



# Description of web-services

[Version 1.1](#)

## Contents

<b>Introduction</b> .....	3
<b>Methods used to receive data</b> .....	4
ListCountries .....	5
ListCities .....	9
ListZips .....	13
ListPoints.....	17
GetStatuses .....	22
DeliveryCalc.....	26
PUPDetails .....	30
ZipCheck .....	34
<b>Methods used to transfer data</b> .....	37
CreateParcel.....	38
DeleteParcel.....	46
UpdateParcel .....	50
ManifestControl.....	59
WarehouseControl .....	62
ControlUser .....	67
<b>List of changes</b> .....	70

## Introduction

Boxberry delivery service provides solution for integrating its information system with information systems of online stores/partners. Such integration allows us to exchange data on-line excluding manual labor.

The interaction with our web-services is performed by using HTTP and HTTPS - standard protocols - and SOAP and JSON protocols, ensuring the simplicity of their use and fast access to the main functions of the Boxberry information system.

You can receive relevant information about the countries and locations of delivery, pick-up points and lockers, events of the parcel, availability of courier delivery service by the receiver's ZIP code, calculate the cost of parcel and information about creating parcels, updating data on a previously created and deleted parcel, creating manifests and pre-alert, creating / editing users and warehouses in Boxberry system.

Production web-interface <a href="https://go.bxb.delivery/login/">https://go.bxb.delivery/login/</a>	Location of web-services JSON: <a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a> SOAP: <a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a> <a href="https://go.bxb.delivery/_soap/1c_lc.php">https://go.bxb.delivery/_soap/1c_lc.php</a>
---	---

Sandbox
<a href="https://test.bxb.delivery/login/">https://test.bxb.delivery/login/</a>

## Methods used to receive data

JSON Url: <a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a>
SOAP Url: <a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a>

*Datatable 1. JSON, SOAP protocol available methods*

Num	Method name	Description
1.	<b>ListCountries</b>	method allows to obtain the list of countries with Boxberry delivery. Read more at page 5.
2.	<b>ListCities</b>	method allows to get a list of settlements in which Boxberry delivery is possible. By default returns all cities in all countries of delivery. Read more at page 9.
3.	<b>ListZips</b>	method allows you to get all the zipcodes of countries in which Boxberry delivery is available. Read more at page 13.
4.	<b>ListPoints</b>	method allows you to get a list of Boxberry pick-up points and lockers. By default, returns all points in all countries. Read more at page 17.
5.	<b>GetStatuses</b>	method allows you to get a list of events for the parcel. Read more at page 22.
6.	<b>DeliveryCalc</b>	method allows you to calculate the cost of parcel. Read more at page 26.
7.	<b>PUPDetails</b>	method allows you to get additional information on the point, including photos. Read more at page 30.
8.	<b>ZipCheck</b>	method provides information on the availability of courier delivery. Read more at page 34.

## ListCountries

Description of API method parameters (for JSON):

*Datatable 2. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	ListCountries

*Datatable 3. Output parameters*

Parameter	Type	Required	Description
result	array	yes	Result
error	array	yes	Errors

*Datatable 4. Parameters in result*

Parameter	Type	Required	Description
Code	integer	yes	Country code
NameRu	string	yes	Name in Russian
NameEng	string	yes	Name in English

*Datatable 5. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [ListCountries](#).

*Datatable 6. Method ListCountries*

<b>JSON</b>	Url: <a href="https://go.bxb.delivery/json.php?token=&amp;method=ListCountries">https://go.bxb.delivery/json.php?token=&amp;method=ListCountries</a>
<b>Request method «GET»</b>	<p><b>Request example:</b> GET token= &amp;method=ListCountries</p> <p>RAW body { "token": "", "method": "ListCountries" }</p>
<b>Response example:</b>	<pre>{ "result": [   {     "Code": "",     "NameRu": "Россия",     "NameEng": "Russia"   },   {     "Code": "",     "NameRu": "Украина",     "NameEng": "Ukraine"   } ], "error": {   "isError": false } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$url = 'https://go.bxb.delivery/json.php?token=;method=ListCountries'; \$handle = fopen(\$url, 'rb'); \$content = stream_get_contents(\$handle); fclose(\$handle); if (!\$content) {   echo 'No response received'; } \$data = json_decode(\$content, true); if (empty(\$data)) { echo 'Error decode response'; } if (!empty(\$data['errors']['isError'])) {   //A request is made with an error   echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {   //Everything is fine, the answer is received, data in \$data:   /*   print_r(\$data);   [ 'result' =&gt;   [   [ 'Code' =&gt; "", 'NameRu' =&gt; 'Россия',   'NameEng' =&gt; 'Russia'   ],   [ 'Code' =&gt; "",   'NameRu' =&gt; 'Украина',   'NameEng' =&gt; 'Ukraine'   ]   ]   */ </pre>

	<pre> ] ], 'error' =&gt; [ 'isError' =&gt; false ] ] */ } </pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a>
<b>SOAP Request</b>	<pre> &lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_public.php" &lt;soapenv:Header/&gt; &lt;soapenv:Body&gt; &lt;c:ListCountries soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;ListCountriesQuery xsi:type="c:ListCountriesQuery"&gt; &lt;token xsi:type="xsd:string"&gt;&lt;/token&gt; &lt;/ListCountriesQuery&gt; &lt;/c:ListCountries&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<b>SOAP Response</b>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://my.boxberry.ru/_soap/1c_public.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;SOAP-ENV:Body&gt; &lt;ns1:ListCountriesResponse&gt; &lt;return xsi:type="ns1:ArrayOfListCountriesData"&gt; &lt;item xsi:type="ns1:ListCitiesData"&gt; &lt;NameRu xsi:type="xsd:string"&gt;Россия&lt;/NameRu&gt; &lt;NameEng xsi:type="xsd:string"&gt;Russia&lt;/NameEng&gt; &lt;code xsi:type="xsd:string"&gt; &lt;/code&gt; &lt;/item&gt; &lt;item xsi:type="ns1:ListCitiesData"&gt; &lt;NameRu xsi:type="xsd:string"&gt;Украина&lt;/NameRu&gt; &lt;NameEng xsi:type="xsd:string"&gt;Ukraine&lt;/NameEng&gt; &lt;code xsi:type="xsd:string"&gt; &lt;/code&gt; &lt;/item&gt; &lt;/return&gt; &lt;/ns1:ListCountriesResponse&gt; &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<b>Example for PHP 5.0 or later version</b>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/_soap/1c_public.php', [ 'cache_wsdl' =&gt; WSDL_CACHE_NONE, 'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]); \$request = [ 'token' =&gt; "", ]; try { \$result = \$client-&gt;ListCountries(\$request); } </pre>

```
catch (\SoapFault $e)
{
//Request execution error
}
if (isset($e))
{ print_r($e);
}
else {
print_r($result);
/* [ 'result' => [
[ 'Code' => ",
'NameRu' => 'Россия',
'NameEng' => 'Russia' ],
[ 'Code' => ",
'NameRu' => 'Украина',
'NameEng' => 'Ukraine' ]
],
'error' => [ 'isError' => false
]
]
*/ }
```



## ListCities

Description of API method parameters (for JSON):

*Datatable 7. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	ListCities
countryCode	integer	no	Country code (ISO 3166-1 numeric)

*Datatable 8. Output parameters*

Parameter	Type	Required	Description
result	array	yes	Result
error	array	yes	Errors

*Datatable 9. Parameters in result*

Parameter	Type	Required	Description
Code	integer	yes	City code
NameRu	string	yes	Name in Russian
NameEng	string	yes	Name in English

*Datatable 10. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [ListCities](#).

*Datatable 11. Method ListCities*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php?token= &amp;method=ListCities">https://go.bxb.delivery/json.php?token= &amp;method=ListCities</a>
<b>Request method «GET»</b>	<pre>GET token= &amp;method=ListCities RAW body { "token": "", "method": "ListCities" }</pre>
<b>Response example:</b>	<pre>{ "result": [ { "Code": "", "NameRu": "Абакан", "NameEng": "Abakan" }, { "Code": "", "NameRu": "Абинск", "NameEng": "Abinsk" } ], "error": { "isError": false } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$url = 'https://go.bxb.delivery/json.php?token=;method=ListCities'; \$handle = fopen(\$url, 'rb'); \$content = stream_get_contents(\$handle); fclose(\$handle); if (!\$content) { Echo 'No response received'; } \$data = json_decode(\$content, true); if (empty(\$data)) { echo 'Error decode response'; } if (!empty(\$data['errors']['isError']) ) ) { //A request is made with an error echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else { //Everything is fine, the answer is received, data in \$data: /* print_r(\$data); [ 'result' =&gt; [ [ 'Code' =&gt; "", 'NameRu' =&gt; 'Абакан', 'NameEng' =&gt; 'Abakan'</pre>

	<pre> ], [ 'Code' =&gt; ', 'NameRu' =&gt; 'Абинск', 'NameEng' =&gt; 'Abinsk' ] ], 'error' =&gt; [ 'isError' =&gt; false ] ] */ } </pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/__soap/1c_public.php">https://go.bxb.delivery/__soap/1c_public.php</a>
<b>SOAP Request</b>	<pre> &lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/__soap/1c_public.php"&gt; &lt;soapenv:Header/&gt; &lt;soapenv:Body&gt; &lt;c&gt;ListCities soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;ListCitiesQuery xsi:type="c&gt;ListCitiesQuery"&gt; &lt;token xsi:type="xsd:string"&gt; &lt;/token&gt; &lt;countryCode xsi:type="xsd:string"&gt;643 &lt;/countryCode&gt; &lt;/ListCitiesQuery&gt; &lt;/c&gt;ListCities&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<b>SOAP Response</b>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/__soap/1c_public.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;SOAP- ENV:Body&gt; &lt;ns1&gt;ListCitiesResponse&gt; &lt;return xsi:type="ns1:ArrayOfListCitiesData"&gt; &lt;item xsi:type="ns1&gt;ListCitiesData"&gt; &lt;NameRu xsi:type="xsd:string"&gt;Москва&lt;/NameRu&gt; &lt;NameEng xsi:type="xsd:string"&gt;Moscow&lt;/NameEng&gt; &lt;code xsi:type="xsd:string"&gt; &lt;/code&gt; &lt;Prefix xsi:nil="true"/&gt; &lt;UniqueName xsi:nil="true"/&gt; &lt;Pup xsi:nil="true"/&gt; &lt;/item&gt; &lt;item xsi:type="ns1&gt;ListCitiesData"&gt; &lt;NameRu xsi:type="xsd:string"&gt;Абакан&lt;/NameRu&gt; &lt;NameEng xsi:type="xsd:string"&gt;Abakan&lt;/NameEng&gt; &lt;code xsi:type="xsd:string"&gt; &lt;/code&gt; &lt;Prefix xsi:nil="true"/&gt; &lt;UniqueName xsi:nil="true"/&gt; &lt;Pup xsi:nil="true"/&gt; &lt;/item&gt; &lt;/return&gt; &lt;/ns1&gt;ListCitiesResponse&gt; &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>

*Example for PHP 5.0 or later version*

```
<?php
$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_public.php', [
    'cache_wsdl' => WSDL_CACHE_NONE,
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS,
]);

$request = [
    'token' => "",
];

try {
    $result = $client->ListCities($request);
} catch (\SoapFault $e) {
    //Request execution error
}

if (isset($e)) {
    print_r($e);
} else {
    print_r($result);
    /*
    [
        'result' => [
            [
                'Code' => "",
                'NameRu' => 'Абакан',
                'NameEng' => 'Abakan'
            ],
            [
                'Code' => "",
                'NameRu' => 'Абинск',
                'NameEng' => 'Abinsk'
            ]
        ],
        'error' => [
            'isError' => false
        ]
    ]
    */
}
}
```

## ListZips

Description of API method parameters (for JSON):

*Datatable 12. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	ListZips
countryCode	integer	no	Country code (ISO 3166-1 numeric)
cityCode	integer	no	City code

*Datatable 13. Output parameters*

Parameter	Type	Required	Description
result	array	yes	Massive postcodes
error	array	yes	Errors

*Datatable 14. Parameters in result*

Parameter	Type	Required	Description
id	string	yes	Postcode
CountryCode	число	yes	Country code (ISO 3166-1 numeric)
CityRU	string	yes	Name in Russian
CityEng	string	yes	Name in English
TariffZone	integer	yes	Tariff zone

*Datatable 15. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [ListZips](#).

