE": "Нет", "Отделка крышки ноутбука": "Текстурированный матовый алюминий", "miniDisplayPort": "Нет", "Цвет корпуса": "Комбинированная расцветка", "GTIN": "00889899057346", "Диагональ (дюймов)": "13.3", "Thunderbolt": "Нет", "Характеристики процессора": "2467M 1.6 ГГц", "Латинские буквы": "Белые (подсвечиваются)", "Конфигурация": "--", "Русские буквы": "Белые (подсвечиваются)", "Клавиатура ноутбука": "Островного типа", "драйвер": "http://h10025.www1.hp.com/ewfrf/wc/contacthp?lc=en&cc=uk&product=5210926&lang=en&cc=uk", "Модель": "B0N00AA", "Макс. скорость вращения HDD ноутбука (об/мин)": "0", "Опции": "--", "Вес ноутбука с аккумулятором (грамм)": "1490", "Видеокарта": "HD Graphics 3000", "ширина(см)": "8.5000", "Вес брутто, грамм": "2411", "Дизайн и отделка": "--", "Bluetooth": "Есть", "id сертификата": "81715", "Кол-во слотов SODIMM": "1", "Тип устройства": "Ноутбук", "Интерфейс": "--", "id": "1213372", "Интерфейс накопителя ноутбука": "SATA", "D-Sub": "Нет", "Клавиатура и др. устройства ввода": "--", "partNumber": "B0N00AA", "описание": "Ноутбук B0N00AA от компании HP на базе процессора Intel Core i5 2467M 1.6 ГГц 3 МБ комплектуется оперативной памятью типа DDR III общим размером 4 ГБ.", "manufacturerCountry": "Китай", "Батарея": "Li-Ion (Литий-ионные)", "IEEE1394": "Нет", "Объем жесткого диска 2 (ГБ)": "0", "Кнопки ноутбука": "Touchpad вкл/выкл", "Накопители": "--", "масса(кг)": "2.4190", "сайт производителя": "http://www8.hp.com/ru/ru/home.html", "Экран": "--", "Прочие характеристики": "--", "USB 3.1 порты": "0", "Максимальный объем памяти (МБ)": "4096", "Тип оперативной памяти": "DDR-III", "Размеры": "31.9 x 2 x 22.2 см", "Блок питания": "Входит в комплект поставки", "Цвет клавиш ноутбука": "Черный", "название": "B0N00AA HP Folio 13 i5-2467M/4G/128G SSD/13.3\" HD/WiFi/BT/Cam/FPR/Win7 PRO", "S/PDIF (Оптический)": "Нет"} }

метод goodsDescriptionByUidAction

http://services.netlab.ru/rest/catalogsZip/goodsDescriptionByUid/<goodsId>.json?oauth_token=<token>

```
{
  } && {"goodsResponse": {"status": {"code": "200", "message": ""}}, "data": {"id": 1353192, "properties": {"Контроллер HDD": "SATA", "Количество внутренних отсеков 3,5\"/2,5\"": "0", "<b>Комплект поставки и опции</b>": "--", "Поддержка горячей замены HDD": "Есть", "Производитель": "Lenovo", "Серия": "ThinkServer", "Видеoadаптер": "Нет", "Тип оперативной памяти": "DDR-IV", "Типы поддерживаемых карт-памяти": "Нет", "Общий объем HDD (ГБ)": "0", "Сеть": "Есть", "Количество процессоров": "1, возможна установка 2", "Операционная система": "Нет", "Слоты расширения": "Есть", "Разъемы на передней панели": "USB", "Пульт ДУ": "Нет", "Поддержка 64bit": "Есть", "Скорость вращения HDD": "0", "Тип корпуса": "Для монтажа в стойку", "Объем оперативной памяти (ГБ)": "8", "Мышь": "Нет", "Кэш L3 (МБ)": "20", "<b>Процессор</b>": "--", "Тип поддерживаемой памяти": "DDR4", "Охлаждение": "Есть", "Кэш L2 (МБ)": "2", "Тип процессора": "Intel Xeon", "Количество ядер": "8", "Мощность блока питания (Вт)": "750", "<b>Совместимость</b>": "--", "Монтаж в шкаф 19\"": "2U", "<b>Особенности корпуса</b>": "--", "Максимальный объем ОЗУ (ГБ)": "512", "Количество отсеков 5,25\"": "0", "Тип оборудования": "Сервер", "<b>Внешние источники информации</b>": "--", "Разъем процессора": "2011-v3", "Видеопамять (МБ)": "0", "Модель": "RD450", "<b>Логистика</b>": "--", "Bluetooth": "Нет", "<b>Питание</b>": "--", "<b>Беспроводной сигнал и приемопередатчики</b>": "--", "Цвета, использованные в оформлении": "Черный", "<b>Прочие характеристики</b>": "--", "Дисковод 1,44 МБ": "Нет", "<b>Программное обеспечение</b>": "--", "Клавиатура": "Нет", "<b>Безопасность</b>": "--"} }
}
```

```
:"-", "<b>Видео</b>": "-", "Размеры (мм)": "482 x 87 x 783 мм", "Частота шины процессора (МГц)": "5000", "<b>Коммуникации</b>": "-", "<b>Поддержка памяти</b>": "-", "Частота ядра (ГГц)": "2,4", "Вес (кг)": "16", "<b>Основные характеристики</b>": "-", "Оптический привод": "Нет", "TV-тюнер": "Нет", "Разъемы компьютера": "USB, RJ45", "Материал корпуса": "Металл, пластик", "<b>Конфигурация</b>": "-", "<b>Интерфейс, разъемы и выходы</b>": "-", "Поддержка Reg / ECC": "ECC / Reg", "GTIN": "00889899057346", "Разъем видеоадаптера": "PCI-Express 16x", "<b>Потребительские свойства</b>": "-", "Количество внешних отсеков 3,5\"/2,5\"": "8", "WiFi": "Нет", "Звук": "Нет"} } }
```

