



Description of web-services

<https://account.boxberry.ru>

<https://api.boxberry.ru>

Release 1.6 dated 11.10.2019

Contents

Modifications	3
ChangeLog API PA IS. Changes in API services after new PA log in	4
Introduction	5
1. Terms and abbreviations to be used	5
2. Integration configuring.....	6
3. API-services for receiving data	7
3.1. ListCities. List of cities of delivery	9
3.2. ListCitiesFull. Full list of cities of delivery	10
3.3. ListPoints. Pick-up points list.	12
3.4. ListPointsShort. A short list of pick-up points.	15
3.5. ListZips. A list of zipcodes for courier delivery service.	17
3.6. ZipCheck. A zipcode check.....	19
3.7. ListStatuses. Parcel shipment status.....	21
3.8. ListStatusesFull. Full list of order statuses.	23
3.9. ListServices. Cost of serviced charged.	26
3.10. CourierListCities. List of cities with courier delivery service.	28
3.11. DeliveryCosts. Delivery cost calculator.....	30
3.12. PointsForParcels. The list of parcel depositaries.	32
3.13. PointsDescription. Extended information about PUP.	34
4. API-services for transferring data.....	40
4.1. ParselCreate. Create/upload an order.....	41
4.2. Restrictions on creating a shipment.....	48
4.3. Labelling requirements for the parcel.	52
4.4. ParselSend. Act of acceptance of delivery formation.....	53
4.5. ParselSend. Error reference.	55
4.6. ParselDel. Delete the order.....	56
4.7. CreateIntake. A request order to issue the parcels by a courier.	58
4.8. ParselCheck. Link to a label print form.	60
4.9. ParselList. List of tracking codes.	62
4.10. ParselStory.....	64
4.11. ParselSendStory	66
4.12. OrdersBalance	68
5. Frequently asked questions	70
6. Additionally.....	71

Modifications

Documentation release	Date	Modifications
1.2	02.07.2019	ParseCreate method. Parameters added: <ul style="list-style-type: none"> • 'x', 'y', 'z' - packaging unit dimensions. For sending by Russian Post. • 'issue' – type of issuance. A delivery type of purchase order can be sent for each order. • sender_name - a name of the store-sender for sms/e-mail notifications and home marking of Boxberry.
		ParseCreate method. Formatting data changes of parameter 'delivery_date'. Expected date format - YYYY-MM-DD
1.3	05.07.2019	ListZips and ZipCheck methods. Parameter added: <ul style="list-style-type: none"> • Remoteness - zipcode remoteness.
1.4	06.08.2019	Reference guide for ParseSend method is added.
1.5	02.09.2019	ChangeLog API PA IS is added.
1.6	11.10.2019	A description of 'sucrh' parameter of DeliveryCosts method is added.

ChangeLog API PA IS. Changes in API services after new PA log in

- Switching to https-protocol.
- New URL for integration - https://api.boxberry.ru
- Token – after new PA log in a new token is generated.
- A token format is changed, e.g. d6f33e419c16131e5325cbd84d5d6459
- Address validation service is switched from iqdq.ru to DaData.ru

Dramatic changes of methods:

Method	Modifications
ParselCreate	<ul style="list-style-type: none"> • 'fio' (recipient's name) - shall be written in Cyrillic. • 'delivery_date' - formatting data changes of parameter. Expected date format - YYYY-MM-DD. <p>Parameters added:</p> <ul style="list-style-type: none"> • 'x', 'y', 'z' - packaging unit dimensions. For sending by Russian Post. • 'issue' – type of issuance. A delivery type of purchase order can be sent for each order. • 'sender_name' - a name of the store-sender for sms/e-mail notifications and home marking of Boxberry. • 'vid': 3 - a new delivery type (Russian Post). <p>Mandatory parameters have been changed:</p> <ul style="list-style-type: none"> • 'price', 'payment_sum', 'delivery_sum' parameters are not mandatory. • 'x', 'y', 'z' parameters for weights are added to describe parcel dimensions (these are mandatory only for Russian Post delivery). • 'name' and 'inn' parameters are mandatory for legal person.
ListZips	Remoteness parameter has been added to describe the zipcode remoteness.
ZipCheck	Remoteness parameter has been added to describe the zipcode remoteness.

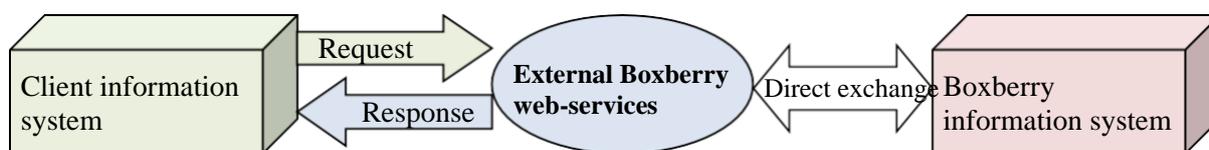
Introduction

Boxberry delivery service provides a solution for its information system integration with information systems of online stores / partners. Such integration would allow for the exchange of necessary data on-line, eliminating manual labor.

Interaction with our web services via standard HTTP and HTTPS protocols and SOAP and JSON protocols provides the ease of use and quick access to the basic functions of the Boxberry information system.

You can receive up-to-date information about delivery cities, pick-up points, delivery times, availability of courier delivery at the recipient's zipcode, services charged, as well as tracking the shipment and sending shipment data to our information system.

Basic interaction principle:



Admin panel:

<https://account.boxberry.ru>

Web-services location:

JSON: <https://api.boxberry.ru/json.php>

WSDL document for work with SOAP protocol:

https://api.boxberry.ru/soap/1c_1c.php?wsdl – for dealing with parcels.

https://api.boxberry.ru/soap/1c_public.php?wsdl – for receiving information.

1. Terms and abbreviations to be used

Table 1. Terms and abbreviations to be used

Abbreviation	Definition
IS	Internet store / partner, entering into a contract with Boxberry for deliveries
PO	Purchase order
PUP	Pick-up point
CD	Courier delivery
RP	Russian Post
IS	Information system
PA	Personal account
DaData.ru	Service of automatic verification and correction of contact data (names, addresses, phone numbers, emails, passports).

2. Integration configuring

Main configuring stages of Integration with Boxberry web-services:

1. Study this document.
2. Get acquainted with Boxberry web-developments (API-services, predefined units, widgets) and choose the suitable IT-solution.
3. Develop a program / script for interacting with web-services.
4. Get a unique key (API-token) for access to Boxberry web-services.
5. Customise your software using web services.
6. It is necessary to conduct beta-testing before using of API-services for shipment data about Boxberry Database in the standard operating mode. Boxberry IT-technicians shall provide the data for beta-testing.