*Datatable 16. Method ListZips*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php?token=&amp;method=ListZips">https://go.bxb.delivery/json.php?token=&amp;method=ListZips</a>
<b>Request method «GET»</b>	<pre>GET token=&amp;method=ListZips&amp;countryCode=643&amp;cityCode=1 RAW body {   "token": "",   "method": "ListZips",   "countryCode": 643,   "cityCode": 1 }</pre>
<b>Response example:</b>	<pre>{   "result":   [     {       "id": "620027",       "CountryCode": 643,       "CityRU": "Екатеринбург",       "CityEng": "Ekaterinburg",       "TariffZone": 4     }   ],   "error": { "isError": false } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$url = 'https://go.bxb.delivery/json.php?token=&amp;method=ListZips'; \$handle = fopen(\$url, 'rb'); \$content = stream_get_contents(\$handle); fclose(\$handle); if (!\$content) {     echo 'No response received'; } \$data = json_decode(\$content, true);  if (empty(\$data)) {     echo 'Error decode response'; } if (!empty(\$data['errors']['isError'])) {     //A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     //Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [       'result' =&gt; [         [           'id' =&gt; '620027',           'CountryCode' =&gt; 643,           'CityRU' =&gt; 'Екатеринбург',           'CityEng' =&gt; 'Ekaterinburg',           'TariffZone' =&gt; 4         ]       ]     ]     */ }</pre>

	<pre> ] ], 'error' =&gt; [   'isError' =&gt; false ] ] */ } </pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a>
<b>SOAP Request</b>	<pre> &lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_public.php"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;c:ListZips soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;       &lt;ListZipsQuery xsi:type="c:ListZipsQuery"&gt;         &lt;countryCode xsi:type="xsd:int"&gt;643&lt;/countryCode&gt;         &lt;cityCode xsi:type="xsd:int"&gt;1&lt;/cityCode&gt;         &lt;token xsi:type="xsd:string"&gt;&lt;/token&gt;       &lt;/ListZipsQuery&gt;     &lt;/c:ListZips&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<b>SOAP Response</b>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/_soap/1c_public.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:ListZipsResponse&gt;       &lt;return xsi:type="ns1:ArrayOfListZipsData"&gt;         &lt;item xsi:type="ns1:ListZipsData"&gt;           &lt;id xsi:type="xsd:int"&gt;656039&lt;/id&gt;           &lt;CountryCode xsi:type="xsd:int"&gt;643&lt;/CountryCode&gt;           &lt;CityRU xsi:type="xsd:string"&gt;Барнаул&lt;/CityRU&gt;           &lt;CityEng xsi:type="xsd:string"&gt;Barnaul&lt;/CityEng&gt;           &lt;TariffZone xsi:type="xsd:int"&gt;0&lt;/TariffZone&gt;           &lt;Delivery xsi:nil="true"/&gt;         &lt;/item&gt;       &lt;/return&gt;     &lt;/ns1:ListZipsResponse&gt;   &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<b>Example for PHP 5.0 or later version</b>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/_soap/1c_public.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]); \$request = [ </pre>

```
'token' => ",
];
try {
    $result = $client->ListZips($request);
} catch (\SoapFault $e) {
    //Request execution error
}

if (isset($e)) {
    print_r($e);
} else {
    print_r($result);
    /*
    [
        'result' => [
            [
                'id' => '620027',
                'CountryCode' => 643,
                'CityRU' => 'Екатеринбург',
                'CityEng' => 'Ekaterinburg',
                'TariffZone' => 4
            ]
        ],
        'error' => [
            'isError' => false
        ]
    ]
    */
}
```



## ListPoints

Description of API method parameters (for JSON):

*Datatable 17. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	ListCities
countryCode	integer	no	Country code (ISO 3166-1 numeric)
cityCode	integer	no	City code
prepaid	bool	no	By default, returns a list of points with the possibility of payment upon receipt of order
interRefunds	bool	no	When 1, displays only points accepting international returns. Default 0

*Datatable 18. Output parameters*

Parameter	Type	Required	Description
result	array	yes	Result
error	array	yes	Errors

*Datatable 19. Parameters in result*

Parameter	Type	Required	Description
Code	integer	yes	Piont code
Name	string	yes	Name point
Address	string	yes	Address
Phone	string	yes	Phone
WorkSchedule	string	yes	Work schedule
TripDescription	string	no	Trip description
CityCode	integer	yes	City code
CityName	string	yes	City name
GPS	string	yes	GPS coordinates
OnlyPrepaidOrders	bool	yes	If the value is "Yes" - a point only works with fully paid orders
Acquiring	bool	no	If value "YES" - pickup point can accept payments with the Debit/Credit cards
CountryCode	integer	yes	Country code
CountryName	string	yes	Country name
ZipCode	integer	no	Require with countryCode 643/804
DeliveryPeriod	integer	yes	Delivery time
Type	string	yes	Point type
InterRefunds	bool	yes	Attribute the point is working with accepting international returns
MaxWeight	integer	yes	Weight Limit

*Datatable 20. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [ListPoints](#).

*Datatable 21. Method ListPoints*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php?token= &amp;method=ListPoints">https://go.bxb.delivery/json.php?token= &amp;method=ListPoints</a>
<b>Request method «GET»</b>	<pre>GET token= &amp;method=ListPoints&amp;countryCode=643&amp;cityCode=1 RAW body {"token": "", "method": "ListPoints", "countryCode": 643, "cityCode": 1}</pre>
<b>Response example:</b>	<pre>{   "result": [     {       "Code": "10.002",       "Name": "Москва 4-ая Тверская-Ямская_7711_С",       "Address": "Тверская-Ямская 4-Я ул, д.2/11, строение 2",       "Phone": "+7(499)391-56-22",       "WorkSchedule": "",       "TripDescription": "Метро: Маяковская. 1-й вагон из центра. Примерное расстояние от метро до отделения - 300 метров. Жилой дом. Второй подвальный спуск. Вход от черного забора 2 дверь. Цокольный этаж.",       "CityCode": "77",       "CityName": "Москва",       "GPS": "55.771884,37.598411",       "OnlyPrepaidOrders": "0",       "Acquiring": "0",       "CountryName": "Россия",       "CountryCode": "643",       "ZipCode": "101000",       "InterRefunds": "0",       "MaxWeight": 10     }   ],   "error": {     "isError": false   } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$url = 'https://go.bxb.delivery/json.php?token=&amp;method=ListPoints'; \$handle = fopen(\$url, 'rb'); \$content = stream_get_contents(\$handle); fclose(\$handle); if (!\$content) {     echo 'No response received'; } \$data = json_decode(\$content, true);  if (empty(\$data)) {     echo 'Error decode response'; } if (!empty(\$data['errors']['isError'])) {     //A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; }</pre>

	<pre> } else { //Everything is fine, the answer is received, data in \$data: /* print_r(\$data); [ 'result' =&gt; [ [ 'Code' =&gt; '10.002', 'Name' =&gt; 'Москва 4-ая Тверская-Ямская_7711_C', 'Address' =&gt; 'Тверская-Ямская 4-Я ул, д.2/11, строение 2', 'Phone' =&gt; '+7(499)391-56-22', 'WorkSchedule' =&gt; ", 'TripDescription' =&gt; 'Метро: Маяковская. 1-й вагон из центра. Примерное расстояние от метро до отделения - 300 метров. Жилой дом. Второй подвальный спуск. Вход от черного забора 2 дверь. Цокольный этаж.', 'CityCode' =&gt; '77', 'CityName' =&gt; 'Москва', 'GPS' =&gt; '55.771884,37.598411', 'OnlyPrepaidOrders' =&gt; '0', 'Acquiring' =&gt; '0', 'CountryName' =&gt; 'Россия', 'CountryCode' =&gt; '643', 'ZipCode' =&gt; '101000', 'InterRefunds' =&gt; '0', 'MaxWeight' =&gt; 10 ] ], 'error' =&gt; [ 'isError' =&gt; false ] ] */ } </pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a>
<b>SOAP Request</b>	<pre> &lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_public.php"&gt; &lt;soapenv:Header/&gt; &lt;soapenv:Body&gt; &lt;c&gt;ListPoints soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;ListPointsQuery xsi:type="c&gt;ListPointsQuery"&gt; &lt;countryCode xsi:type="xsd:string"&gt;643&lt;/countryCode&gt; &lt;cityCode xsi:type="xsd:string"&gt;1&lt;/cityCode&gt; &lt;prepaid xsi:type="xsd:boolean"&gt;1&lt;/prepaid&gt; &lt;interRefunds xsi:type="xsd:boolean"&gt;1&lt;/interRefunds&gt; &lt;token xsi:type="xsd:string"&gt;&lt;/token&gt; &lt;/ListPointsQuery&gt; &lt;/c&gt;ListPoints&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<b>SOAP Response</b>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" </pre>