метод goodsDescriptionByCategoryAction

http://services.netlab.ru/rest/catalogsZip/goodsDescriptionByCategory/<categoryId>/<pageNumber>.json?oauth_token=<token>

```
{ } && {"categoryResponse": {"status": {"code": "200", "message": ""}}, "data": {"id": "1307088 [Page 1 of 1]", "name": null, "parentId": null, "leaf": null, "goods": [{"id": 1284948, "properties": {"масса (кг)": "0.1250", "Назначение": "Для планшета", "сайт производителя": "http://redlinegroup.ru/", "гарантия": "1 год", "ширина (см)": "15.5000", "Производитель": "Red Line", "Цвет": "Черный", "длина (см)": "24.5000", "Липучки для фиксации ноутбука": "Нет", "Диагональ ноутбука / Планшета / КПК / GPS / Электронной книги": "7", "Дополнительная информация": "Тонкий и прочный чехол надежно защищает от загрязнения и внешних повреждений."}, "id": 1284948, "id сертификата": 0, "rusName": "[Чехол для планшета] iRidium Чехол - книжка для Samsung Galaxy Tab 3 7.0 [000004033] черный {натуральная кожа}", "инструкция": null, "Тип застежки": "Клипса", "Плечевой ремень": "Нет", "драйвер": null, "manufacturerCountry": "Китай", "высота (см)": "1.5000", "GTIN": null, "pictureID": 131839, "Карманы": "Нет", "описание": "Чехол <b> 4033 </b> от компании Red Line.", "partNumber": "000004033", "Размер главного отделения": "Нет", "Материал": "Кожа", "gtdNumber": "10002010/230216/0045984/2", "название": "[iRidium Чехол - книжка для Samsung Galaxy Tab 3 7.0 [000004033] черный {натуральная кожа}]", "Модель": "4033"} } }
```

метод goodsImagesAction

http://services.netlab.ru/rest/catalogsZip/goodsImages/<goodsId>.json?oauth_token=<token>

```
{ } && {"entityListResponse": {"status": {"code": "200", "message": ""}}, "data": {"items": [{"id": 1, "properties": {"numberOfImages": 1, "Heigth": null, "Size": 6969, "id": "1353192", "Width": null, "Url": "http://www.netlab.ru/ISAPI/TestISAPI19.dll?168436&1&91", "lastUpdateTime": "1467186838343"}]} }}
```

метод goodsImagesByCategoryAction

http://services.netlab.ru/rest/catalogsZip/goodsImagesByCategory/<categoryId>.json?oauth_token=<token>

```
{ } && {"categoryResponse": {"status": {"code": "200", "message": ""}}, "data": {"id": "1193488", "name": null, "parentId": null, "leaf": null, "goods": [{"id": 1131545, "properties": {"id": 1131545, "lastUpdateTime": 1467205513847, "Image3Url": null, "Image1Url": "http://www.netlab.ru/ISAPI/TestISAPI19.dll?48410&1&91", "Image2Url": null, "numberOfImages": 1}, {"id": 1331130, "properties": {"id": 1331130, "lastUpdateTime": 1467205513847, "Image3Url": null, "Image1Url": "http://www.netlab.ru/ISAPI/TestISAPI19.dll?154280&1&91", "Image2Url": null, "numberOfImages": 1}}]}
```

метод goodsByUidAction

http://services.netlab.ru/rest/catalogsZip/goodsByUid/<goodsId>.json?oauth_token=<token>

```
{ } &&
{"goodsResponse": {"status": {"code": "200", "message": ""}, "data": {"id": 1432468, "properties": {"количество на Курской": 0, "цена по категории В": 310.0000, "цена по категории А": 303.1600, "цена по категории D": 296.0000, "удаленный склад": 10, "цена по категории С": 299.0000, "количество на Калужской": 0, "цена по категории F": 290.0000, "цена по категории Е": 293.0000, "id": 1432468, " удален": false, "количество на Лобненской": 12, "количество в транзите": 0.00, "название": "HP 15-ba504ur [X5D88EA] E2 7110/4Gb/500Gb/15.6\"/HD/W10 Blue", "цена по категории N": 316.0000}}}}
```