3. API-services for receiving data

This section lists the API services for receiving information from the Boxberry system, using which you can get information about delivery cities, pick-up points, shipment status, services charged etc. (it is advised to cache the data received via these services):

1. **ListCities** - allows you to get the list of cities with Boxberry pick-up points.
2. **ListCitiesFull** - allows you to get the list of cities with Boxberry delivery service.
3. **ListPoints** - allows you to get information about all pick-up points. When using an additional parameter.
4. **ListPointsShort** - allows you to get codes of all pick-up points or the points of a city when "CityCode" specified and also the date of the latest change.
5. **ListZips** - allows you to get the list of zipcodes with courier delivery available.
6. **ZipCheck** - allows you to get information about courier delivery service available for the specified zipcode.
7. **ListStatuses** - allows you to get information about parcel shipment status. Updating is enough
8. **ListStatusesFull** - allows you to get extended information about shipment status, including items issued and returned in case of partial issuance, actual weight of the purchase, money received and charge type.
9. **ListServices** - allows you to get the list of delivery services provided and their cost, the services are charged upon delivery.
10. **CourierListCities** - allows you to get the list of cities with courier delivery service.
11. **DeliveryCosts** - allows you to get a quote of delivery to a pick-up point, including constant service cost provided by the contract, it is possible to pay taking courier delivery into consideration. It returns the base cost, service cost (notifications, issuance, insurance charge, payment acceptance, courier delivery etc.), delivery period taking the type of issuance into consideration (at a pick-up point or via courier delivery service).
12. **PointsForParcels** - allows you to get the list of Boxberry parcel depositaries.

13. **PointsDescription** - allows you to get the exact information about a specific pick-up point, including photos.

3.1. ListCities. List of cities of delivery

Allows you to get the list of cities with Boxberry pick-up points.
Refresh rate recommended: once a day.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
CountryCode	No	string	Country code

Response data:

Parameter	Type	Description
Name	string	Name of the city
Code	string	Boxberry city code
CountryCode	string	Country code
Prefix	string	Prefix: City, Village etc.
ReceptionLaP	integer	Receipt of letters and parcels from a physical person (0/1)
DeliveryLaP	integer	Issuance of letters and parcels from a physical person (0/1)
Reception	integer	Receipt of IS orders at pick-up points (0/1)
ForeignReceptionReturns	integer	Foreign reception returns (0/1)
Terminal	integer	Terminal availability in the city (0/1)
Kladr	string	Kladr ID
Region	string	Region
UniqName	string	Compound unique name (for cities having a non-unique name: city area + district)
District	string	District
CourierReception	integer	Courier reception availability (0/1)

Table 2. ListCities method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ListCities</p> <p>Implementing example:</p> <pre> \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ListCities'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // a list of all cities with pick-up points as: /* \$data[0...n]=array('Name'=>'Name of the city', 'Code'=>'Boxberry city code', 'Prefix' => 'Prefix: City, Village etc. 'ReceptionLaP' => 'Receipt of letters and parcels from a physical person (0/1) ', 'DeliveryLaP' => 'Issuance of letters and parcels from a physical person (0/1)', 'Reception' => 'Receipt of IS orders at pick-up points (0/1)', */ </pre>

	<pre> 'ForeignReceptionReturns' => 'Foreign reception returns (0/1)', 'Terminal' => 'Terminal availability in the city (0/1)', 'Kladr' => 'Kladr ID', 'Region' => 'Region', 'CountryCode' => 'Country code', 'UniqName' => 'Compound unique name', 'District' => 'District', 'CourierReception' => 'Courier reception availability (0/1)'); e.g. echo \$data[0]['Name']; echo \$data[5]['Code']; */ } </pre>
SOAP	<p>Implementing example:</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/lc_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$s=array(); \$\$['token']='YourToken'; try {\$data=\$clientS->ListCities(\$s);} catch (SoapFault \$exception){\$errS=\$exception;} if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { print_r(\$data); } </pre>

3.2. ListCitiesFull. Full list of cities of delivery

It allows you to get the list of cities with Boxberry delivery service.
Refresh rate recommended: once a day.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
CountryCode	No	string	Country code

Response data:

Parameter	Type	Description
Name	string	Name of the city
Code	string	Boxberry city code
CountryCode	string	Country code
Prefix	string	Prefix: City, Village etc.
ReceptionLaP	integer	Receipt of letters and parcels from a physical person (0/1)
DeliveryLaP	integer	Issuance of letters and parcels from a physical person (0/1)
Reception	integer	Receipt of IS orders at pick-up points (0/1)
ForeignReceptionReturns	integer	Foreign reception returns (0/1)
Terminal	integer	Terminal availability in the city (0/1)
Kladr	string	Kladr ID
PickupPoint	integer	Pick-up points available in the city (0/1)

CourierDelivery	integer	Courier delivery available in the city
Region	string	Region
UniqName	string	Compound unique name (for cities having a non-unique name: city area + district)
District	string	District
CourierReception	integer	Courier reception availability (0/1)

Table 3. ListCitiesFull method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ListCitiesFull</p> <p>Implementing example:</p> <pre> \$url="https://api.boxberry.ru/json.php?token=YourToken&method=ListCitiesFull"; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // a list of all cities with pick-up points as: /* \$data[0..n]=array('Name'=> 'Name of the city', 'Code'=> 'Boxberry city code', 'Prefix' => 'Prefix: City, Village etc. 'ReceptionLaP' => 'Receipt of letters and parcels from a physical person (0/1)', 'DeliveryLaP' => 'Issuance of letters and parcels from a physical person (0/1)', 'Reception' => 'Receipt of IS orders at pick-up points (0/1)', 'ForeignReceptionReturns' => 'Foreign reception returns', 'Terminal' => 'Terminal availability', 'Kladr' => 'Kladr ID', 'Region' => 'Region', 'CountryCode' => 'Country code', 'UniqName' => 'Compound unique name', 'District' => 'District', 'PickupPoint' => ' Pick-up points available in the city (0/1)', 'CourierDelivery' => ' Courier delivery available in the city ', 'CourierReception' => ' Courier reception availability (0/1)'); e.g. echo \$data[0]['Name']; echo \$data[5]['Code']; */ } </pre>
SOAP	<p>Implementing example:</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$\$=array(); \$\$['token']='YourToken'; try { \$data=\$clientS->ListCities(\$\$); } catch (SoapFault \$exception){ \$error=\$exception; } </pre>

	<pre> if(isset(\$errS)){// if something went wrong and the response has not been received. echo \$errS; } else { // everything is correct, the response has been received and sent to \$data array. print_r(\$data); } </pre>
--	---

3.3. ListPoints. Pick-up points list.

It allows you to get information about all pick-up points.
Refresh rate recommended: once an hour.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
CityCode	No	string	Boxberry city code (get via ListCities method)
prepaid	No	boolean	Payment options at a pick-up point while issuing: 0 - return to a list of departments having the possibility of payment upon receipt of the order (OnlyPrepaidOrders=No) 1 - all departments working with any type of payment.