	<pre> xmlns:ns1="https://go.bxb.delivery/__soap/1c_public.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;SOAP-ENV:Body&gt;   &lt;ns1:ListPointsResponse&gt;     &lt;return xsi:type="ns1:ArrayOfListPointsData"&gt;       &lt;item xsi:type="ns1:ListPointsData"&gt;         &lt;Name xsi:type="xsd:string"&gt;Октябрьский Девонская_0256_C&lt;/Name&gt;         &lt;Address xsi:type="xsd:string"&gt;Девонская ул, д.8, корпус 1&lt;/Address&gt;         &lt;Phone xsi:type="xsd:string"&gt;8-800-222-80-00&lt;/Phone&gt;         &lt;WorkSchedule xsi:type="xsd:string"&gt;&lt;/WorkSchedule&gt;         &lt;TripDescription xsi:type="xsd:string"&gt;Проезд:\навтобус - №№1, 1М, 14, 16, 29, 40.\nОстановка \"Ресторан Башкирия\".\nC остановки \"Ресторан Башкирия\" в сторону перекрестка улиц Девонская и Садовое кольцо. Отделение выдачи расположено в магазине \"Лайм\".&lt;/TripDescription&gt;         &lt;CityCode xsi:type="xsd:int"&gt;77&lt;/CityCode&gt;         &lt;CityName xsi:type="xsd:string"&gt;Москва&lt;/CityName&gt;         &lt;TariffZone xsi:type="xsd:int"&gt;0&lt;/TariffZone&gt;         &lt;GPS xsi:type="xsd:string"&gt;54.489174,53.457163&lt;/GPS&gt;         &lt;OnlyPrepaidOrders xsi:type="xsd:boolean"&gt;&gt;false&lt;/OnlyPrepaidOrders&gt;         &lt;Acquiring xsi:type="xsd:boolean"&gt;&gt;false&lt;/Acquiring&gt;         &lt;CountryName xsi:type="xsd:string"&gt;Россия&lt;/CountryName&gt;         &lt;CountryCode xsi:type="xsd:int"&gt;643&lt;/CountryCode&gt;         &lt;ZipCode xsi:type="xsd:int"&gt;452607&lt;/ZipCode&gt;         &lt;DeliveryPeriod xsi:type="xsd:int"&gt;0&lt;/DeliveryPeriod&gt;         &lt;Type xsi:type="xsd:string"&gt;&lt;/Type&gt;         &lt;InterRefunds xsi:type="xsd:boolean"&gt;&gt;true&lt;/InterRefunds&gt;         &lt;MaxWeight xsi:type="xsd:int"&gt;10&lt;/MaxWeight&gt;       &lt;/item&gt;     &lt;/return&gt;   &lt;/ns1:ListPointsResponse&gt; &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_public.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]);  \$request = [   'token' =&gt; "", ]; try {   \$result = \$client-&gt;ListPoints(\$request); } catch (\SoapFault \$e) {   //Request execution error }  if (isset(\$e)) { </pre>

```
print_r($e);
} else {
print_r($result);
/*
[
  'result' => [
    [
      'Code' => '10.002',
      'Name' => 'Москва 4-ая Тверская-Ямская_7711_C',
      'Address' => 'Тверская-Ямская 4-Я ул, д.2/11, строение 2',
      'Phone' => '+7(499)391-56-22',
      'WorkSchedule' => "",
      'TripDescription' => 'Метро: Маяковская. 1-й вагон из центра.
Примерное расстояние от метро до отделения - 300 метров. Жилой дом.
Второй подвальный спуск. Вход от черного забора 2 дверь. Цокольный этаж.',
      'CityCode' => '77',
      'CityName' => 'Москва',
      'GPS' => '55.771884,37.598411',
      'OnlyPrepaidOrders' => '0',
      'Acquiring' => '0',
      'CountryName' => 'Россия',
      'CountryCode' => '643',
      'ZipCode' => '101000',
      'InterRefunds' => '0',
      'MaxWeight' => 10
    ]
  ],
  'error' => [
    'isError' => false
  ]
]
*/
}
```

## GetStatuses

Description of API method parameters (for JSON):

*Datatable 22. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	GetStatuses
track	string	yes	Tracks (comma separated)

*Datatable 23. Output parameters*

Parameter	Type	Required	Description
result	array	yes	Parcels
error	array	yes	Errors

*Datatable 24. Parameters for status*

Parameter	Type	Required	Description
Code	string	yes	Identifier
NameRu	string	yes	Name in Russian
NameEng	string	yes	Name in English
Date	string	yes	Date
Comment	string	no	Status Comment

*Datatable 25. Parameters for tracking*

Parameter	Type	Required	Description
id	string	yes	Identifier
status	array	yes	Statuses

*Datatable 26. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method `GetStatuses`.

*Datatable 27. Method `GetStatuses`*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php?token=&amp;method=GetStatuses">https://go.bxb.delivery/json.php?token=&amp;method=GetStatuses</a>
<b>Request method</b> «GET»	<p>GET</p> <p>token=&amp;method=GetStatuses&amp;track=track_id_1</p> <p>RAW body</p> <pre>{   "token": "",   "method": "GetStatuses",   "track": "track_id_1" }</pre>
<b>Response example:</b>	<pre>{   "result": [     {       "id": "track_id_1",       "status": [         {           "Code": "10",           "NameRu": "Получена информация о заказе",           "NameEng": "Package details received",           "Date": "дата статуса ДД.ММ.ГГГГ",           "Comment": "комментарий"         }       ]     }   ],   "error": {     "isError": false   } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$url = 'https://go.bxb.delivery/json.php?token=&amp;method=GetStatuses'; \$handle = fopen(\$url, 'rb'); \$content = stream_get_contents(\$handle); fclose(\$handle); if (!\$content) {     echo 'No response received'; } \$data = json_decode(\$content, true);  if (empty(\$data)) {     echo 'Error decode response'; } if (!empty(\$data['errors']['isError'])) {     //A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     //Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [</pre>

	<pre>'result' =&gt; [   [     'id' =&gt; 'track_id_1',     'status' =&gt; [       [         'Code' =&gt; '10',         'NameRu' =&gt; 'Получена информация о заказе',         'NameEng' =&gt; 'Package details received',         'Date' =&gt; 'дата статуса ДД.ММ.ГГГГ',         'Comment' =&gt; 'комментарий'       ]     ]   ] ], 'error' =&gt; [   'isError' =&gt; false ] ] */ }</pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a>
<b>SOAP Request</b>	<pre>&lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_public.php"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;c:GetStatuses soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;       &lt;GetStatusQuery xsi:type="c:GetStatusQuery"&gt;         &lt;track xsi:type="xsd:string"&gt;track_num&lt;/track&gt;         &lt;token xsi:type="xsd:string"&gt;&lt;/token&gt;       &lt;/GetStatusQuery&gt;     &lt;/c:GetStatuses&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;</pre>
<b>SOAP Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/_soap/1c_public.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:GetStatusesResponse&gt;       &lt;return xsi:type="ns1:ArrayOfGetStatusData"&gt;         &lt;item xsi:type="ns1:GetStatusData"&gt;           &lt;id xsi:type="xsd:string"&gt;ID000001&lt;/id&gt;           &lt;status xsi:type="ns1:ArrayOfStatusData"&gt;             &lt;item xsi:type="ns1:StatusData"&gt;               &lt;Code xsi:type="xsd:string"&gt;10&lt;/Code&gt;               &lt;NameRu xsi:type="xsd:string"&gt;Получена информация о заказе&lt;/NameRu&gt;               &lt;NameEng xsi:type="xsd:string"&gt;Package details received&lt;/NameEng&gt;</pre>



	<pre>                 &lt;Date xsi:type="xsd:string"&gt;10.12.2019&lt;/Date&gt;                 &lt;Comment xsi:nil="true"/&gt;             &lt;/item&gt;         &lt;/status&gt;     &lt;/item&gt; &lt;/return&gt; &lt;/ns1:GetStatusesResponse&gt; &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre> &lt;?php  \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_public.php', [     'cache_wsdl' =&gt; WSDL_CACHE_NONE,     'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]);  \$request = [     'token' =&gt; "", ];  try {     \$result = \$client-&gt;GetStatuses(\$request); } catch (\SoapFault \$e) {     //Request execution error }  if (isset(\$e)) {     print_r(\$e); } else {     print_r(\$result);     /*     [         'result' =&gt; [             [                 'id' =&gt; 'track_id_1',                 'status' =&gt; [                     [                         'Code' =&gt; '10',                         'NameRu' =&gt; 'Получена информация о заказе',                         'NameEng' =&gt; 'Package details received',                         'Date' =&gt; 'дата статуса ДД.ММ.ГГГГ',                         'Comment' =&gt; 'комментарий'                     ]                 ]             ]         ]     ],     'error' =&gt; [         'isError' =&gt; false     ]     */ } </pre>

## DeliveryCalc

The method allows you to calculate the cost of parcel.

Description of API method parameters (for JSON):

*Datatable 28. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	DeliveryCalc
weight	float	yes	The weight of the parcel, with type = 2, the value should not exceed 15 kg, with type = 1, a check is performed for MaxWeight of the issuing point
type	bool	yes	Delivery type, 1 = to point, 2 = express
destination	integer	yes	Postcode - express delivery, else point code
value	float	no	Shipping cost, for insurance calculation, 0 if not specified
currency	string	no	Currency of value. The default currency shop
cod	float	no	Cash on delivery

*Datatable 29. Output parameters*

Parameter	Type	Required	Description
DelCost	float	yes	Delivery cost
ServiceCost	float	yes	Cost of additional services
Total	float	yes	The sum of the cost of delivery and additional services
Currency	string	yes	Currency
TariffZone	integer	yes	Tariff zone
error	array	yes	Errors

*Datatable 30. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method `GetStatuses`.

*Datatable 31. Method DeliveryCalc*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php?token= &amp;method=DeliveryCalc">https://go.bxb.delivery/json.php?token= &amp;method=DeliveryCalc</a>
<b>Request method</b> «GET»	<p>GET token= &amp;method=DeliveryCalc&amp;weight=900&amp;type=1&amp; destination=19753&amp;value=50&amp;currency=USD</p> <p>RAW body { "token": "", "method": "DeliveryCalc", "weight": 900, "type": 1, "destination": 19753, "value": 50, "currency": "USD" }</p>
<b>Response example:</b>	<pre>{   "result": {     "DelCost": 8,     "ServiceCost": 2,     "Total": 10,     "Currency": "USD",     "TariffZone": 3,     "error": {       "isError": false     }   } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php  \$url = 'https://go.bxb.delivery/json.php?token= &amp;method=DeliveryCalc'; \$handle = fopen(\$url, 'rb'); \$content = stream_get_contents(\$handle); fclose(\$handle); if (!\$content) {     echo 'No response received'; } \$data = json_decode(\$content, true);  if (empty(\$data)) {     echo 'Error decode response'; } if (!empty(\$data['errors']['isError'])) {     //A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     //Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [       'result' =&gt; [         'DelCost' =&gt; 8,         'ServiceCost' =&gt; 2,         'Total' =&gt; 10,         'Currency' =&gt; 'USD',</pre>

	<pre>'TariffZone' =&gt; 3, 'error' =&gt; [   'isError' =&gt; false ] ] ] */ }</pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a>
<b>SOAP Request</b>	<pre>&lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_public.php"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;c:DeliveryCalc soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;       &lt;DeliveryCalcQuery xsi:type="c:DeliveryCalcQuery"&gt;         &lt;weight xsi:type="xsd:int"&gt;900&lt;/weight&gt;         &lt;type xsi:type="xsd:boolean"&gt;1&lt;/type&gt;         &lt;destination xsi:type="xsd:int"&gt;19753&lt;/destination&gt;         &lt;value xsi:type="xsd:float"&gt;50&lt;/value&gt;         &lt;currency xsi:type="xsd:string"&gt;USD&lt;/currency&gt;         &lt;cod xsi:type="xsd:float"&gt;&lt;/cod&gt;         &lt;token xsi:type="xsd:string"&gt;&lt;/token&gt;       &lt;/DeliveryCalcQuery&gt;     &lt;/c:DeliveryCalc&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;</pre>
<b>SOAP Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/_soap/1c_public.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:DeliveryCalcResponse&gt;       &lt;return xsi:type="ns1:DeliveryCalcData"&gt;         &lt;DelCost xsi:type="xsd:float"&gt;804&lt;/DelCost&gt;         &lt;ServiceCost xsi:type="xsd:float"&gt;120&lt;/ServiceCost&gt;         &lt;Total xsi:type="xsd:float"&gt;924&lt;/Total&gt;         &lt;Currency xsi:type="xsd:string"&gt;643&lt;/Currency&gt;         &lt;TariffZone xsi:type="xsd:int"&gt;3&lt;/TariffZone&gt;       &lt;/return&gt;     &lt;/ns1:DeliveryCalcResponse&gt;   &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt;</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$client = new \SoapClient('https://go.bxb.delivery/_soap/1c_public.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]);</pre>

```
$request = [  
    'token' => "",  
];  
  
try {  
    $result = $client->DeliveryCalc($request);  
} catch (\SoapFault $e) {  
    //Request execution error  
}  
  
if (isset($e)) {  
    print_r($e);  
} else {  
    print_r($result);  
    /*  
    [  
        'result' => [  
            'DelCost' => 8,  
            'ServiceCost' => 2,  
            'Total' => 10,  
            'Currency' => 'USD',  
            'TariffZone' => 3,  
            'error' => [  
                'isError' => false  
            ]  
        ]  
    ]  
    */  
}
```

## PUPDetails

Description of API method parameters (for JSON):

*Datatable 32. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	PUPDetails
code	string	yes	Piont code

*Datatable 33. Output parameters*

Parameter	Type	Required	Description
Code	string	yes	Piont code
Name	string	yes	Name point
Photo	array	yes	Photos of pick-up point
error	array	yes	Errors

*Datatable 34. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [PUPDetails](#).