метод newGoodsAction

http://services.netlab.ru/rest/catalogsZip/newGoods.json?oauth_token=<token>

```
{ } && {"categoryResponse": {"status": {"code": "200", "message": ""}, "data": {"id": "newGoodsId", "name": null, "parentId": null, "leaf": null, "goods": [{"id": 1398519, "properties": {"количество на Курской": 0, "id": 1398519, "categoryName": "Планшеты APPLE", "цена по категории В": 1180.2700, "количество на Лобненской": 0, "цена по категории D": 1135.5500, "количество на Калужской": 0, "цена по категории С": 1140.9500, "цена по категории F": 1124.7500, "цена по категории Е": 1130.1500, "количество в транзите": 0.00, "part#": "MLMV2RU/A", "название": "Apple iPad Pro 9.7-inch Wi-Fi 128GB - Space Gray [MLMV2RU/A]"}, {"id": 1398592, "properties": {"количество на Курской": 0, "id": 1398592, "categoryName": "Планшеты APPLE", "цена по категории В": 1334.9100, "количество на Лобненской": 0, "цена по категории D": 1284.3500, "количество на Калужской": 0, "цена по категории С": 1290.4500, "цена по категории F": 1272.1300, "цена по категории Е": 1278.2400, "количество в транзите": 0.00, "part#": "MLQ62RU/A", "название": "Apple iPad Pro 9.7-inch Wi-Fi + Cellular 256GB - Space Gray [MLQ62RU/A]"}]}}}
```

метод usedGoodsAction

http://services.netlab.ru/rest/catalogsZip/usedGoods.json?oauth_token=<token>

```
{ } && {"categoryResponse": {"status": {"code": "200", "message": ""}, "data": {"id": "usedGoodsId", "name": null, "parentId": null, "leaf": null, "goods": [{"id": 1367003, "properties": {"id": 1367003, "categoryName": "Планшеты APPLE", "цена": "696.83", "количество": 1, "БУ": 0, "название": "Apple iPad mini 4 Wi-Fi + Cellular 64GB - Gold (MK752RU/A)"}, {"id": 1272757, "properties": {"id": 1272757, "categoryName": "Планшеты TURBO", "цена": "28.30", "количество": 1, "БУ": 1, "название": "Планшетный компьютер \"TurboPad 1013\" 3G/10.1' IPS/1280x800/16Gb/1024Mb/0.3x2.0Mpx/Android 4.2"}, {"id": 1296640, "properties": {"id": 1296640, "categoryName": "ASUS Серверные материнские платы", "цена": "55.93", "количество": 2, "БУ": 1, "название": "Asus Материнская плата P9D-E / 4L (RTL) { LGA1150 PCI-E+SVGA+4xGbLAN SATA RAID ATX 4DDR-III }"}, {"id": 1337924, "properties": {"id": 1337924, "categoryName": "Интерактивные доски, дисплеи SMART", "цена": "140.74", "количество": 1, "БУ": 1, "название": "SMART Board SB480 Интерактивная доска 3 77\" (195.6 см), 4:3 [SB480]"}, {"id": 196557, "properties": {"id": 196557, "categoryName": "Бумага широкофотоматная Xerox", "цена": "12.50", "количество": 1, "БУ": 0, "название": "XEROX 003R93238 Бумага в рулонах 175м XEROX A1, 594мм, 75г"}]}}}
```

метод goodsByPartnumber

http://services.netlab.ru/rest/catalogsZip/goodsByPartnumber/{partnumber}.json?oauth_token=<token>

```
{"goodsResponse": {"status": {"code": "200", "message": ""}, "data": {"id": 1432468, "properties": {"количество на Курской": 0, "цена по категории В": 310.0000, "цена по категории А": 303.1600, "цена по категории D": 296.0000, "удаленный склад": 10, "цена по категории С": 299.0000, "количество на Калужской": 0, "цена по категории F": 290.0000, "цена по категории Е": 293.0000, "id": 1432468, " удален": false, "количество на Лобненской": 12, "количество в транзите": 0.00, "название": "HP 15-ba504ur [X5D88EA] E2 7110/4Gb/500Gb/15.6\"/HD/W10 Blue", "цена по категории N": 316.0000}}}}
```

метод getDateByStickerAction

http://services.netlab.ru/rest/catalogsZip/dateBySticker.json?oauth_token=<token>&sticker=<stickerNumber>

```
{ } && {"entityListResponse": {"status": {"code": "200", "message": ""}, "data": {"items": []}}}
```