Response data:

Parameter	Type	Description
Code	string	Pick-up point code on a Boxberry database
Name	string	Pick-up point name
Address	string	Full address
Phone	string	Phone number(s)
WorkSchedule	string	Schedule of work
TripDescription	string	Directions
DeliveryPeriod	string	Delivery period (days, default delivery period from Moscow)
CityCode	string	Boxberry city code
CityName	string	Name of the city
TariffZone	string	Tariff zone (default zone for a departure city - Moscow)
Settlement	string	Name of the settlement
Area	string	Area
Country	string	Name of the country
GPS	string	GPS coordinates
AddressReduce	string	Abbreviated address (street, house, flat)
OnlyPrepaidOrders	string	Issuing of prepaid orders: "Yes" - only issuing without payment acceptance, "No" - issuing of any orders
Acquiring	string	Debit card payment available (Yes/No)
DigitalSignature	string	A tablet for digital signature available: "Yes" - the recipient's signature will be stored in electronic format in the Boxberry system, "No" - no electronic signatures
CountryCode	string	Boxberry country code
NalKD	string	The department provides the courier delivery service (Yes/No)
Metro	string	Metro station
TypeOfOffice	string	Pick-up point type: 1- own department, 2- partner department

VolumeLimit	string	Volume limit, m3
LoadLimit	string	Limit of weight, kg.

Table 4. ListPoints method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ListPoints&prepaid https://api.boxberry.ru/json.php?token=YourToken&method=ListPoints&prepaid=0 – the list of pick-up points providing the possibility to pay when collect the delivery. https://api.boxberry.ru/json.php?token=YourToken&method=ListPoints&prepaid=1 – the list of pick-up points working with any type of orders https://api.boxberry.ru/json.php?token=YourToken&method=ListPoints&CityCode=68 – the list of pick-up points in the city with the code given providing the possibility to pay when collect the delivery https://api.boxberry.ru/json.php?token=YourToken&method=ListPoints&CityCode=68&prepaid=1 – the list of all pick-up points in the city with the code given</p> <p>Implementing example: \$curl='https://api.boxberry.ru/json.php?token=YourToken&method=ListPoints'; \$handle = fopen(\$curl, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // a list of all pick-up points as: /* \$data[0...n]=array('Code'=>'Boxberry city code', 'Name'=>'Name of the pick-up point', 'Address'=>'Full address', 'Phone'=>'Phone number(s)', 'WorkSchedule'=>'Schedule of work', 'TripDescription'=>'Directions', 'DeliveryPeriod'=>'Delivery period (delivery period from Moscow, days)', 'CityCode'=>'Boxberry city code', 'CityName'=>'Name of the city', 'TariffZone'=>'Tariff zone (a departure city - Moscow)', 'Settlement'=>'Name of the settlement', 'Area'=>'Area', 'Country'=>'Country', 'OnlyPrepaidOrders'=>'If "Yes" - a point works with prepaid orders', 'AddressReduce'=>'Abbreviated address', 'Acquiring'=>'If "Yes" - Debit (bank) card payment available ', 'DigitalSignature'=>'If "Yes" - the recipient's signature will be stored in electronic format in the Boxberry system', 'TypeOfOffice'=>'Pick-up point type: 1-PUP, 2-PUPN', 'NaIKD'=>'Courier delivery', 'Metro'=>'Metro station', 'VolumeLimit'=>'Volume limit', 'LoadLimit'=>'Limit of weight, kg', 'GPS'=>'GPS coordinates'); e.g. echo \$data[0]['Name']; echo \$data[5]['Code']; */ }</p>

SOAP	<p>Implementing example:</p> <pre>ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$s=array(); \$s['token']='YourToken'; \$s['code']=0; \$s['prepaid']=0; try {\$data=\$clientS->ListPoints(\$s);} catch (SoapFault \$exception){\$errS=\$exception;} if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { print_r(\$data); }</pre>
------	--

3.4. ListPointsShort. A short list of pick-up points.

It allows you to get codes of all pick-up points or the points of a city when "CityCode" and also the date of the latest change are specified.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
CityCode	No	string	Boxberry city code

Response data:

Parameter	Type	Description
CityCode	string	Boxberry city code
Code	string	Pick-up point code on a Boxberry database
UpdateDate	string	The latest date of changing information about the pick-up point

Table 5. ListPointsShort

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request:</p> <p>The list of all pick-up points https://api.boxberry.ru/json.php?token=YourToken&method=ListPointsShort</p> <p>The list of all pick-up points in the specified city https://api.boxberry.ru/json.php?token=YourToken&method=ListPointsShort&CityCode=68</p> <p>Implementing example:</p> <pre>\$url='https://api.boxberry.ru/json.php?token=YourToken&method=ListPointsShort'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(!empty(\$data[0]['err'])) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // a list of all pick-up points as: /* \$data[0..n]=array('Code'=>'Boxberry city code', 'CityCode'=>'Boxberry city code', 'UpdateDate'=>'The date of the latest change'); e.g. echo \$data[0]['Name']; echo \$data[5]['Code']; */ }</pre>
SOAP	<p>Implementing example:</p> <pre>ini_set("soap.wsdl_cache_enabled", "0"); \$client = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$\$ = array(); \$\$['token'] = 'YourToken'; \$\$['CityCode'] = 0;</pre>

	<pre>try { \$data=\$clientS->ListPointsShort(\$s); } catch (SoapFault \$exception){ \$errS=\$exception; } if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { print_r(\$data); }</pre>
--	--

3.5. ListZips. A list of zipcodes for courier delivery service.

It allows you to get the list of zipcodes with courier delivery service available.
Refresh rate recommended: once a day.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token

Response data:

Parameter	Type	Description
Zipcode array		
Zip	string	Zipcode
City	string	Name of the city
Region	string	District
Area	string	Region
ZoneExpressDelivery	integer	The zone of courier delivery
Remoteness	integer	The zipcode remoteness with values from 0 to 9, where 0 is not considered to be remote.

Table 6. ListZips method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ListZips</p> <p>Implementing example:</p> <pre> \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ListZips'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // a list of all zipcodes as: /* \$data[0..n]=array('Zip'=>'Zipcode', 'City'=>'City', 'Region' => 'Region', 'Area'=>'Area', 'ZoneExpressDelivery'=>'The zone of courier delivery', 'Remoteness'=> 'The zipcode remoteness'); e.g. echo \$data[0]['Zip']; echo \$data[5]['City']; */ } </pre>
SOAP	<p>Implementing example:</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); </pre>

	<pre>\$s=array(); \$s['token']='YourToken'; try {\$data=\$clientS->ListZips(\$s);} catch (SoapFault \$exception){\$errS=\$exception;} if(@\$errS){echo \$errS;} // if something went wrong and the response has not been received. else { print_r(\$data); }</pre>
--	---

3.6. ZipCheck. A zipcode check.

It allows you to get information about courier delivery service available for the specified zipcode.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
zip	Yes	integer	Zipcode
CountryCode	No	string	Country code

Response data:

Parameter	Type	Description
ExpressDelivery	boolean	Courier delivery service available for the specified zipcode (true/false)
ZoneExpressDelivery	integer	The zone of courier delivery
Remoteness	integer	The zipcode remoteness with values from 0 to 9, where 0 is not considered to be remote.