*Datatable 35. Method PUPDetails*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php?token= &amp;method=PUPDetails">https://go.bxb.delivery/json.php?token= &amp;method=PUPDetails</a>
<b>Request method «GET»</b>	
<b>Response example:</b>	<pre>{   "result": {     "Code": 96951,     "Name": "Екатеринбург Крауля_9695_C",     "Photo": [       "https://boxberry.ru/photo_pvz/22221_1.jpg"     ],     "error": {       "isError": false     }   } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$url = 'https://go.bxb.delivery/json.php?token= &amp;method=PUPDetails'; \$handle = fopen(\$url, 'rb'); \$content = stream_get_contents(\$handle); fclose(\$handle); if (!\$content) {     echo 'No response received'; } \$data = json_decode(\$content, true);  if (empty(\$data)) {     echo 'Error decode response'; } if (!empty(\$data['errors']['isError'])) {     //A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     //Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [       'result' =&gt; [         'Code' =&gt; 96951,         'Name' =&gt; 'Екатеринбург Крауля_9695_C',         'Photo' =&gt; [           'https://boxberry.ru/photo_pvz/22221_1.jpg'         ],         'error' =&gt; [           'isError' =&gt; false         ]       ]     ]     */   } }</pre>

<p><b>SOAP WSDL URL</b></p> <p><b>SOAP Request</b></p>	<p><a href="https://go.bxb.delivery/_soap/1c_public.php">https://go.bxb.delivery/_soap/1c_public.php</a></p> <pre>&lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_public.php"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;c:PUPDetails soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;       &lt;PUPDetailQuery xsi:type="c:PUPDetailsQuery"&gt;         &lt;code xsi:type="xsd:string"?&lt;/code&gt;         &lt;token xsi:type="xsd:string"?&lt;/token&gt;       &lt;/PUPDetailQuery&gt;     &lt;/c:PUPDetails&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;</pre>
<p><b>SOAP Response</b></p>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/_soap/1c_public.php" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:PUPDetailsResponse&gt;       &lt;return xsi:type="ns1:PUPDetailsData"&gt;         &lt;Code xsi:nil="true"/&gt;         &lt;Name xsi:type="xsd:string"&gt;Екатеринбург Крауля_9695_C&lt;/Name&gt;         &lt;Photo SOAP-ENC:arrayType="xsd:string[1]" xsi:type="SOAP- ENC:Array"&gt;           &lt;item xsi:type="xsd:string"&gt;https://boxberry.ru/photo_pvz/96951_1.jpg&lt;/item&gt;         &lt;/Photo&gt;       &lt;/return&gt;     &lt;/ns1:PUPDetailsResponse&gt;   &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt;</pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre>&lt;?php \$client = new \SoapClient('https://go.bxb.delivery/_soap/1c_public.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]);  \$request = [   'token' =&gt; '?', ];  try {   \$result = \$client-&gt;PUPDetails(\$request); } catch (\SoapFault \$e) {   //Request execution error }  if (isset(\$e)) {   print_r(\$e); }</pre>



```
} else {
  print_r($result);
  /*
  [
    'result' => [
      'Code' => 96951,
      'Name' => 'Екатеринбург Крауля_9695_C',
      'Photo' => [
        'https://boxberry.ru/photo_pvz/22221_1.jpg'
      ],
      'error' => [
        'isError' => false
      ]
    ]
  ]
  */
}
```

## ZipCheck

Description of API method parameters (for JSON):

*Datatable 36. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	ZipCheck
countryCode	integer	no	Country code (ISO 3166-1 numeric)
zipcode	integer	no	Postcode

*Datatable 37. Output parameters*

Parameter	Type	Required	Description
Delivery	boolean	yes	1 if delivery is available, otherwise 0
error	array	yes	Errors
Delivery	boolean	yes	1 if delivery is available, otherwise 0
error	array	yes	Errors

*Datatable 38. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [ZipCheck](#).

*Datatable 39. Method ZipCheck*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php?token= &amp;method=ZipCheck">https://go.bxb.delivery/json.php?token= &amp;method=ZipCheck</a>
<b>Request method</b> «GET»	GET token=&method=ZipCheck&countryCode=643&zipcode=656039 RAW body { "token": "", "method": "ListZips", "countryCode": 643, "zipcode": 656039 }
<b>Response example:</b>	{ "Delivery": true, "error": { "isError": false } }
<b>Example for PHP 5.0 or later version</b>	<?php \$url = 'https://go.bxb.delivery/json.php?token= &method=ZipCheck'; \$handle = fopen(\$url, 'rb'); \$contents = stream_get_contents(\$handle); fclose(\$handle); if (!\$contents) { echo 'No response received'; } \$data = json_decode(\$contents, true);  if (empty(\$data)) { echo 'Error decode response'; } if (!empty(\$data['errors']['isError'])) { //A request is made with an error echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else { //Everything is fine, the answer is received, data in \$data: /* print_r(\$data); [ 'Delivery' => true, 'error' => [ 'isError' => false ] ] */ }
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/soap/1c_public.php">https://go.bxb.delivery/soap/1c_public.php</a>
<b>SOAP Request</b>	<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/__soap/1c_public.php"> <soapenv:Header/> <soapenv:Body>

	<pre> &lt;c:ListZips soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;ZipCheckQuery&gt;     &lt;token xsi:type="xsd:string"&gt; &lt;/token&gt;     &lt;countryCode&gt;643&lt;/countryCode&gt;     &lt;zipcode&gt;656039&lt;/zipcode&gt;   &lt;/ZipCheckQuery&gt; &lt;/c:ListZips&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<p><b>SOAP Response</b></p>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/__soap/1c_public.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:ZipCheckResponse&gt;       &lt;return xsi:type="ns1:ZipCheckData"&gt;         &lt;Delivery xsi:type="xsd:boolean"&gt;true&lt;/Delivery&gt;       &lt;/return&gt;     &lt;/ns1:ZipCheckResponse&gt;   &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre> &lt;?php  \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_public.php', [     'cache_wsdl' =&gt; WSDL_CACHE_NONE,     'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]);  \$request = [     'token' =&gt; "", ];  try {     \$result = \$client-&gt;ZipCheck(\$request); } catch (\SoapFault \$e) {     //Request execution error }  if (isset(\$e)) {     print_r(\$e); } else {     print_r(\$result);     /*     [         'Delivery' =&gt; true,         'error' =&gt; [             'isError' =&gt; false         ]     ]     */ } </pre>

## Methods used to transfer data

JSON Url: <https://go.bxb.delivery/json.php>  
 SOAP Url: [https://go.bxb.delivery/soap/1c\\_1c.php](https://go.bxb.delivery/soap/1c_1c.php)

*Datatable 40. JSON, SOAP protocol available methods*

Num	Method name	Description
1.	<b>CreateParcel</b>	method allows you to create a parcel in Boxberry personal account of. Read more at page 38.
2.	<b>DeleteParcel</b>	The method allows you to delete a parcel by the track number Read more at page 46.
3.	<b>UpdateParcel</b>	method allows you to update data on a previously created parcel in the Boxberry system. Read more at page 50.
4.	<b>ManifestControl</b>	method allows you to create manifests, manage their content and form pre-alert. Read more at page 59.
5.	<b>WarehouseControl</b>	method allows you to create / edit / delete / view the warehouse. Read more at page 62.
6	<b>ControlUser</b>	method allows you to create / edit a user. Read more at page 67.

## CreateParcel

Description of API method parameters (for JSON):

*Datatable 41. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	CreateParcel
parcels	array	yes	Parcels data

*Datatable 42. Parameters for parcels*

Parameter	Type	Required	Description
barcode	string	yes	The barcode, if not filled, is assigned from the pool
orderNum	string	yes	Order number
countryTo	integer	yes	Country of destination in ISO 3166-1 numeric
type	integer	yes	Delivery Type, 1 or 2
pointcode	string	no	Point
address	array	no	Address
recipient	array	yes	Recipient Data
passport	array	yes	Recipient Document Data
seller	array	yes	Seller data
cod	array	no	Cash on delivery data; the availability of this service is checked with IM
box	array	yes	Boxes

*Datatable 43. Parameters for address*

Parameter	Type	Required	Description
postcode	string	no	Postcode, required with type = 2
city	string	no	City, required with type = 2
addressString	string	no	Address, required with type = 2, if street / home / building / flat is not set
street	string	no	Street, required with type = 2 if addressString is not set
home	string	no	Home, required with type = 2 if addressString is not set
building	string	no	Build, required with type = 2 if addressString is not set
flat	string	no	Flat/office, required with type = 2, if addressString is not set

*Datatable 44. Parameters for recipient*

Parameter	Type	Required	Description
ownerid	string	no	User ID in the passport data storage
fullNameString	string	no	Name of the recipient, required if name / surname / middlename is not set
name	string	no	Name, required if fullNameString is not set
surname	string	no	Surname, required if fullNameString is not set
middlename	string	no	Middle name, required if fullNameString is not set
email	string	yes	Email
phone	string	yes	Phone

Datatable 45. Parameters for passport

Parameter	Type	Required	Description
countryCode	integer	no	Country of issue, ISO 3166-1 numeric
passport	string	no	Series and number of the document, check on the mask of the country's passport
issuedWhen	string	no	Date of issue of the document, dd.mm.yyyy
issuedBy	string	no	Issuing authority
inn	string	no	INN, with countryCode = 643 required
kzId	string	no	IIN, required with countryTo = 398 and countryCode passports = 398
kzIdNum	string	no	Document number, required with countryTo = 398 and countryCode passport = 398
birthDate	string	no	Date of birth, dd.mm.yyyy

Datatable 46. Parameters for seller

Parameter	Type	Required	Description
EORI	string	no	EORI Number
legalName	string	no	Official name
webAddress	string	no	Link to the seller's website
sAddress	array	no	Massive with data of seller address

Datatable 47. Parameters for massive sAddress

Parameter	Type	Required	Description
countryCode	integer	no	Country, ISO 3166-1 numeric
postcode	integer	no	Postcode
city	string	no	City
SaddressString	string	no	Address, full address is filled, or individual address parameters
street	string	no	Street
home	string	no	Home
building	string	no	Building
flat	string	no	Flat/office