метод getDateBySerialNumberAction

http://services.netlab.ru/rest/catalogsZip/dateBySerialNumber.json?oauth_token=<token>&serialnumber=<serialNumber>

```
{ } && {"entityListResponse": {"status": {"code": "200", "message": ""}, "data": {"items": []}}}
```

метод getInfoAction

http://services.netlab.ru/rest/catalogsZip/info.json?oauth_token=<token>

```
{ } && {"entityListResponse": {"status": {"code": "200", "message": ""}, "data": {"items": [{"id": 1, "properties": {"usdRateNonCash": 65.594900, "usdRateCash": 64.321200, "id": 1}}]}}}
```

метод getGoodsSearch

http://services.netlab.ru/rest/catalogsZip/getGoodsSearch/{descr}.json?oauth_token=<token>

```
{ } && {"categoryResponse": {"status": {"code": "200", "message": "Search string: apple:iphone ; Quantity goods: 475"}, "data": {"id": null, "name": null, "parentId": null, "leaf": null, "goods": [{"id": 423633, "properties": {"Deleted": false, "OutOfProduction": false, "Transit": 0.0, "remoteStock": null, "Descr": "Чехол (флип-кейс) Armor-X для Apple iPhone 6 flip full синий (962278)", "PriceN": 6.8600, "RRC": 6.8600, "Kurskaya": 0, "PriceB": 4.4900, "PriceD": 4.1100, "Kaluzhskaya": 0, "PriceC": 4.2000, "PriceF": 4.110000, "PriceE": 4.0100, "UID": 1423633, "Guarantee": "1 год", "ID": 423633, "Lobnenskaya": 0}]}]}
```

3. МОДУЛЬ «ДОКУМЕНТЫ»

метод getOrganizationsAction

http://services.netlab.ru/rest/documents/organizations.json?oauth_token=<token>

```
{ } && {"entityListResponse": {"status": {"code": "200", "message": ""}, "data": {"items": [{"id": 101076, "properties": {"id": 101076, "название": "Ваша компания"}]}]}}
```

метод getDocumentTypesAction

http://services.netlab.ru/rest/documents/documentTypes.json?oauth_token=<token>

```
{ } && {"entityListResponse": {"status": {"code": "200", "message": ""}, "data": {"items": [{"id": 0, "properties": {"id": 0, "name": "Резерв"}}, {"id": 1, "properties": {"id": 1, "name": "Пр оплата"}}, {"id": 2, "properties": {"id": 2, "name": "Счет"}}, {"id": 3, "properties": {"id": 3, "на ме": "Отгрузка"}}, {"id": 4, "properties": {"id": 4, "name": "Ремонт"}}, {"id": 5, "properties": {"id": 5, "name": "Bce"}]}]}}
```

метод listDocuments

http://services.netlab.ru/rest/documents/list.json?oauth_token=<token>&documentTypeId=<typeId>&beginDate=<beginInterval>&endDate=<endInterval>

```
{ } && {"documentsResponse": {"status": {"code": "200", "message": null}, "data": {"documents": [{"id": "0513213695Ц", "properties": {"id": "0513213695Ц", "old": 0, "weight": 1.6100, "organizationName": "Наша компания", "summa": 76.5200, "invoiceID": "0", "name": "Резервирование товара", "volume": 0.0106, "priceReserved": false, "parentDocId": null, "comment": "", "date": "19.07.2016"}, "goods": []}, {"id": "0513563696Ц", "properties": {"id": "0513563696Ц", "old": 0, "weight": 0.1200, "organizationName": "Наша компания", "summa": 17.2900, "invoiceID": "0", "name": "Резервирование товара", "volume": 0.0008, "priceReserved": false, "parentDocId": null, "comment": "", "date": "19.07.2016"}, "goods": []}]}}}
```

метод readAction

http://services.netlab.ru/rest/documents/get.json?oauth_token=<token>&documentId=<Id>

```
{ } && {"documentResponse": {"status": {"code": "200", "message": null}, "data": {"id": "0514253047Ц", "properties": {"id": "0514253047Ц", "weight": 0.0000, "organizationName": "Моя Компания", "volume": 0.0000, "priceReserved": 0, "comment": "testAPI", "date": "20170615"}, "goods": [{"id": 1305782, "properties": {"weight": 0.0000, "summa": 15.3800, "summaCur": 1.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "2 года", "country": "Китай", "id": 1305782, "ChangePrice": false, "category": "", "order": 1.0000, "name": "Коврик для мыши Mad Catz G.L.I.D.E.3 Gaming Surface (300x220) пористая резина (MCB4381100A1/06/1) [PCAI40]", "transitDate": "", "volume": 0.0000, "usdRate": 58.2854, "quantity": 0.00}}, {"id": 1305784, "properties": {"weight": 0.0000, "summa": 22.3200, "summaCur": 1.0000, "rowId": 2, "orderStatus": "Ожидание ответа", "garantia": "2 года", "country": "Китай", "id": 1305784, "ChangePrice": false, "category": "", "order": 1.0000, "name": "Коврик для мыши Mad Catz G.L.I.D.E.5 Gaming Surface (400x500) пористая резина (MCB4380900A1/06/1)"}, {"id": 1348257, "properties": {"weight": 0.0000, "summa": 37.9000, "summaCur": 1.0000, "rowId": 3, "orderStatus": "Уточнить 1 р.день", "garantia": "2 года", "country": "Китай", "id": 1348257, "ChangePrice": false, "category": "", "order": 1.0000, "name": "Коврик для мыши Mad Catz Premium Hybrid Mouse Pad (MCB4380600A1/12/4) [PCAmc1]", "transitDate": "", "volume": 0.0000, "usdRate": 58.2854, "quantity": 0.00}}]}}}
```