Table 7. ZipCheck method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ZipCheck&Zip=101000</p> <p>Implementing example:</p> <pre>\$url='https://api.boxberry.ru/json.php?token=YourToken&method=ZipCheck&Zip=101000'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // the only element containing data: /* \$data[0]=array('ExpressDelivery'=>'1 - delivery available, 0 - not available', 'ZoneExpressDelivery'=>'Tariff zone', 'Remoteness'=>'Remoteness'); e.g. echo \$data[0]['ExpressDelivery']; */ }</pre>
SOAP	<p>Implementing example:</p> <pre>ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$\$=array(); \$\$['token']='YourToken'; \$\$['zip']=XXXXXX; try {\$data=\$clientS->ZipCheck(\$\$);} catch (SoapFault \$exception){\$errS=\$exception;}</pre>

	<pre>if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { print_r(\$data); }</pre>
--	---

3.7. ListStatuses. Parcel shipment status

Allows you to get information about parcel shipment status. The <order tracking code> parameter is mandatory.

In examples XXXXXX means order tracking code or the IS order number.

Refresh rate recommended: once an hour.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
ImId	Yes	string	Parcel tracking number

Response data:

Parameter	Type	Description
Status array		
Date	string	Designation date as YYYY-MM-DD HH:MM:SS
Name	string	Status name
Comment	string	Comment

Table 8. ListStatuses method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ListStatuses&ImId=XXXXXX</p> <p>Implementing example: <pre>\$url='https://api.boxberry.ru/json.php?token=YourToken&method=ListStatuses&ImId=XXXXXX'; // XXXXXX - tracking code \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // all statuses referring to the order as /* \$data[0...n]=array('Date'=>'Status date', 'Name'=>'Name', 'Comment'=>'Comment'); e.g. echo \$data[0]['Date']; echo \$data[0]['Name']; */ }</pre> </p>
SOAP	<p>Implementing example:</p> <pre>ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/_soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$s=array(); \$s['token']='YourToken'; \$s['imcode']='XXXXXX'; try {\$data=\$clientS->ListStatuses(\$s);} catch (SoapFault \$exception){\$errS=\$exception;}</pre>

	<pre>if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { print_r(\$data); }</pre>
--	---

3.8. ListStatusesFull. Full list of order statuses.

It allows you to get extended information about shipment status, including items issued and returned in case of partial issuance, actual weight of the purchase, money received and charge type.

XXXXXX means order tracking code or the IS order number.

Refresh rate recommended: once an hour.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
ImId	Yes	string	Parcel tracking number

Response data:

Parameter	Type	Description
statuses		Status array
Date	string	Designation date as YYYY-MM-DD HH:MM:SS
Name	string	Status name
Comment	string	Comment
PD	boolean	Sign of partial order delivery (true / false) Goods in the 'product' unit are displayed during partial issuance. If the order is fully issued, the unit is empty.
sum	string	The amount of actually received money from the recipient
PaymentMethod	string	Charge type. Possible values: "Till", "Bank", "Acquiring" "Till" – payment in cash. The default value for the status is "Issued" "Bank" – the order is fully prepaid "Acquiring" – payment by debit card
Weight	float	Actual weight, kg
products		Produst array (if PD=true)
Name	string	Commodity heading
Give	integer	Number of items issued
Return	integer	Number of items returned
Item_ID	string	Item number

Table 9. ListStatusesFull method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ListStatusesFull&ImId=XXXXXX</p> <p>Implementing example: \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ListStatusesFull&ImId=XXXXXX'; // XXXXXX - tracking code \$handle = fopen(\$url, "rb"); \$contents = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$contents,true); if(count(\$data)<=0 or \$data['err']) { // if something went wrong and the response has not been received: echo \$data['err']; } else { // everything is correct, the response has been received and sent to \$data array, // all statuses referring to the order as /*</p>

	<pre> \$data['statuses'][0..n]=array('Date'=>'Status date', 'Name'=>'Name', 'Comment'=>'Comment'); \$data['PD']=>'sign of PI (true / false)'; \$data['sum']=>'the amount of actually received money from the recipient'; \$data['PaymentMethod']=>'Charge type: Till / Bank / Acquiring'; \$data['Weight']=>'Actual weight'; // information about the items in the parcel is available only if PD = true: \$data['products'][0..n]=array('Name'=>'Name of the item', 'Give'=>'number of items issued', 'Return'=>'number of items returned', 'Item_ID'=>'Item number'); */ } </pre>
SOAP	<p>Implementing example:</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$s=array(); \$s['token']='YourToken'; \$s['imcode']='XXXXXX'; try {\$data=\$clientS->ListStatusesFull(\$s);} catch (SoapFault \$exception){\$errS=\$exception;} if(@\$errS){echo \$errS;} // if something went wrong and the response has not been received. else { print_r(\$data); } </pre>

Table 10. A list of possible statuses

Direct flow statuses	Description	Delivery type
The IS registry is uploaded	The order is uploaded to Boxberry IS (The act is formed via ParseSend method) The status is available only for ListStatusesFull method).	PUP, CD
Delivery accepted	The parcel is marked and received at the Boxberry warehouse.	PUP, CD
Handed on sortation	The order is prepared for shipment to a destination city.	PUP, CD
Sent to a marshalling yard	During transit to a marshalling yard.	PUP, CD
Sent to a destination city	The parcel is actually sent to a destination city.	PUP, CD
Handed to be delivered to a pick-up point	Sent to a pick-up point.	PUP
Handed to be delivered by a courier	The parcel has arrived to a destination city, delivery time is arranged with a recipient. The parcel is handed to a courier.	CD
Delivered to a pick-up point	The parcel has arrived at a pick-up point. Ready to be picked.	PUP
Issued	The parcel has been issued by a recipient.	PUP, CD

Return flow statuses	Description	
Returned from courier delivery	The parcel has been returned from courier delivery	CD
Prepared to be returned	The parcel is being prepared to be returned but is actually at the pick-up point.	PUP, CD
Sent to a parcel depository	The parcel has been sent to be returned.	PUP, CD
Returned to a parcel depository	The parcel has been delivered to a Boxberry return warehouse.	PUP, CD
Returned to IS	The parcel has been sent to an internet store.	PUP, CD

3.9. ListServices. Cost of serviced charged.

This service allows you to get the list of services actually rendered with the cost.

The services are available for direct flow orders.