Datatable 48. Parameters for cod

Parameter	Type	Required	Description
value	float	no	Amount to be paid
currency	string	no	Currency payable

Datatable 49. Parameters for box

Parameter	Type	Required	Description
size	array	yes	Dimensions
weightBruto	float	yes	Weight, in grams, with type = 2 ranging from 1 to 15000. With type = 1, check for the maximum weight of the point
items	array	yes	Goods item data

Datatable 50. Parameters for size

Parameter	Type	Required	Description
x	float	yes	Length, cm
y	float	yes	Width, cm
z	float	yes	Height, cm

v	float	yes	Volume, cc
---	-------	-----	------------

*Datatable 51. Parameters for items*

Parameter	Type	Required	Description
sku	string	yes	SKU
brand	string	yes	Brand / Manufacturer
name	string	yes	Name
quantity	integer	yes	Quantity
price	float	yes	Unit Price
currency	string	yes	Currency
weight	float	no	Weight, grams
descrRu	string	no	Description of goods in Russian, the presence of a description in one of the parameters descr is checked
descrEn	string	no	Description of goods in English, the presence of a description in one of the parameters descr is checked
descrAlt	string	no	Description of goods in the seller's language, the presence of a description in one of the parameters descr
webLink	string	yes	Link to goods on the Internet
webPhoto	string	no	Link to goods image
country	integer	no	Country of manufacture, ISO 3166-1 numeric
notiNum	string	no	Notification Number
notiDate	string	no	Notification end date, dd.mm.yyyy
notiCode	integer	no	Notification Code
name	string	yes	Name

*Datatable 52. Output parameters*

Parameter	Type	Required	Description
track	string	yes	Order ID
orderNum	string	yes	Order number
label	string	yes	Label Printing Link
barcode	string	yes	Barcode
error	array	yes	Errors

*Datatable 53. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description



Example request and response for method [CreateParcel](#).

*Datatable 54. Method CreateParcel*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a>
<b>Request method</b> <b>«POST»</b>	RAW body <pre> {   "method": "CreateParcel",   "token": "",   "parcels": [     {       "orderNum": "orderNum-1588155275-2",       "countryTo": 643,       "address": {         "postcode": 656039       },       "pointcode": "02561",       "type": 1,       "recipient": {         "phone": "79876543210",         "email": "exampla@example.com",         "fullNameString": "Иванов Иван"       },       "goods_cost_currency": "RUB",       "box": [         {           "size": {             "x": "54.2",             "y": 41.9,             "z": 37.1           },           "weightBruto": 380,           "items": [             {               "sku": "#sku-2746",               "brand": "Samsung",               "name": "SyncMaster 943",               "currency": "RUB",               "quantity": 1,               "price": 5199.99,               "webLink": "http://boxberry.ru",               "descrEn": "Lorem ipsum"             }           ]         }       ]     }   ],   {     "orderNum": "orderNum-1588155275-14",     "countryTo": 643,     "address": {       "postcode": 656039     },     "pointcode": "02561",     "token": 123, </pre>

	<pre> "type": 1, "recipient": {   "phone": "79876543210",   "email": "example@example.com",   "fullNameString": "Иванов Иван" }, "goods_cost_currency": "RUB", "box": [   {     "size": {       "x": "51.2",       "y": "45.7",       "z": "38.9"     },     "weightBruto": 330,     "items": [       {         "sku": "#sku-4699",         "brand": "Samsung",         "name": "SyncMaster 943",         "currency": "RUB",         "quantity": 1,         "price": 5199.99,         "webLink": "http://boxberry.ru",         "descrEn": "Lorem ipsum"       }     ]   } ] } </pre>
<p><b>Response example:</b></p>	<pre> {   "result": [     {       "track": "LKIM0000079924",       "orderNum": "orderNum-1588155275-2",       "label":         "https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX1",       "barcode": "LKIM0000079924"     },     {       "track": "LKIM0000079925",       "orderNum": "orderNum-1588155275-14",       "label":         "https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX2",       "barcode": "LKIM0000079925"     }   ],   "error": {     "isError": false   } } </pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre> &lt;?php \$ch = curl_init(); curl_setopt(\$ch, CURLOPT_URL, 'https://go.bxb.delivery/json.php'); </pre>

	<pre> curl_setopt(\$ch, CURLOPT_POST, true); //\$request - RAW body curl_setopt(\$ch, CURLOPT_POSTFIELDS, \$request); curl_setopt(\$ch, CURLOPT_HTTPHEADER, ['Content-Type:application/json']); curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); \$data = json_decode(curl_exec(\$ch), 1); if (!empty(\$data['errors']['isError'])) {     //A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     //Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [     'result' =&gt; [         [             'track' =&gt; 'LKIM0000079924',             'orderNum' =&gt; 'orderNum-1588155275-2',             'label' =&gt; 'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX1',             'barcode' =&gt; 'LKIM0000079924'         ],         [             'track' =&gt; 'LKIM0000079925',             'orderNum' =&gt; 'orderNum-1588155275-14',             'label' =&gt; 'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX2',             'barcode' =&gt; 'LKIM0000079925'         ]     ],     'error' =&gt; [         'isError' =&gt; false     ]     ]     */ } </pre>
<p><b>SOAP WSDL URL</b></p>	<p><a href="https://go.bxb.delivery/_soap/1c_lc.php">https://go.bxb.delivery/_soap/1c_lc.php</a></p>
<p><b>SOAP Request</b></p>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="https://go.bxb.delivery/_soap/1c_lc.php"&gt;   &lt;soapenv:Header /&gt;   &lt;soapenv:Body&gt;     &lt;ns:CreateParcel&gt;       &lt;token&gt;&lt;/token&gt;       &lt;parcels&gt;         &lt;item&gt;           &lt;orderNum&gt;orderNum-1587635191&lt;/orderNum&gt;           &lt;countryTo&gt;643&lt;/countryTo&gt;           &lt;pointcode&gt;19753&lt;/pointcode&gt;           &lt;type&gt;1&lt;/type&gt;           &lt;recipient&gt;             &lt;phone&gt;79876543210&lt;/phone&gt;             &lt;email&gt;example@example.com&lt;/email&gt;             &lt;fullNameString&gt;Иванов Иван&lt;/fullNameString&gt;           &lt;/recipient&gt;         &lt;/item&gt;       &lt;/parcels&gt;     &lt;/ns:CreateParcel&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>

	<pre> &lt;item&gt;   &lt;size&gt;     &lt;x&gt;54.3&lt;/x&gt;     &lt;y&gt;39.7&lt;/y&gt;     &lt;z&gt;34.6&lt;/z&gt;   &lt;/size&gt;   &lt;weightBruto&gt;380&lt;/weightBruto&gt;   &lt;items&gt;     &lt;item&gt;       &lt;brand&gt;Samsung&lt;/brand&gt;       &lt;name&gt;SyncMaster 943&lt;/name&gt;       &lt;currency&gt;RUB&lt;/currency&gt;       &lt;quantity&gt;1&lt;/quantity&gt;       &lt;price&gt;5199.99&lt;/price&gt;       &lt;webLink&gt;http://boxberry.ru&lt;/webLink&gt;       &lt;descrEn&gt;Lorem ipsum&lt;/descrEn&gt;     &lt;/item&gt;   &lt;/items&gt; &lt;/item&gt; &lt;/box&gt; &lt;/item&gt; &lt;barcode&gt;barcode-7635191&lt;/barcode&gt; &lt;/parcels&gt; &lt;/ns:CreateParcel&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<p><b>SOAP Response</b></p>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"   xmlns:ns1="https://go.bxb.delivery/__soap/1c_lc.php"   xmlns:xsd="http://www.w3.org/2001/XMLSchema"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:CreateParcelResponse&gt;       &lt;return xsi:type="ns1:ArrayOfCreateParcelData"&gt;         &lt;item xsi:type="ns1:CreateParcelData"&gt;           &lt;track xsi:type="xsd:string"&gt;track_00022&lt;/track&gt;           &lt;orderNum xsi:type="xsd:string"&gt;orderNum_0004102&lt;/orderNum&gt;           &lt;label xsi:type="xsd:string"&gt;https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id =XXX&lt;/label&gt;           &lt;barcode xsi:type="xsd:string"&gt;barcode_123456789&lt;/barcode&gt;         &lt;/item&gt;       &lt;/return&gt;     &lt;/ns1:CreateParcelResponse&gt;   &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_lc.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]); \$request = [ </pre>

```
'token' => ",
];

try {
    $result = $client->CreateParcel($request);
} catch (\SoapFault $e) {
    //Request execution error
}

if (isset($e)) {
    print_r($e);
} else {
    print_r($result);
    /*
    [
        'result' => [
            [
                'track' => 'LKIM0000079924',
                'orderNum' => 'orderNum-1588155275-2',
                'label' =>
                'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX1',
                'barcode' => 'LKIM0000079924'
            ],
            [
                'track' => 'LKIM0000079925',
                'orderNum' => 'orderNum-1588155275-14',
                'label' =>
                'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX2',
                'barcode' => 'LKIM0000079925'
            ]
        ],
        'error' => [
            'isError' => false
        ]
    ]
    */
}
```

## DeleteParcel

Description of API method parameters (for JSON):

*Datatable 55. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	DeleteParcel
track	string	yes	Order ID

*Datatable 56. Output parameters*

Parameter	Type	Required	Description
track	string	yes	Order ID
deleted	boolean	yes	Attribute of the successful removal
error	array	yes	Errors

*Datatable 57. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [DeleteParcel](#).

*Datatable 58. Method DeleteParcel*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a>
<b>Request method</b> «POST»	Request example: RAW body <pre>{   "method": "DeleteParcel",   "token": "a85dee936de5f9bc7809",   "track": "LKIM00000XXXXX" }</pre>
<b>Response example:</b>	<pre>{   "result": [     {       "track": "LKIM00000XXXXX",       "deleted": true     }   ],   "error": {     "isError": false   } }</pre>
<b>Example for PHP 5.0</b> <b>or later version</b>	<pre>&lt;?php \$ch = curl_init(); curl_setopt(\$ch, CURLOPT_URL, 'https://go.bxb.delivery/json.php'); curl_setopt(\$ch, CURLOPT_POST, true); // \$request - RAW body curl_setopt(\$ch, CURLOPT_POSTFIELDS, \$request); curl_setopt(\$ch, CURLOPT_HTTPHEADER, ['Content-Type:application/json']); curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); \$data = json_decode(curl_exec(\$ch), 1); if (!empty(\$data['errors']['isError'])) {     // A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     // Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [       'result' =&gt; [         [           'track' =&gt; 'LKIM00000XXXXX',           'deleted' =&gt; true         ]       ],       'error' =&gt; [         'isError' =&gt; false       ]     ]     */ }</pre>