метод createAction

http://services.netlab.ru/rest/documents/new.json?oauth_token=<token>&documentTypeId=<documentType>&organizationId=<org>&date=<date>&comment=<comment>&goodsString=<goods>

```
{ } && {"documentResponse": {"status": {"code": "200", "message": "Резерв на 14.06.2017 принят/изменен, сумма резерва: 0.00 у.е"}, "data": {"id": "0514253052Ц", "properties": {"id": "0514253052Ц", "weight": 0.0000, "organizationName": "Наша Компания", "volume": 0.0000, "priceReserved": 0, "comment": "test", "date": "20170614"}, "goods": [{"id": 1467086, "properties": {"weight": 0.0000, "summa": 4.4400, "summaCur": 2.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "5 лет", "country": "", "id": 1467086, "ChangePrice": false, "category": "", "order": 2.0000, "name": "CBR CMP 808 Armor, черный Игровой коврик для мыши"}, {"id": 1271487, "properties": {"weight": 0.0000, "summa": 1.5400, "summaCur": 2.0000, "rowId": 2, "orderStatus": "Уточнить 1 р.день", "garantia": "3 месяца", "country": "Китай", "id": 1271487, "ChangePrice": false, "category": "", "order": 2.0000, "name": "KREOLZ Tanks, Коврик для мыши, 270*230 мм"}, {"id": 1271488, "properties": {"weight": 0.0000, "summa": 1.5400, "summaCur": 2.0000, "rowId": 3, "orderStatus": "Уточнить 1 р.день", "garantia": "3 месяца", "country": "Китай", "id": 1271488, "ChangePrice": false, "category": "", "order": 2.0000, "name": "KREOLZ Tanks, Коврик для мыши, 270*230 мм"}]}]}
```

метод updateAction

http://services.netlab.ru/rest/documents/update.json?oauth_token=<token>&documentTypeId=<documentType>&documentId=<docId>&organizationId=<org>&date=<date>&comment=<comment>&goodsString=<goods>

```
{} && {"documentResponse": {"status": {"code": "200", "message": "Резерв на 21.07.2016 изменен, сумма изменения: 0.00 у.е"}, "data": {"id": "0513563695Ц", "properties": {"id": "0513563695Ц", "weight": 0.0000, "organizationName": "Наша компания", "volume": 0.0000, "priceReserved": 0, "comment": "comment", "date": "20160721"}, "goods": [{"id": 1232845, "properties": {"weight": 0.0000, "summa": 1544.0000, "summaCur": 1.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "2 года", "country": "Россия", "id": 1232845, "ChangePrice": false, "category": "", "order": 1.0000, "name": "C207416Ц-i7-2600 / 2 x 8GB / 2TB / DVDRW", "transitDate": "", "volume": 0.0000, "usdRate": 65.0089, "quantity": 0.00}}]}}}
```

метод changeAction

http://services.netlab.ru/rest/documents/change.json?oauth_token=<token>&documentTypeId=<documentType>&documentId=<docId>&goodsString=<goodsChanges>

```
{} && {"documentResponse": {"status": {"code": "200", "message": "Резерв на 21.07.2016 изменен, сумма изменения: 0.00 у.е"}, "data": {"id": "0513563695Ц", "properties": {"id": "0513563695Ц", "weight": 0.0000, "organizationName": "Наша компания", "volume": 0.0000, "priceReserved": 0, "comment": "", "date": "20160721"}, "goods": [{"id": 1232845, "properties": {"weight": 0.0000, "summa": 1544.0000, "summaCur": 2.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "2 года", "country": "Россия", "id": 1232845, "ChangePrice": false, "category": "", "order": 2.0000, "name": "C207416Ц-i7-2600 / 2 x 8GB / 2TB / DVDRW", "transitDate": "", "volume": 0.0000, "usdRate": 65.0089, "quantity": 0.00}}]}}}
```

метод deleteAction

http://services.netlab.ru/rest/documents/delete.json?oauth_token=<token>&documentTypeId=<documentType>&documentId=<docId>

```
{} && {"documentResponse": {"status": {"code": "200", "message": "Резерв удален"}, "data": null}}
```