In examples XXXXXX means order tracking code or the IS order number.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
ImId	Yes	string	Parcel tracking number

Response data:

Parameter	Type	Description
Services provided array		
Date	string	Service charging date as YYYY-MM-DD HH:MM:SS"
Name	string	Service name
Sum	string	Cost of service charged, rub
PaymentMethod	integer	Way of payment. Possible values: "Till", "Bank", "Acquiring"

Table 11. ListServices method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ListServices&ImId=XXXXXX</p> <p>Implementing example: <pre>\$url='https://api.boxberry.ru/json.php?token=YourToken&method=ListServices&ImId=XXXXXX'; // XXXXXX - tracking code \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // all statuses referring to the order as /* \$data[0...n]=array('Data'=>'Service charging date', 'Name'=>'Name', 'Sum'=>'Sum', 'PaymentMethod '=>'Way of payment'); e.g. echo \$data[0]['Date']; echo \$data[0]['Sum']; */ }</pre> </p>
SOAP	<p>Implementing example:</p> <pre>ini_set("soap.wsdl_cache_enabled", "0"); \$client = new SoapClient("https://api.boxberry.ru/_soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$soap = array(); \$soap['token']='YourToken'; \$soap['imcode']='XXXXXX';</pre>

	<pre>try {\$data=\$clientS->ListServices(\$s);} catch (SoapFault \$exception){\$errS=\$exception;} if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { print_r(\$data); }</pre>
--	--

3.10. CourierListCities. List of cities with courier delivery service.

It allows you to get the list of cities with courier delivery service.

It also returns the delivery period from Moscow. There is no delivery period for cities with no Boxberry pick-up points.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token

Response data:

Parameter	Type	Description
City	string	Settlement
Region	string	District
Area	string	Area
DeliveryPeriod	string	Delivery period, calendar days

Table 12. CourierListCities method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=CourierListCities</p> <p>Implementing example:</p> <pre> \$url='https://api.boxberry.ru/json.php?token=YourToken&method=CourierListCities'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // a list of cities having a courier delivery service as /* \$data[0..n]=array('City'=>'Settlement', 'Region'=>'Region', 'Area'=>'Area', 'DeliveryPeriod'=>'Delivery period'); */ } </pre>
SOAP	<p>Implementing example:</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$s=array(); \$s['token']='YourToken'; \$s['imcode']='XXXXXX'; try {\$data=\$clientS->CourierListCities (\$s);} catch (SoapFault \$exception){\$err=\$exception;} if (@\$err){ echo \$err; } // if something went wrong and the response has not been received. else { </pre>

	<pre>print_r(\$data); }</pre>
--	-----------------------------------

3.11. DeliveryCosts. Delivery cost calculator.

It allows you to get a quote of delivery to a pick-up point, including constant service cost provided by the contract, it is possible to pay taking courier delivery into consideration.

The availability of **courier delivery** is defined by the recipient's zipcode.

The zipcode should be preliminary checked via API [ListZips](#) or [ZipCheck](#) methods.

Attention! The method can be only used for calculation of delivery cost and is not acceptable for input data checking.

Input parameters:

Parameter	Mandatory	Type	Description	Example to PUP	Example of CD
token	Yes	string	API-service access token		
weight	Yes	integer	Weight of the parcel, gr	1000	1000
targetstart	Yes	string	Parcel depositary code	010	010
target	No	string	Pick-up point code	1002	
ordersum	No	float	Declared value of the parcel (insured value)	2000	2000
deliverysum	No	float	Declared delivery cost	100	100
paysum	No	float	Sum to be paid by a recipient	2000	2000
height	No	string	The height of the box, cm	10	10
width	No	string	The width of the box, cm	10	10
depth	No	string	The depth of the box, cm	10	10
zip	No	string	The zipcode for courier delivery (if sent, the target is ignored)		620028
suchr	No	string	Calculation with the settings in the PA IS taken into consideration: «Integration» - «Widgets customization». Possible values: 1 - get the calculation with the customized settings taken into consideration 0 - get the calculation without any settings. To apply the settings, you must enable the mode «Payroll customizing on».	1	1
cms	No	string	The parameter is for the CMS implementers, carrying out the integration with Boxberry. CMS name must be transferred in the parameter. To be applied only to JSON.	Bitrix	Bitrix

Response data:

Parameter	Type	Description
price	float	Total delivery cost (price_base + price_service).
price_base	float	Basic tariff price. Basic tariff price is formed depending on the weight and dimensions of the parcel, as well as the tariff zone.
price_service	float	The price of mandatory additional services chosen when entering into the contract, such as: <ul style="list-style-type: none"> • notifications of the recipient, • extra charge for density if the dimensions are given, • trying on, • courier delivery if the zipcode for delivery is given etc.
delivery_period	string/ integer	Delivery period taking the type of issuance into consideration (at a pick-up point or via courier delivery service). If <i>sucrh</i> parameter is not specified or is 0, the value has the <i>string</i> type. If <i>sucrh</i> parameter is 1, the value has the <i>integer</i> type.

Table 13. DeliveryCost method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=DeliveryCosts&weight=500&target=010&ordersum=0&deliverysum=0&paysum=0&targetstart=010&height=200&width=200&depth=200&zip=624000</p> <p>Input parameters:</p> <p>weight - weight of the parcel, gr, target - pick-up point code,</p> <p>Attention! The following fields are considered to be 0 if they are empty:</p> <p>ordersum - item price without delivery cost, deliverysum - declared IS delivery cost, targetstart - parcel depository code, height - height of the box (cm), width - width of the box (cm), depth - depth of the box (cm), zip - recipient's zipcode (for courier delivery service), paysum - sum of money to be paid, sucrh - the parameter is added to DeliveryCosts and allows you to receive calculations taking into account the extra charges, set in the PA - Integration tools settings (sucrh=1), cms - the parameter is for the CMS implementers, carrying out the integration with Boxberry. CMS name must be transferred in the parameter.</p> <p>The output contains a variable price having the total cost, rub, and the components of this price (the base cost and the service cost).</p> <pre>\$url='https://api.boxberry.ru/json.php?token=YourToken&method=DeliveryCosts&weight=500&target=010&ordersum=0&deliverysum=0&targetstart=010&height=120&width=80&depth=50&zip=624000'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else {</pre>

	<pre> // everything is correct, the response has been received and sent to \$data array, // price of sending of the parcel given /* \$data['price']=30.60; \$data['price_base']=25.60; \$data['price_service']=5.00; \$data['delivery_period']=1; // delivery period, calendar days. */ } </pre>
SOAP	<p>Implementing example:</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$s = array(); \$s['token']='YourToken'; \$s['weight']=500; \$s['target']='010'; try {\$data=\$clientS->DeliveryCosts(\$s);} catch (SoapFault \$exception){ \$errS=\$exception;} if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received else { print_r(\$data); } </pre>

3.12. PointsForParcels. The list of parcel depositaries.

It allows you to get the list of Boxberry parcel depositaries which can receive parcels from the internet store for further delivery.