<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/soap/1c_1c.php">https://go.bxb.delivery/soap/1c_1c.php</a>
<b>SOAP Request</b>	<pre>&lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/__soap/1c_1c.php"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;c:DeleteParcel soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;       &lt;DeleteParcelQuery xsi:type="c:DeleteParcelQuery"&gt;         &lt;track xsi:type="xsd:string"&gt;LKIM00000XXXXX&lt;/track&gt;         &lt;token xsi:type="xsd:string"&gt;a85dee936de5f9bc7809&lt;/token&gt;       &lt;/DeleteParcelQuery&gt;     &lt;/c:DeleteParcel&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;</pre>
<b>SOAP Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"   xmlns:ns1="https://go.bxb.delivery/__soap/1c_1c.php"   xmlns:xsd="http://www.w3.org/2001/XMLSchema"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:DeleteParcelResponse&gt;       &lt;return xsi:type="ns1:DeleteParcelData"&gt;         &lt;track xsi:type="xsd:string"&gt;LKIM00000XXXXX&lt;/track&gt;         &lt;deleted xsi:type="xsd:boolean"&gt;true&lt;/deleted&gt;       &lt;/return&gt;     &lt;/ns1:DeleteParcelResponse&gt;   &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt;</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_1c.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]); \$request = [   'token' =&gt; 'a85dee936de5f9bc7809', ]; try {   \$result = \$client-&gt;DeleteParcel(\$request); } catch (\SoapFault \$e) {   //Request execution error } if (isset(\$e)) {   print_r(\$e); } else {   print_r(\$result);   /*   [     'result' =&gt; [       [         'track' =&gt; 'LKIM00000XXXXX',         'deleted' =&gt; true       ]     ]   ]   */   */ </pre>



```
    ]  
  ],  
  'error' => [  
    'isError' => false  
  ]  
]  
*/  
}
```

## UpdateParcel

Description of API method parameters (for JSON):

*Datatable 59. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	UpdateParcel
track	string	yes	Unique track number of a previously created the parcel
barcode	string	yes	The barcode, if not filled, is assigned from the pool
orderNum	string	yes	Order number
countryTo	integer	yes	Country of destination in ISO 3166-1 numeric
type	integer	yes	Delivery Type, 1 or 2
pointcode	string	no	Point
address	array	no	Address
recipient	array	yes	Recipient Data
passport	array	yes	Recipient Document Data
seller	array	yes	Seller data
cod	array	no	Cash on delivery data; the availability of this service is checked with IM
box	array	yes	Boxes

*Datatable 60. Parameters for address*

Parameter	Type	Required	Description
postcode	string	no	Postcode, required with type = 2
city	string	no	City, required with type = 2
addressString	string	no	Address, required with type = 2, if street / home / building / flat is not set
street	string	no	Street, required with type = 2 if addressString is not set
home	string	no	Home, required with type = 2 if addressString is not set
building	string	no	Build, required with type = 2 if addressString is not set
flat	string	no	Flat/office, required with type = 2, if addressString is not set

*Datatable 61. Parameters for recipient*

Parameter	Type	Required	Description
ownerid	string	no	User ID in the passport data storage
fullNameString	string	no	Name of the recipient, required if name / surname / middlename is not set
name	string	no	Name, required if fullNameString is not set
surname	string	no	Surname, required if fullNameString is not set
middlename	string	no	Middle name, required if fullNameString is not set
email	string	yes	Email
phone	string	yes	Phone

Datatable 62. Parameters for passport

Parameter	Type	Required	Description
countryCode	integer	no	Country of issue, ISO 3166-1 numeric
passport	string	no	Series and number of the document, check on the mask of the country's passport
issuedWhen	string	no	Date of issue of the document, dd.mm.yyyy
issuedBy	string	no	Issuing authority
inn	string	no	INN, with countryCode = 643 required
kzId	string	no	IIN, required with countryTo = 398 and countryCode passports = 398
kzIdNum	string	no	Document number, required with countryTo = 398 and countryCode passport = 398
birthDate	string	no	Date of birth, dd.mm.yyyy

Datatable 63. Parameters for seller

Parameter	Type	Required	Description
EORI	string	no	EORI Number
legalName	string	no	Official name
webAddress	string	no	Link to the seller's website
sAddress	array	no	Massive with data of seller address

Datatable 64. Parameters for massive sAddress

Parameter	Type	Required	Description
countryCode	integer	no	Country, ISO 3166-1 numeric
postcode	integer	no	Postcode
city	string	no	City
SaddressString	string	no	Address, full address is filled, or individual address parameters
street	string	no	Street
home	string	no	Home
building	string	no	Building
flat	string	no	Flat/office

Datatable 65. Parameters for cod

Parameter	Type	Required	Description
value	float	no	Amount to be paid
currency	string	no	Currency payable

Datatable 66. Parameters for box

Parameter	Type	Required	Description
size	array	yes	Dimensions
weightBruto	float	yes	Weight, in grams, with type = 2 ranging from 1 to 15000. With type = 1, check for the maximum weight of the point
items	array	yes	Goods item data

Datatable 67. Parameters for size

Parameter	Type	Required	Description
x	float	yes	Length, cm
y	float	yes	Width, cm
z	float	yes	Height, cm

v	float	yes	Volume, cc
---	-------	-----	------------

*Datatable 68. Parameters for items*

Parameter	Type	Required	Description
sku	string	yes	SKU
brand	string	yes	Brand / Manufacturer
name	string	yes	Name
quantity	integer	yes	Quantity
price	float	yes	Unit Price
currency	string	yes	Currency
weight	float	no	Weight, grams
descrRu	string	no	Description of goods in Russian, the presence of a description in one of the parameters descr is checked
descrEn	string	no	Description of goods in English, the presence of a description in one of the parameters descr is checked
descrAlt	string	no	Description of goods in the seller's language, the presence of a description in one of the parameters descr
webLink	string	yes	Link to goods on the Internet
webPhoto	string	no	Link to goods image
country	integer	no	Country of manufacture, ISO 3166-1 numeric
notiNum	string	no	Notification Number
notiDate	string	no	Notification end date, dd.mm.yyyy
notiCode	integer	no	Notification Code
sku	string	yes	SKU

*Datatable 69. Output parameters*

Parameter	Type	Required	Description
track	string	yes	Order ID
orderNum	string	yes	Order number
label	string	yes	Label Printing Link
barcode	string	yes	Barcode
error	array	yes	Errors

*Datatable 70. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [UpdateParcel](#).

*Datatable 71. Method UpdateParcel*

<i>JSON</i>	<a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a>
<b>Request method</b> «POST»	<pre> RAW body {   "method": "CreateParcel",   "token": ,   "parcels": [     {       "track": "LKIM0000079924",       "orderNum": "orderNum-1588155275-2",       "countryTo": 643,       "address": {         "postcode": 656039       },       "pointcode": "02561",       "type": 1,       "recipient": {         "phone": "79876543210",         "email": "exampla@example.com",         "fullNameString": "Иванов Иван"       },       "goods_cost_currency": "RUB",       "box": [         {           "size": {             "x": "54.2",             "y": 41.9,             "z": 37.1           },           "weightBruto": 380,           "items": [             {               "sku": "#sku-2746",               "brand": "Samsung",               "name": "SyncMaster 943",               "currency": "RUB",               "quantity": 1,               "price": 5199.99,               "webLink": "http://boxberry.ru",               "descrEn": "Lorem ipsum"             }           ]         }       ]     },     {       "track": "LKIM0000079925",       "orderNum": "orderNum-1588155275-14",       "countryTo": 643,       "address": {         "postcode": 656039       },     }   ] } </pre>

	<pre> "pointcode": "02561", "token": 123, "type": 1, "recipient": {   "phone": "79876543210",   "email": "exampla@example.com",   "fullNameString": "Иванов Иван" }, "goods_cost_currency": "RUB", "box": [   {     "size": {       "x": "51.2",       "y": "45.7",       "z": "38.9"     },     "weightBruto": 330,     "items": [       {         "sku": "#sku-4699",         "brand": "Samsung",         "name": "SyncMaster 943",         "currency": "RUB",         "quantity": 1,         "price": 5199.99,         "webLink": "http://boxberry.ru",         "descrEn": "Lorem ipsum"       }     ]   } ] </pre>
<p><b>Response example:</b></p>	<pre> {   "result": [     {       "track": "LKIM0000079924",       "orderNum": "orderNum-1588155275-2",       "label":         "https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX1",       "barcode": "LKIM0000079924"     },     {       "track": "LKIM0000079925",       "orderNum": "orderNum-1588155275-14",       "label":         "https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX2",       "barcode": "LKIM0000079925"     }   ],   "error": {     "isError": false   } } </pre>

<p><b>Example for PHP 5.0 or later version</b></p>	<pre>&lt;?php \$ch = curl_init(); curl_setopt(\$ch, CURLOPT_URL, 'https://go.bxb.delivery/json.php'); curl_setopt(\$ch, CURLOPT_POST, true); // \$request - RAW body curl_setopt(\$ch, CURLOPT_POSTFIELDS, \$request); curl_setopt(\$ch, CURLOPT_HTTPHEADER, ['Content-Type:application/json']); curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); \$data = json_decode(curl_exec(\$ch), 1); if (!empty(\$data['errors']['isError'])) {     // A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     // Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [     'result' =&gt; [         [             'track' =&gt; 'LKIM0000079924',             'orderNum' =&gt; 'orderNum-1588155275-2',             'label' =&gt; 'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX1',             'barcode' =&gt; 'LKIM0000079924'         ],         [             'track' =&gt; 'LKIM0000079925',             'orderNum' =&gt; 'orderNum-1588155275-14',             'label' =&gt; 'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX2',             'barcode' =&gt; 'LKIM0000079925'         ]     ],     'error' =&gt; [         'isError' =&gt; false     ]     ]     */ }</pre>
<p><b>SOAP WSDL URL</b></p>	<p><a href="https://go.bxb.delivery/__soap/1c_lc.php">https://go.bxb.delivery/__soap/1c_lc.php</a></p>
<p><b>SOAP Request</b></p>	<pre>&lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/__soap/1c_lc.php"&gt; &lt;soapenv:Header/&gt; &lt;soapenv:Body&gt; &lt;c:UpdateParcel soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;UpdateParcelQuery xsi:type="c:UpdateParcelQuery"&gt; &lt;track xsi:type="xsd:string"?&gt;&lt;/track&gt; &lt;orderNum xsi:type="xsd:string"?&gt;&lt;/orderNum&gt; &lt;barcode xsi:type="xsd:string"?&gt;&lt;/barcode&gt; &lt;countryTo xsi:type="xsd:int"?&gt;&lt;/countryTo&gt; &lt;type xsi:type="xsd:int"?&gt;&lt;/type&gt; &lt;pointcode xsi:type="xsd:string"?&gt;&lt;/pointcode&gt;</pre>

```

<address xsi:type="c:AddressData">
  <postcode xsi:type="xsd:int"?></postcode>
  <city xsi:type="xsd:string"?></city>
  <addressString xsi:type="xsd:string"?></addressString>
  <street xsi:type="xsd:string"?></street>
  <house xsi:type="xsd:string"?></house>
  <building xsi:type="xsd:string"?></building>
  <flat xsi:type="xsd:string"?></flat>
</address>
<recipient xsi:type="c:RecipientData">
  <ownerid xsi:type="xsd:string"?></ownerid>
  <fullNameString xsi:type="xsd:string"?></fullNameString>
  <name xsi:type="xsd:string"?></name>
  <surname xsi:type="xsd:string"?></surname>
  <middlename xsi:type="xsd:string"?></middlename>
  <email xsi:type="xsd:string"?></email>
  <phone xsi:type="xsd:string"?></phone>
  <passport xsi:type="c:PassportData">
    <countryCode xsi:type="xsd:int"?></countryCode>
    <passport xsi:type="xsd:string"?></passport>
    <IssuedWhen xsi:type="xsd:string"?></IssuedWhen>
    <IssuedBy xsi:type="xsd:string"?></IssuedBy>
    <birthDate xsi:type="xsd:string"?></birthDate>
    <inn xsi:type="xsd:string"?></inn>
    <kzId xsi:type="xsd:string"?></kzId>
    <kzIdNum xsi:type="xsd:string"?></kzIdNum>
  </passport>
  <regAddress xsi:type="c:RegAddressData">
    <countryCode xsi:type="xsd:int"?></countryCode>
    <postcode xsi:type="xsd:int"?></postcode>
    <city xsi:type="xsd:string"?></city>
    <regAddressString xsi:type="xsd:string"?></regAddressString>
    <street xsi:type="xsd:string"?></street>
    <house xsi:type="xsd:string"?></house>
    <building xsi:type="xsd:string"?></building>
    <flat xsi:type="xsd:string"?></flat>
  </regAddress>
</recipient>
<cod xsi:type="c:CodData">
  <value xsi:type="xsd:double"?></value>
  <currency xsi:type="xsd:string"?></currency>
</cod>
<box xsi:type="c:ArrayOfBoxData">
  <item xsi:type="c:BoxData">
    <size xsi:type="c:SizeData">
      <x xsi:type="xsd:int"?></x>
      <y xsi:type="xsd:int"?></y>
      <z xsi:type="xsd:int"?></z>
    </size>
    <weightBruto xsi:type="xsd:int"?></weightBruto>
    <items xsi:type="c:ArrayOfGoodsData">
      <item xsi:type="c:GoodsData">
        <customsCode xsi:type="xsd:string"?></customsCode>
        <sku xsi:type="xsd:string"?></sku>
        <brand xsi:type="xsd:string"?></brand>
        <name xsi:type="xsd:string"?></name>
      </item>
    </items>
  </item>
</box>