метод createReserve

http://services.netlab.ru/rest/documents/newReserve.json?oauth_token=<token>&organizationId=<organization>&date=<date>&comment=<comment>&goodsString=<goods>&reservePrice=<reservePrice>

```
{} && {"documentResponse": {"status": {"code": "200", "message": "Резерв на 21.07.2016 принят/изменен, сумма резерва: 0.00 у.е"}, "data": {"id": "0513568318Ц", "properties": {"id": "0513568318Ц", "weight": 0.0000, "organizationName": "Наша компания", "volume": 0.0000, "priceReserved": 1, "comment": "comment", "date": "20160721"}, "goods": [{"id": 1232845, "properties": {"weight": 0.0000, "summa": 1544.0000, "summaCur": 1.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "2 года", "country": "Россия", "id": 1232845, "ChangePrice": false, "category": "", "order": 1.0000, "name": "C207416Ц-i7-2600 / 2 x 8GB / 2TB / DVDRW", "transitDate": "", "volume": 0.0000, "usdRate": 65.0089, "quantity": 0.00}}]}}}
```

метод repairGoodsList

http://services.netlab.ru/rest/documents/repairGoodsList.json?oauth_token=<token>

```
{} && {"entityListResponse": {"status": {"code": "200", "message": ""}, "data": {"items": []}}}}
```

метод readShipment

http://services.netlab.ru/rest/documents/readShipment.json?oauth_token=<token>&documentId=<documentId>

```
{} && {"documentResponse": {"status": {"code": "200", "message": null}, "data": {}}
```

```
{"id":"048749525Ц","properties":{"id":"048749525Ц","weight":62.8800,"organizationName":"Наша компания","volume":0.1900,"priceReserved":0,"comment":null,"date":"20160718"}, "goods":[{"id":1171057,"properties":{"Software":0,"weight":62.8800,"summa":126.0000,"summaCur":8251.1500,"rowId":1,"garantia":"2 года","country":"Филиппины","ChangePrice":false,"id":1171057,"category":"","order":8251.1500,"name":"[ИБП] APC Back-UPS ES 700VA BE700G-RS","transitDate":"","volume":0.1900,"usdRate":65.4846,"quantity":8.00,"gtd":"10130120/290616/0006036/3"}]}]}
```

метод splitReserve

http://services.netlab.ru/rest/documents/splitReserve.json?oauth_token=<token>&documentId=<documentId>&goodsString=<goodsId>

```
{ } && {"documentsResponse": {"status": {"code": "6", "message": "0513541156Ц"}, "data": {"documents": [{"id": "0513541155Ц", "properties": {"date": "20160721", "volume": 0.0000, "organizationName": "Наша компания", "weight": 0.0000, "comment": "WS_Created", "priceReserved": 0, "id": "0513541155Ц"}, "goods": [{"id": 1232845, "properties": {"transitDate": "", "country": "Россия", "ChangePrice": false, "quantity": 0.00, "weight": 0.0000, "summaCur": 1.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "2 года", "volume": 0.0000, "summa": 1544.0000, "usdRate": 65.5949, "name": "C207416Ц-i7-2600 / 2 x 8GB / 2TB / DVDRW", "id": 1232845, "category": "", "order": 1.0000}], {"id": "0513541156Ц", "properties": {"date": "20160719", "volume": 0.0000, "organizationName": "Наша компания", "weight": 0.1200, "comment": "Часть резерва №0513541155Ц", "priceReserved": 0, "id": "0513541156Ц"}, "goods": [{"id": 1400085, "properties": {"transitDate": "", "country": "КИТАЙ", "ChangePrice": false, "quantity": 1.00, "weight": 0.1200, "summaCur": 0.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "Б.", "volume": 0.0000, "summa": 4.1100, "usdRate": 65.5949, "name": "\\"Щетка ЗУБР \\"ЭКСПЕРТ\\\" дисковая для дрели, полимерно-абразивная, 100мм [35161-100_z01]", "id": 1400085, "category": "", "order": 1.0000}, {"id": 1400093, "properties": {"transitDate": "", "country": "", "ChangePrice": false, "quantity": 0.00, "weight": 0.0000, "summaCur": 1.0000, "rowId": 2, "orderStatus": "Уточнить 1 р.день", "garantia": "Б.", "volume": 0.0000, "summa": 4.9000, "usdRate": 65.5949, "name": "\\"Щетка ЗУБР \\"ЭКСПЕРТ\\\" дисковая \\"звезда\\" для дрели, полимерно-абразивная, зерно P120, 100мм", "id": 1400093, "category": "", "order": 1.0000}]}]}]}}}
```

метод mergeReserves

http://services.netlab.ru/rest/documents/mergeReserves.json?oauth_token=<token>&documentId=<documentId>&document2Id=<document2Id>

```
{ } && {"documentResponse": {"status": {"code": "200", "message": "Операция успешно завершена."}, "data": null}}
```