Table 14. PointsForParcels method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=PointsForParcels</p> <p>Implementing example:</p> <pre> \$url="https://api.boxberry.ru/json.php?token=YourToken&method=PointsForParcels"; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data[0]['err']) { // if something went wrong and the response has not been received: echo \$data[0]['err']; } else { // everything is correct, the response has been received and sent to \$data array, // the list of parcel depositaries as /* \$data[0..n]=array('Code'=>'Destination code', 'Name'=>'Name', 'City'=>'City'); */ } </pre>
SOAP	<p>Implementing example:</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); </pre>

	<pre>\$s=array(); \$s['token']='YourToken'; try {\$data=\$clientS->PointsForParcels(\$s);} catch (SoapFault \$exception){\$errS=\$exception;} if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { print_r(\$data); }</pre>
--	---

3.13. PointsDescription. Extended information about PUP.

It allows you to get the exact information about a specific pick-up point, including photos.

In examples XXXX means the code of a PUP ("code", which is a Boxberry PUP, can be received via ListPoints method).

To display the photos you should insert an argument "photo=1". The photo array will be returned into base64.

The request can be sent without an argument "photo", in this case, the information about a PUP will be displayed without photos.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
code	No	string	Pick-up point code on a Boxberry database
photo	No	boolean	Display photos (true/false or 0/1)

Return data:

N	Parameter	Type	Description
1	Name	string	Pick-up point name
2	Organization	string	Legal name
3	ZipCode	integer	Zipcode
4	Country	string	Name of the country
5	Area	string	Area
6	CityCode	string	Boxberry city code
7	CityName	string	Name of the city
8	Settlement	string	Name of the settlement
9	Metro	string	Metro station
10	Street	string	The name of the street
11	House	string	House
12	Structure	string	Building
13	Housing	string	Housing
14	Apartment	string	Office/Apartment
15	Address	string	Full address
16	AddressReduce	string	Abbreviated address (street, house, flat)
17	GPS	string	GPS coordinates
18	TripDescription	string	Directions
19	Phone	string	Phone number(s)
20	ForeignOnlineStoresOnly	boolean	Issuance of parcels from foreign internet stores only (true/false)
21	PrepaidOrdersOnly	boolean	Issuing of prepaid orders: true - only issuing without payment acceptance, false - issuing of any orders
22	Acquiring	boolean	Debit card payment available (true/false)
23	DigitalSignature	boolean	A tablet for digital signature available: true - the recipient's signature will be stored in electronic format in the Boxberry system, false - no electronic signatures
24	TypeOfOffice	byte	Pick-up point type: 1- own department, 2- partner department

25	CourierDelivery	boolean	The pick-up point provides the courier delivery service (true/false)
26	Reception	boolean	Receipt parcels from internet stores (true/false)
27	ReceptionLaP	boolean	Receipt of letters and parcels from a physical person (true/false)
28	DeliveryLaP	boolean	Issuance of letters and parcels from a physical person (true/false)
29	LoadLimit	decimal	Limit of weight, kg
30	VolumeLimit	decimal	Volume limit, m3
31	EnablePartialDelivery	boolean	Possibility of partial order delivery (true / false)
32	EnableFitting	boolean	Possibility of trying-on (true/false)
33	fittingType	boolean	Type of trying-on: 1 - partial, 2 - full
34	WorkShedule	string	A string for work schedule
35	WorkMoBegin	string	Monday, start of working time
36	WorkMoEnd	string	Monday, close of a business day
37	WorkTuBegin	string	Tuesday, start of working time
38	WorkTuEnd	string	Tuesday, close of a business day
39	WorkWeBegin	string	Wednesday, start of working time
40	WorkWeEnd	string	Wednesday, close of a business day
41	WorkThBegin	string	Thursday, start of working time
42	WorkThEnd	string	Thursday, close of a business day
43	WorkFrBegin	string	Friday, start of working time
44	WorkFrEnd	string	Friday, close of a business day
45	WorkSaBegin	string	Saturday, start of working time
46	WorkSaEnd	string	Saturday, close of a business day
47	WorkSuBegin	string	Sunday, start of working time
48	WorkSuEnd	string	Sunday, close of a business day
49	LunchMoBegin	string	Monday, lunch, beginning
50	LunchMoEnd	string	Monday, lunch, end
51	LunchTuBegin	string	Tuesday, lunch, beginning
52	LunchTuEnd	string	Tuesday, lunch, end
53	LunchWeBegin	string	Wednesday, lunch, beginning
54	LunchWeEnd	string	Wednesday, lunch, end
55	LunchThBegin	string	Thursday, lunch, beginning
56	LunchThEnd	string	Thursday, lunch, end
57	LunchFrBegin	string	Friday, lunch, beginning
58	LunchFrEnd	string	Friday, lunch, end
59	LunchSaBegin	string	Saturday, lunch, beginning
60	LunchSaEnd	string	Saturday, lunch, end
61	LunchSuBegin	string	Sunday, lunch, beginning
62	LunchSuEnd	string	Sunday, lunch, end
63	photos [0..∞]		PUP photos array
64	photos		A photo in base64
65	TerminalCode	string	A code in Boxberry base
66	TerminalName	string	Item name
67	TerminalOrganization	string	Legal name
68	TerminalCityCode	string	Boxberry city code

69	TerminalCityName	string	Name of the city
70	TerminalAddress	string	Full address
71	TerminalPhone	string	Phone number
72	CountryCode	string	City code
73	TransType	integer	Possible way of transportation: 0 - any type transportation, 1 – vehicles, 2 - airshipment
74	InterRefunds	integer	Foreign reception returns (0/1)
75	ExpressReception	integer	Express-ordering from internet stores (0/1)
76	Terminal	integer	The department is a terminal (0/1)
77	IssuanceBoxberry	integer	The department provides the issuance of parcels (0/1)
78	schedule	string	An actual working schedule of the department contains relevant information in case of temporary change of department working hours. It is sent by the json-coded string