```



	<pre> &lt;quantity xsi:type="xsd:string"?&gt;&lt;/quantity&gt; &lt;price xsi:type="xsd:string"?&gt;&lt;/price&gt; &lt;currency xsi:type="xsd:string"?&gt;&lt;/currency&gt; &lt;weight xsi:type="xsd:string"?&gt;&lt;/weight&gt; &lt;descrRu xsi:type="xsd:string"?&gt;&lt;/descrRu&gt; &lt;descrEn xsi:type="xsd:string"?&gt;&lt;/descrEn&gt; &lt;descrAlt xsi:type="xsd:string"?&gt;&lt;/descrAlt&gt; &lt;id xsi:type="xsd:string"?&gt;&lt;/id&gt; &lt;webLink xsi:type="xsd:string"?&gt;&lt;/webLink&gt; &lt;webPhoto xsi:type="xsd:string"?&gt;&lt;/webPhoto&gt; &lt;country xsi:type="xsd:int"?&gt;&lt;/country&gt; &lt;notiNum xsi:type="xsd:int"?&gt;&lt;/notiNum&gt; &lt;notiDate xsi:type="xsd:string"?&gt;&lt;/notiDate&gt; &lt;notiCode xsi:type="xsd:int"?&gt;&lt;/notiCode&gt; &lt;/item&gt; &lt;/items&gt; &lt;/item&gt; &lt;/box&gt; &lt;token xsi:type="xsd:string"&gt;&lt;/token&gt; &lt;/UpdateParcelQuery&gt; &lt;/c:UpdateParcel&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<p><b>SOAP Response</b></p>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;SOAP-ENV:Envelope xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/__soap/1c_lc.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt; &lt;SOAP-ENV:Body&gt; &lt;ns1:CreateParcelResponse&gt; &lt;return xsi:type="ns1:CreateParcelData"&gt; &lt;track xsi:type="xsd:string"&gt;track_00022&lt;/track&gt; &lt;orderNum xsi:type="xsd:string"&gt;orderNum_0004102&lt;/orderNum&gt; &lt;label xsi:type="xsd:string"&gt;https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id =XXX&lt;/label&gt; &lt;barcode xsi:type="xsd:string"&gt;barcode_123456789&lt;/barcode&gt; &lt;/return&gt; &lt;/ns1:CreateParcelResponse&gt; &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_lc.php', [ 'cache_wsdl' =&gt; WSDL_CACHE_NONE, 'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]);  \$request = [ 'token' =&gt; "", ]; </pre>

```
try {
    $result = $client->UpdateParcel($request);
} catch (\SoapFault $e) {
    //Request execution error
}

if (isset($e)) {
    print_r($e);
} else {
    print_r($result);
    /*
    [
        'result' => [
            [
                'track' => 'LKIM0000079924',
                'orderNum' => 'orderNum-1588155275-2',
                'label' =>
                'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX1',
                'barcode' => 'LKIM0000079924'
            ],
            [
                'track' => 'LKIM0000079925',
                'orderNum' => 'orderNum-1588155275-14',
                'label' =>
                'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX2',
                'barcode' => 'LKIM0000079925'
            ]
        ],
        'error' => [
            'isError' => false
        ]
    ]
    */
}
```

## ManifestControl

Description of API method parameters (for JSON):

*Datatable 72. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	ManifestControl
act	string	yes	Valid values: create (create a manifest and add parcels) / delete (delete parcels from the manifest) / finish (finish the manifest and create a prealert) / view (view the parcels in the manifest)
id	string	yes	Номер манифета, обязателен при delete/finish
parcels	array	yes	One of the box / track parameters is required with act = create / delete
prealert	array	yes	Required with act = finish

*Datatable 73. Parameters for parcel*

Parameter	Type	Required	Description
box	string	no	Comma-separated list of barcodes for manifest
track	string	no	A list of track codes for parcels separated by a comma to form a manifest

*Datatable 74. Parameters for prealert*

Parameter	Type	Required	Description
number	string	no	Number of transport document
transportType	string	no	avia/avto/more/zd
from	string	no	Warehouse code for pickup service, or country code according to ISO 3166-1 numeric
to	string	no	Reception point code
date	string	no	Date of cargo pickup dd.mm.yyyy
comment	string	no	Comment

*Datatable 75. Output parameters*

Parameter	Type	Required	Description
id	string	yes	Manifest number
track	string	yes	Comma-separated track numbers
barcode	string	yes	Comma-separated barcodes
error	array	yes	Errors

*Datatable 76. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method [ManifestControl](#).

*Datatable 77. Method ManifestControl*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a>
<b>Request method</b> «POST»	POST token= &method=ManifestControl&id=id(act=delete/finish) &act=create/delete/finish/view&parcels[track]=track1, track2&prealert[number]=N001&prealert[transportType]= &prealert[from]=&prealert[to]=&prealert[comment]=
<b>Response example:</b>	{ "token": "", "method": "ManifestControl", "id": "id (act=delete/finish/view)", "act": "create/delete/finish/view", "parcels": { "box": "barcode1,barcode2", "track": "track1,track2" } "prealert": { "number": "N001", "transportType": "", "from": "", "to": "", "comment": "", } }
<b>Example for PHP 5.0 or later version</b>	<?php \$ch = curl_init(); curl_setopt(\$ch, CURLOPT_URL, 'https://go.bxb.delivery/json.php'); curl_setopt(\$ch, CURLOPT_POST, true); // \$request - RAW body curl_setopt(\$ch, CURLOPT_POSTFIELDS, \$request); curl_setopt(\$ch, CURLOPT_HTTPHEADER, ['Content-Type:application/json']); curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); \$data = json_decode(curl_exec(\$ch), 1); if (!empty(\$data['errors']['isError'])) { // A request is made with an error echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else { // Everything is fine, the answer is received, data in \$data: /* print_r(\$data); [ 'result' => [ 'id' => 'XXXXXXXXXXXX', 'track' => 'track_00022', 'barcode' => 'barcode_123456789', 'label' => 'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX', 'error' => [ 'isError' => false ] ] ] */ }
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_lc.php">https://go.bxb.delivery/_soap/1c_lc.php</a>
<b>SOAP Request</b>	<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"

	<pre> xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/__soap/1c_lc.php"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;c:ManifestControl soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;       &lt;ManifestControlQuery xsi:type="c:ManifestControlQuery"&gt;         &lt;act xsi:type="xsd:string"&gt;create/delete/finish/view&lt;/act&gt;         &lt;id xsi:type="xsd:string"&gt;id (act=delete/finish)&lt;/id&gt;         &lt;parcels xsi:type="c:ParcelData"&gt;           &lt;box xsi:type="xsd:string"&gt;barcode1,barcode2&lt;/box&gt;           &lt;track xsi:type="xsd:string"&gt;track1,track2&lt;/track&gt;         &lt;/parcels&gt;         &lt;prealert xsi:type="c:PrealertData"&gt;           &lt;number xsi:type="xsd:string"&gt;N001&lt;/number&gt;           &lt;transportType xsi:type="xsd:string"&gt;&lt;/transportType&gt;           &lt;from xsi:type="xsd:string"&gt;?&lt;/from&gt;           &lt;to xsi:type="xsd:string"&gt;?&lt;/to&gt;           &lt;date xsi:type="xsd:string"&gt;?&lt;/date&gt;           &lt;comment xsi:type="xsd:string"&gt;?&lt;/comment&gt;         &lt;/prealert&gt;         &lt;token xsi:type="xsd:string"&gt;?&lt;/token&gt;       &lt;/c:ManifestControl&gt;     &lt;/soapenv:Body&gt;   &lt;/soapenv:Envelope&gt; </pre>
<b>SOAP Response</b>	-
<b>Example for PHP 5.0 or later version</b>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_lc.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]); \$request = [   'token' =&gt; "", ]; try {   \$result = \$client-&gt;ManifestControl(\$request); } catch (\SoapFault \$e) {   //Request execution error if (isset(\$e)) {   print_r(\$e); } else {   print_r(\$result); } /* [  'result' =&gt; [   'id' =&gt; 'XXXXXXXXXXXX',   'track' =&gt; 'track_00022',   'barcode' =&gt; 'barcode_123456789',   'label' =&gt; 'https://go.bxb.delivery/personaloffice/export/parcel/?parcel_id=XXX',   'error' =&gt; [     'isError' =&gt; false    ]  ] ] */ </pre>

## WarehouseControl

Description of API method parameters (for JSON):

*Datatable 78. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	WarehouseControl
act	string	yes	Valid values: create (create a warehouse) / edit (edit data of a warehouse previously created for a given token) / delete (delete a previously created warehouse) / view (view data on warehouses)
id	string	yes	Warehouse number to change / view / delete
warehouse	array	yes	Massive with data about country and address where located warehouse

*Datatable 79. Parameters for massive warehouse*

Parameter	Type	Required	Description
country	string	yes	Country, ISO 3166-1 numeric
address	array	yes	Обязательно заполнение адреса строкой или отдельных параметров

*Datatable 80. Parameters for address*

Parameter	Type	Required	Description
addressString	string	no	Address
zipcode	integer	no	Postcode
city	string	no	City
street	string	no	Street
house	string	no	Home
building	string	no	Building
room	string	no	Flat/office
contactName	string	yes	Contact person
contactPhone	string	yes	Contact number
comment	string	no	Comment

*Datatable 81. Output parameters*

Parameter	Type	Required	Description
id	string	yes	Warehouse number to change / view / delete
deleted	bool	no	1/0 sign of successful warehouse delete. Returned only with act=delete
edited	bool	no	1/0 sign of a successful warehouse change. Returned only with act=edit
country	string	no	returned only when act = view
address	array	no	returned only when act = view
error	array	yes	errors

*Datatable 82. Parameters for error*

<b>Parameter</b>	<b>Type</b>	<b>Required</b>	<b>Description</b>
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description
isError	bool	yes	0 - no errors, 1 - there is an error

Example request and response for method [WarehouseControl](#).