метод prolongReserve

http://services.netlab.ru/rest/documents/prolongReserve.xml?oauth_token=<token>&documentId=<documentId>&date=<date>

```
{ } && {"documentResponse": {"status": {"code": "200", "message": "Резерв на 05.10.2016 изменен, сумма изменения: 0.00"}, "data": {"id": "0513593183Ц", "properties": {"date": "20161005", "volume": 0.0000, "organizationName": "ВашаКомпания", "weight": 0.0000, "comment": "", "priceReserved": 0, "id": "0513593183Ц"}, "goods": [{"id": 1324893, "properties": {"transitDate": "", "country": "Тайвань (Китай)", "ChangePrice": false, "quantity": 0.00, "weight": 0.0000, "summaCur": 1.0000, "rowId": 1, "orderStatus": "Уточнить 1 р.день", "garantia": "3 года", "volume": 0.0000, "summa": 35.8000, "usdRate": 67.9340, "name": "Corsair USB Drive 64Gb Voyager GT CMFVYGT3B-64GB {USB3.0}", "id": 1324893, "category": "", "order": 1.0000}]}]}
```

метод orderGoods

http://services.netlab.ru/rest/documents/orderGoods.json?oauth_token=<token>&documentId=<documentId>&goodsUid=<goodsUid>

```
{ } && {"documentResponse": {"status": {"code": "200", "message": "Операция успешно завершена."}, "data": {"id": "0514253040Ц", "properties": {"id": "0514253040Ц", "weight": 0.0000, "organizationName": "Наша Компания", "volume": 0.0000, "priceReserved": 0, "comment": "", "comment", "date": "20170613"}, "goods": [{"id": 1467086, "properties": {"weight": 0.0000, "summa": 4.4400, "summaCur": 2.0000, "rowId": 1, "orderStatus": "Ожидание ответа", "garantia": "5 лет", "country": "", "id": 1467086, "ChangePrice": false, "category": "", "order": 2.0000, "name": "CBR CMP 808 Armor, черный Игровой коврик для мыши", "transitDate": "Отправлен запрос 2017-06-13 11:35", "volume": 0.0000, "usdRate": 58.2854, "quantity": 0.00}}, {"id": 1271487, "properties": {"weight": 0.0000, "summa": 1.5400, "summaCur": 2.0000, "rowId": 2, "orderStatus": "Ожидание ответа", "garantia": "3 месяца", "country": "Китай", "id": 1271487, "ChangePrice": false, "category": "", "order": 2.0000, "name": "KREOLZ PAN-10 Tanks, Коврик для мыши, тканево-резиновый, 270*230 мм", "transitDate": "Отправлен запрос 2017-06-13 12:11", "volume": 0.0000, "usdRate": 58.2854, "quantity": 0.00}}]}}}
```

4. МОДУЛЬ «ДОСТАВКА»

метод listDocuments

http://services.netlab.ru/rest/delivery/list.json?oauth_token=<token>&beginDate=<beginInterval>&endDate=<endInterval>

```
{ } &&
{"documentsResponse": {"status": {"code": "200", "message": ""}, "data": {"documents": [{"id": "101076", "properties": {"DocumentIds": "048731945Ц, 048731947Ц, 048731948Ц", "address": "МОСКВА, ул. Ленина д.13", "deliveryType": 0, "oplType": "", "weight": "147.6000", "rateClientId": "26:\u0001:71879", "rateId": "", "packId": 977940, "volume": "0.8100", "rateType": "", "boxCount": "15.0000", "stateText": "Доставлен", "price": "", "absent": 0, "id": 101076, "state": 2, "deliveryDate": "20160210", "tkDescr": "", "oplDidExt": "", "priceCurr": "$"}, "goods": []}, {"id": "101076", "properties": {"DocumentIds": "048742308Ц, 048742313Ц, 048742314Ц, 048742317Ц", "address": "МОСКВА, ул.Молодогвардейская д 21 кор 1", "deliveryType": 0, "oplType": "", "weight": "200.5600", "rateClientId": "26:\u0001:75674", "rateId": "", "packId": 1001880, "volume": "1.6000", "rateType": "", "boxCount": "8.0000", "stateText": "Доставлен", "price": "", "absent": 0, "id": 101076, "state": 2, "deliveryDate": "20160516", "tkDescr": "", "oplDidExt": "", "priceCurr": "$"}, "goods": []}]}}}
```

метод availableDocuments

http://services.netlab.ru/rest/delivery/docs.json?oauth_token=<token>&beginDate=<beginInterval>&endDate=<endInterval>

```
{ } &&
{"documentsResponse": {"status": {"code": "200", "message": ""}, "data": {"documents": []}}}
```