Table 15. PointsDescription method

Protocol	Order format / implementing example for PHP-version above 5.0
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=PointsDescription&code=XXXXX&photo=1</p> <p>Integration example:</p> <pre> Surl = https://api.boxberry.ru/json.php?token=YourToken&method=PointsDescription&code=XXXXX&photo=1'; // token – your API token, // code – PUP code, // photo – get a photo, if required., // If photo = 1, full-sized images of PUP array will be returned into base64. \$handle = fopen(\$surl, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data = json_decode(\$content, true); if (count(\$data) <= 0 or \$data['err']) { // if something went wrong and the response has not been received: echo \$data['err']; } else { /* print_r(\$data); */ // everything is correct, the response has been received and sent to \$data array, // detailed information about a pick-up point: /* Array([Name] => A name of a pick-up point, [IssuanceBoxberry] => 1 – the department provides the issuance of parcels, 0 – packaging release is unavailable, [Organization] => Organization, [ZipCode] => Zipcode, [Country] => Country, [Area] => Area/Province, [CityCode] => City code, [CityName] => Name of the city, [Settlement] => Settlement, [Metro] => Metro station, [Street] => Street, [House] => House, [Structure] => Structure, [Housing] => Housing, [Apartment] => Apartment/office, [Address] => Full address, [AddressReduce] => Abbreviated address, [GPS] => GPS coordinates, */ </pre>


```
        array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)
    );

    $s = array();
    $s['token'] = 'YourToken';
    $s['code'] = 'XXXX';
    $s['photo'] = '1';

    // token – your API token,
    // code – PUP code,
    // photo – Get a photo,if required.
    // If photo = 1, fullsized images of PUP array will be returned into base64.

    $data = array();

    try {
        $data = json_decode(json_encode($clientS->PointsDescription($s)), true);
    } catch (SoapFault $exception) {
        $errS = $exception;
    }

    if (@$errS) {
        echo $errS;
    } // if something went wrong and the response has not been received.
    else {
        /*
            print_r($data);
        */
        /*
            // everything is correct, the response has been received and sent to $data array,
            // detailed information about a pick-up point:
        */
        Array(
            [result] => Array(
                [Name] => A name of a pick-up point,
                [Organization] => Organization,
                [ZipCode] => Zipcode,
                [Country] => Country,
                [Area] => Area/Province,
                [CityCode] => City code,
                [CityName] => Name of the city,
                [Settlement] => Settlement,
                [Metro] => Metro station,
                [Street] => Street,
                [House] => House,
                [Structure] => Structure,
                [Housing] => Housing,
                [Apartment] => Apartment/office,
                [Address] => Full address,
                [AddressReduce] => Abbreviated address,
                [GPS] => GPS coordinates,
                [TripDescription] => General information about location,
                [Phone] => Phone number,
                [ForeignOnlineStoresOnly] => For foreign internet stores only,
                [PrepaidOrdersOnly] => Receipt of prepaid orders only,
                [Acquiring] => Acquiring (debit cards are accepted, the terminal is available),
                [DigitalSignature] => Digital signature,
                [TypeOfOffice] => Pick-up point type: 1-PUP, 2-PUPN,
                [CourierDelivery] => Courier delivery available,
                [ReceptionLaP] => Receives letters,
                [DeliveryLaP] => Issues letters,
                [LoadLimit] => Limit of weight, kg,
                [VolumeLimit] => Volume limit,
                [EnablePartialDelivery] => Partial issuance is available,
                [EnableFitting] => Trying-on available,
                [WorkShedule] => Work schedule,
                [WorkMoBegin] => Monday, start of working time,
                [WorkMoEnd] => Monday, close of a business day,
                [WorkTuBegin] => Tuesday, start of working time,
                [WorkTuEnd] => Tuesday, close of a business day,
                [WorkWeBegin] => Wednesday, start of working time,
                [WorkWeEnd] => Wednesday, close of a business day,
                [WorkThBegin] => Thursday, start of working time,
                [WorkThEnd] => Thursday, close of a business day,
```

	<pre>[WorkFrBegin] => Friday, start of working time, [WorkFrEnd] => Friday, close of a business day, [WorkSaBegin] => Saturday, start of working time, [WorkSaEnd] => Saturday, close of a business day, [WorkSuBegin] => Sunday, start of working time, [WorkSuEnd] => Sunday, close of a business day, [LunchMoBegin] => Monday, lunch, beginning, [LunchMoEnd] => Monday, lunch, end, [LunchTuBegin] => Tuesday, lunch, beginning, [LunchTuEnd] => Tuesday, lunch, end, [LunchWeBegin] => Wednesday, lunch, beginning, [LunchWeEnd] => Wednesday, lunch, end, [LunchThBegin] => Thursday, lunch, beginning, [LunchThEnd] => Thursday, lunch, end, [LunchFrBegin] => Friday, lunch, beginning, [LunchFrEnd] => Friday, lunch, end, [LunchSaBegin] => Saturday, lunch, beginning, [LunchSaEnd] => Saturday, lunch, end, [LunchSuBegin] => Sunday, lunch, beginning, [LunchSuEnd] => Sunday, lunch, end, [TemporaryWorkSchedule] => Temporary work schedule of departments, [Photos] => Photos array in base64, // Data of the terminal which has the department under the control. // For example, for Kamensk-Uralsky it is "Ekaterinburg (terminal)", // Data is similar to the contact details of the PUP: Code, Name... Prefix Terminal. [TerminalCode] => [TerminalName] => [TerminalOrganization] => [TerminalCityCode] => [TerminalCityName] => [TerminalAddress] => [TerminalPhone] => [Reception] => [TransType] => 0 - any type of transportation, 1 - vehicles, 2 - airshipment,) */ }</pre>
--	--

4. API-services for transferring data

The API services described in this section allow you to create an order for uploading to the Boxberry database, as well as acts of acceptance of delivery (incl. labels and print forms used when the parcels are sent to a Boxberry warehouse).

1. **ParselCreate** – allows to create/update the order in the Personal account. The information with an order identifier (tracking code) and a tag printing file reference are displayed.
2. **ParselCheck** - allows to get a tag printing file reference according a specified order.
3. **ParselList** - allows to get a list of all tracking codes that were created in the Personal account but not included in any act of acceptance of delivery.
4. **ParselDel** - allows to delete the order from the Personal account if it is not included in any act of acceptance of delivery.
5. **ParselStory** - allows to get a list of orders created in the Personal account. If you do not specify date intervals, the last created order will be returned.
Attention! The service works with any parcels.
6. **ParselSend** - allows to form an act of acceptance of delivery in the Personal account (uploading the orders to the Boxberry Database is triggered at the same time).
Attention! The service works with any parcels.
7. **ParselSendStory** - allows to get a list of acts of acceptance of delivery created via API. If you do not specify date intervals, the last created act will be returned. Attention! The service works with any parcels.
8. **CreateIntake** - allows to create a request order to issue the parcels.
9. **OrdersBalance** - allows to get information about the orders that have actually been sent to Boxberry to be delivered but have not been delivered to a client.

4.1. ParselCreate. Create/upload an order.

This method allows you to create/update an order in Boxberry. It is important to note that

- while creating a new order, the services chosen in the Personal account, substitute automatically (<https://account.boxberry.ru>) – "Services" section.
- 'issue' parameter is responsible for changing the type of issuance of every shipment.