*Datatable 83. Method WarehouseControl*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a>
<b>Request method</b> «POST»	<p>POST</p> <p>token=&amp;method=WarehouseControl&amp;id=id (act=edit/delete/view)&amp;act=create/delete/finish/view&amp; warehouse[country]=643&amp; warehouse[address][addressString]=ул. Ломоносова, 2 кор.1&amp; warehouse[address][zipcode]=111111&amp; warehouse[address][city]=Москва&amp; warehouse[address][street]=ул. Ломоносова&amp; warehouse[address][house]=2&amp; warehouse[address][building]=1&amp; warehouse[address][room]=&amp; warehouse[address][contactName]=Сидорова Алёна&amp; warehouse[address][contactPhone]=558557&amp; warehouse[address][comment]=</p> <p>RAW body</p> <pre>{   "token": "",   "method": "WarehouseControl",   "id": "id (act=edit/delete/view)",   "act": "create/delete/finish/view",   "warehouse": {     "country": "643",     "address": {       "addressString": "ул. Ломоносова, 2 кор.1",       "zipcode": "111111",       "city": "Москва",       "street": "ул. Ломоносова",       "house": "2",       "building": "1",       "room": "",       "contactName": "Сидорова Алёна",       "contactPhone": "558557",       "comment": ""     }   } }</pre>
<b>Response example:</b>	<pre>{   "country": 643,   "address": {     "addressString": "111111, Москва, Королева, 12",     "city": "Москва",     "zipcode": "111111",     "street": "Королева",     "house": "12",     "building": "",     "room": "",     "contactName": "Иванов Сергей",     "contactPhone": "555-888",     "comment": ""   } }</pre>
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$ch = curl_init(); curl_setopt(\$ch, CURLOPT_URL, 'https://go.bxb.delivery/json.php');</pre>



	<pre> curl_setopt(\$ch, CURLOPT_POST, true); //\$request - RAW body curl_setopt(\$ch, CURLOPT_POSTFIELDS, \$request); curl_setopt(\$ch, CURLOPT_HTTPHEADER, ['Content-Type:application/json']); curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); \$data = json_decode(curl_exec(\$ch), 1); if (!empty(\$data['errors']['isError'])) {     //A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     //Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [     'country' =&gt; 643,     'address' =&gt; [     'addressString' =&gt; '111111, Москва, Королева, 12',     'city' =&gt; 'Москва',     'zipcode' =&gt; '111111',     'street' =&gt; 'Королева',     'house' =&gt; '12',     'building' =&gt; '',     'room' =&gt; '',     'contactName' =&gt; 'Иванов Сергей',     'contactPhone' =&gt; '555-888',     'comment' =&gt; ''     ]     ]     */ } </pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_lc.php">https://go.bxb.delivery/_soap/1c_lc.php</a>
<b>SOAP Request</b>	<pre> &lt;soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_lc.php"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;c:WarehouseControl soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;       &lt;WarehouseControlQuery xsi:type="c:WarehouseControlQuery"&gt;         &lt;act xsi:type="xsd:string"&gt;create/edit/delete/view&lt;/act&gt;         &lt;id xsi:type="xsd:int"&gt;?&lt;/id&gt;         &lt;warehouse xsi:type="c:WarehouseData"&gt;           &lt;country xsi:type="xsd:string"&gt;643&lt;/country&gt;           &lt;address xsi:type="c:WarehouseAddressData"&gt;             &lt;addressString xsi:type="xsd:string"&gt;ул. Ломоносова, 2 кор. 1&lt;/addressString&gt;             &lt;zipcode xsi:type="xsd:int"&gt;111111&lt;/zipcode&gt;             &lt;city xsi:type="xsd:string"&gt;Москва&lt;/city&gt;             &lt;street xsi:type="xsd:string"&gt;ул. Ломоносова&lt;/street&gt;             &lt;house xsi:type="xsd:string"&gt;2&lt;/house&gt;             &lt;building xsi:type="xsd:string"&gt;1&lt;/building&gt;             &lt;room xsi:type="xsd:string"&gt;&lt;/room&gt;             &lt;contactName xsi:type="xsd:string"&gt;Сидорова Алёна&lt;/contactName&gt;             &lt;contactPhone xsi:type="xsd:string"&gt;558557&lt;/contactPhone&gt; </pre>

	<pre> &lt;comment xsi:type="xsd:string"&gt;&lt;/comment&gt;   &lt;/address&gt; &lt;/warehouse&gt;   &lt;token xsi:type="xsd:string"&gt;?&lt;/token&gt; &lt;/WarehouseControlQuery&gt; &lt;/c:WarehouseControl&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<b>SOAP Response</b>	-
<b>Example for PHP 5.0 or later version</b>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_lc.php', [     'cache_wsdl' =&gt; WSDL_CACHE_NONE,     'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]);  \$request = [     'token' =&gt; "", ];  try {     \$result = \$client-&gt;WarehouseControl(\$request); } catch (\SoapFault \$e) {     //Request execution error } if (isset(\$e)) {     print_r(\$e); } else {     print_r(\$result);     /*     [         'country' =&gt; 643,         'address' =&gt; [             'addressString' =&gt; '111111, Москва, Королева, 12',             'city' =&gt; 'Москва',             'zipcode' =&gt; '111111',             'street' =&gt; 'Королева',             'house' =&gt; '12',             'building' =&gt; "",             'room' =&gt; "",             'contactName' =&gt; 'Иванов Сергей',             'contactPhone' =&gt; '555-888',             'comment' =&gt; ""         ]     ]     */ } </pre>

## ControlUser

Description of API method parameters (for JSON):

*Datatable 84. Input parameters*

Parameter	Type	Required	Description
token	string	yes	Your API token
method	string	yes	ControlUser
act	string	yes	Valid Values: create (create) / edit (edit)
id	string	yes	User ID, required with act = edit
name	string	no	Full name or any name for contact
phone	string	no	Contact number
email	string	no	Contact email
access	bool	no	1 = access open, 0 = access denied
setRole	string	no	Set user role. Possible values: employee, manager

*Datatable 85. Output parameters*

Parameter	Type	Required	Description
id	integer	yes	User Id
edited	bool	no	1/0 attribute for successful user editing. Returns only when act=edit
error	array	yes	Errors

*Datatable 86. Parameters for error*

Parameter	Type	Required	Description
isError	bool	yes	0 - no errors, 1 - there is an error
errorCode	string	no	Error code
errorMessage	string	no	Error description

Example request and response for method `ControlUser`.

*Datatable 87. Method ControlUser*

<b>JSON</b>	<a href="https://go.bxb.delivery/json.php">https://go.bxb.delivery/json.php</a>
<b>Request method</b> «POST»	<p>POST</p> <p>token=&amp;method=ControlUser&amp;id=user_id (act=edit)&amp;act=create/edit&amp;name=Ivanov Petr&amp;email=i.petr@company.com&amp;phone=353535&amp;access=1&amp;setRole=employee/chief</p> <p>RAW body</p> <pre>{   "token": "",   "method": "ControlUser",   "id": "user_id (act=edit)",   "act": "create/edit",   "name": "Ivanov Petr",   "email": "i.petr@company.com",   "phone": "353535",   "access": "1",   "setRole": "employee/chief" }</pre>
<b>Response example:</b>	-
<b>Example for PHP 5.0 or later version</b>	<pre>&lt;?php \$ch = curl_init(); curl_setopt(\$ch, CURLOPT_URL, 'https://go.bxb.delivery/json.php'); curl_setopt(\$ch, CURLOPT_POST, true); // \$request - RAW body curl_setopt(\$ch, CURLOPT_POSTFIELDS, \$request); curl_setopt(\$ch, CURLOPT_HTTPHEADER, ['Content-Type:application/json']); curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); \$data = json_decode(curl_exec(\$ch), 1); if (!empty(\$data['errors']['isError'])) {     // A request is made with an error     echo \$data['errors']['errorCode'].': '.\$data['errors']['errorMessage']; } else {     // Everything is fine, the answer is received, data in \$data:     /*     print_r(\$data);     [     ]     */ }</pre>
<b>SOAP WSDL URL</b>	<a href="https://go.bxb.delivery/_soap/1c_lc.php">https://go.bxb.delivery/_soap/1c_lc.php</a>
<b>SOAP Request</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:c="https://go.bxb.delivery/_soap/1c_lc.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;soapenv:Header /&gt;   &lt;soapenv:Body&gt;     &lt;c:ControlUser soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"&gt;</pre>

	<pre> &lt;ControlUserQuery xsi:type="c:ControlUserQuery"&gt;   &lt;id xsi:type="xsd:int"&gt;user_id (act=edit)&lt;/id&gt;   &lt;act xsi:type="xsd:string"&gt;create/edit&lt;/act&gt;   &lt;name xsi:type="xsd:string"&gt;Ivanov Petr&lt;/name&gt;   &lt;email xsi:type="xsd:string"&gt;i.petr@company.com&lt;/email&gt;   &lt;phone xsi:type="xsd:string"&gt;353535&lt;/phone&gt;   &lt;access xsi:type="xsd:boolean"&gt;1&lt;/access&gt;   &lt;setRole xsi:type="xsd:string"&gt;employee/chief&lt;/setRole&gt;   &lt;token xsi:type="xsd:string"&gt;&lt;/token&gt; &lt;/ControlUserQuery&gt; &lt;/c:ControlUser&gt; &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>
<p><b>SOAP Response</b></p>	<pre> &lt;SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https://go.bxb.delivery/__soap/1c_lc.php" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"&gt;   &lt;SOAP-ENV:Body&gt;     &lt;ns1:ControlUserResponse&gt;       &lt;return xsi:type="ns1:ControlUserData"&gt;         &lt;id xsi:type="xsd:int"&gt;15502&lt;/id&gt;         &lt;edited xsi:type="xsd:boolean"&gt;&gt;false&lt;/edited&gt;       &lt;/return&gt;     &lt;/ns1:ControlUserResponse&gt;   &lt;/SOAP-ENV:Body&gt; &lt;/SOAP-ENV:Envelope&gt; </pre>
<p><b>Example for PHP 5.0 or later version</b></p>	<pre> &lt;?php \$client = new \SoapClient('https://go.bxb.delivery/__soap/1c_lc.php', [   'cache_wsdl' =&gt; WSDL_CACHE_NONE,   'features' =&gt; SOAP_SINGLE_ELEMENT_ARRAYS, ]); \$request = [   'token' =&gt; "", ]; try {   \$result = \$client-&gt;ControlUser(\$request); } catch (\SoapFault \$e) {   //Request execution error } if (isset(\$e)) {   print_r(\$e); } else {   print_r(\$result);   /*   [   ]   */ } </pre>

## List of changes

*Datatable 88. List of changes*

Date	Description of change
05.08.2020	«DeleteParcel» added on page 46