метод details

http://services.netlab.ru/rest/delivery/details.json?oauth_token=<token>&packId=<packId>

```
{ } && {"deliveryDocumentResponse": {"status": {"code": "200", "message": ""}, "data": {"id": null, "properties": {"packId": "977940"}, "goods": [{"id": 1381731, "properties": {"quantity": 1, "name": "C346983Ц-Intel / Z97M-PLUS RTL / 2 x 4GB / 500Gb / DVDRW / Win Pro 7 Russian", "box": "Box"}, {"id": 1381731}], "id": 1381731, "properties": {"quantity": 1, "name": "C346983Ц-Intel / Z97M-PLUS RTL / 2 x 4GB / 500Gb / DVDRW / Win Pro 7 Russian", "box": "Box"}]}}
```

```

№2","id":1381731}}, {"id":1381731,"properties":{"quantity":1,"name":"C346983Ц-Intel /  
Z97M-PLUS RTL / 2 x 4GB / 500Gb / DVDRW / Win Pro 7 Russian ","box":"Box  
№3","id":1381731}}, {"id":1381731,"properties":{"quantity":1,"name":"C346983Ц-Intel /  
Z97M-PLUS RTL / 2 x 4GB / 500Gb / DVDRW / Win Pro 7 Russian ","box":"Box  
№4","id":1381731}}, {"id":1387011,"properties":{"quantity":1,"name":"C351399Ц-Intel Core  
i7-4770 / Z87-A RTL / 2x8GB / 120Gb / 1TB / Win Pro 7 Russian ","box":"Box  
№5","id":1387011}}],"absentGoods":[]}}}

```

метод dates

http://services.netlab.ru/rest/delivery/dates.json?oauth_token=<token>&beginDate=<beginInterval>&endDate=<endInterval>

```
{ } && {"entityListResponse":{"status":{"code":"200","message":""}, "data":{"items":[]}}}
```

метод deliveryTypes

http://services.netlab.ru/rest/delivery/deliveryTypes.json?oauth_token=<token>

```
{ } && {"entityListResponse":{"status":{"code":"200","message":""}, "data":{"items":[{"id":0,"properties":{"name":"Доставка","id":0}}]}}}
```

метод paymentTypes

http://services.netlab.ru/rest/delivery/paymentTypes.json?oauth_token=<token>

```
{ } && {"entityListResponse":{"status":{"code":"200","message":""}, "data":{"items":[{"id":0,"properties":{"name":"Позиция в счете","id":0}}, {"id":1,"properties":{"name":"Новый счет","id":1}}]}}}
```

метод rates

http://services.netlab.ru/rest/delivery/rates.json?oauth_token=<token>&documentsIds=<documentIds>&deliveryDate=<deliveryDate>&deliveryType=<deliveryType>

```
{ } && {"entityListResponse":{"status":{"code":"23","message":"Ошибка при загрузке  
данных (список документов)!"}, "data":null}}
```

5. ЧТО НОВОГО? ИЗМЕНЕНИЯ В ДОКУМЕНТАЦИИ

Версия 1.32 от 03 июля 2019 года

- Добавлено новое поле РРЦ для categoryAction.

Версия 1.31 от 30 января 2019 года

- Устранены мелкие недочеты

Версия 1.30 от 14 сентября 2018 года

- Добавлено свойство «Производитель» в методе **categoryAction**

Версия 1.29 от 03 мая 2018 года

- Удален метод **allExistingGoods** в модуле «Каталог»

Версия 1.28 от 22 декабря 2017 года

- Добавлен метод **getGoodsSearch** в модуле «Каталог»

Версия 1.27 от 24 августа 2017 года

- Добавлены примеры кода на PHP и JavaScript методов: **orderGoods**, **prolongReserve**, **goodsDescriptionByCategoryAction**, **goodsImagesByCategoryAction**.

Версия 1.26 от 07 июля 2017 года

- Добавлено свойство «Удаленный склад» в методах **categoryAction**, **goodsByUidAction** и **goodsByPartnumber**.

Версия 1.25 от 15 июня 2017 года

- Добавлено свойство «orderStatus» в методах работы с резервами.
- Добавлен метод **orderGoods**, позволяющий уточнять информацию по отсутствующим товарам и заказывать их.

Версия 1.24 от 05 июня 2017 года

- Добавлен метод **goodsImagesByCategoryAction**, возвращающий изображения товаров по категориям

Версия 1.23 от 24 апреля 2017 года

- Добавлено свойство GTIN в методах **goodsAction** и **goodsDescriptionByUidAction**
- Добавлен метод **goodsDescriptionByCategoryAction**, возвращающий описания товаров по категориям

Версия 1.22 от 07 апреля 2017 года

- Добавлено более детальное описание для модуля «Аутентификация»

Версия 1.21 от 22 марта 2017 года

- Добавлены свойства «вес» и «объём» в методе **categoryAction** модуля «Каталог»

Версия 1.20 от 02 марта 2017 года

- Добавлены свойства «НДС» и «гарантия» в методе **categoryAction** модуля «Каталог»