The description of data transferred:

Parameter	Type	Description	Mandatory fields		
			PUP	CD	RP
token	string	API-service access token.	Yes	Yes	Yes
partner_token	string	Token of an integration partner	No	No	No
updateByTrack	string	Tracking code of a previously created order. Take into consideration that if the updateByTrack parameter is filled, it will mean that you want to update a previously created order.	No	No	No
order_id	string	id - identifier (number) of an IS order.	Yes	Yes	Yes
PalletNumber	string	Pallet number, is given if you send the parcels using pallets.	No	No	No
barcode	string	Parcel barcode. It is given when the own parcel labelling is used. See Labelling requirements.	No	No	No
price	string	Declared value - total (estimated) IS value, rub.	No	No	No
payment_sum	string	Sum of money to be paid (by a recipient), rub. 0 is specified for a prepaid order.	No	No	No
delivery_sum	string	Delivery sum of money, declared to the recipient, rub.	No	No	No
vid	string	Delivery type. Possible values: 1 - Delivery to a pick-up point (PUP) 2 - Courier delivery (CD) 3 - Delivery by Russian Post (RP) - this delivery type is not available.	Yes	Yes	Yes
kurdost		<i>Data unit about courier delivery (vid => 2) and Russian Post delivery (vid => 3)</i>	No	Yes	Yes
index	string	Recipient's zipcode	No	No	No
citi	string	The name of recipient's city that has the courier delivery. It is necessary to specify the Area, District, City for small settlements, the address is checked by a third-party service and must be unequivocally defined. <i>Admissible alternatives:</i> 1. Sverdlovskaya, Kushva, Baranchinsky. 2. Sverdlovskaya area, Kushva town, Baranchinsky village. 3. Ekaterinburg. 4. Ekaterinburg city.	No	Yes	Yes
addressp	string	PO recipient's address (street, hose, flat/office).	No	Yes	Yes

The following definitions are sent to the Courier delivery only for directions: — Moscow - Moscow — Moscow - Saint Petersburg — Saint Petersburg - Moscow — Saint Petersburg - Saint Petersburg			No	No	No
delivery_date	string	Courier delivery date (YYYY-MM-DD). The delivery period can take from +1 to +5 days from the current date. Default delivery period - current date + 1 day. It is ignored for other directions.	No	No	No
timesfrom1	string	Delivery period FROM (as HH:MM). Default time 10:00	No	No	No
timesto1	string	Delivery period TO (as HH:MM) Default time 18:00 The interval "FROM - TO" should take not less than 3 hours. If the interval takes less, it can be corrected without notifications.	No	No	No
timesfrom2	string	Alternative time, from	No	No	No
timesto2	string	Alternative time, to	No	No	No
timep	string	The text format of delivery period (cannot be processed)	No	No	No
comentk	string	Comment for a courier. Before delivery start, a courier has to call the recipients and agree on the delivery conditions.	No	No	No
<i>The following definitions are sent to the Russian Post delivery</i>			No	No	No
type	string	Shipment type, possible values: 0 - Parcel, 2 - Courier online, 3 - Parcel online, 5 - Parcel of the 1st class. Default value is 0, if not specified.	No	No	No
fragile	string	Fragile parcel, possible values are 0 and 1. Default value is 0, if not specified.	No	No	No
strong	string	Strong type, possible values are 0 and 1. Default value is 0, if not specified.	No	No	No
optimize	string	Tariff optimization, possible values are 0 and 1. Default value is 1, if not specified. Strong and optimize parameters are conflicting and cannot have value 1 at the same time.	No	No	No
packing_type	integer	Packing type, possible values: 0 - IS packing, sum of sides is <= 106 cm 1 - IS packing, sum of sides is > 106 cm 2 - Boxberry packing, sum of sides is <= 106 cm 3 - Boxberry packing, sum of sides is > 106 cm	No	No	No
packing_strict	boolean	Strict packing, possible values: true - packing changing is forbidden during transportation false - packing changing is allowed	No	No	No
shop		<i>Data unit about a parcel depositary and a pick-up point</i>	Yes	Yes	Yes
name	string	Pick-up point code	Yes	No	No
name1	string	Depositary point code	Yes	Yes	Yes

customer		<i>Data unit about a recipient</i>	Yes	Yes	Yes
fio	string	Recipient's name	Yes	Yes	Yes
phone	string	Recipient's phone number. The phone number must contain 10 figures, otherwise the amount of characters is reduced and only 10 figures from the end are used.	Yes	Yes	Yes
phone2	string	Additional phone number of a recipient.	No	No	No
email	string	Recipient's e-mail. It is mandatory if the order options include "e-mail notifications" and/or "additional e-mail notifications".	No	No	No
<i>The following values are sent to a legal recipient.</i>			No	No	No
name	string	Name of the organization	Yes	Yes	Yes
address	string	Company's legal address	No	No	No
inn	string	TIN	Yes	Yes	Yes
kpp	string	TRRC	No	No	No
r_s	string	Current account	No	No	No
bank	string	Organization bank	No	No	No
kor_s	string	Offset Account	No	No	No
bik	string	BIC	No	No	No
items		<i>Data unit about items included in the order</i>	No*	No*	No**
id	string	Identifier/item number	No	No	No
name	string	Name of the item	Yes	Yes	Yes
UnitName	string	Unit measure, pcs are used as default.	No	No	No
nds	string	VAT (figure from 0 to 20)	No	No	No
price	string	Price per unit	Yes	Yes	Yes
quantity	string	Number of items in the commodity heading	Yes	Yes	Yes
weights		<p><i>Information about PO weight:</i></p> <ul style="list-style-type: none"> — weight - means the weight of the first or single packaging unit, gr Minimum value is 5 gr, maximum value is 31,000 gr. — weight2, weight3, ... weight24 - mean the weight of the second and following packaging units, gr. Take into consideration that these strings are added provided that PO is delivered in two or more packaging units. <p><i>Barcode units information</i> is given if each packaging unit is marked with own unique barcode labelling.</p> <ul style="list-style-type: none"> — Each barcode has to be filled in the corresponding weight string — If the units have at least one barcode, the others should be defined, too. — If the units have a barcode, it is not necessary to fill in the "PO barcode" string. <p><i>SOAP format has 24 elements, JSON has an unlimited number of elements.</i></p>	Yes	Yes	Yes
weight	string	The weight of the first packing unit (gr). It shall be filled in in all cases!	Yes	Yes	Yes
barcode	string	Barcode 1 of the first packing unit.	No	No	No

		See Labelling requirements.			
x	string	Packing unit dimensions (cm) They are mandatory for Russian Post delivery.	No	No	Yes
y	string		No	No	Yes
z	string		No	No	Yes
weight2	string	Weight and barcodes of the following packing units	No	No	-
barcode2	string		No	No	-
...					
weight24	string		No	No	-
barcode24	string		No	No	-
issue	integer	'issue' – type of issuance, possible values: 0 - issuance without unpacking, 1 - issuance with unpacking and checking of completeness, 2 - issuance of parts of the contents. If not specified, the default value, arranged in the profile page of the internet store - "Services" section.	No	No	No
sender_name	string	A name of the store-sender for sms/e-mail notifications and home marking of Boxberry.	No	No	No

Response data

Parameter	Type	Description
track	string	Tracking number of the parcel
label	string	A tag printing file link (is generated if the IS barcode is not given)

Table 16. ParselCreate method

Protocol	Order format / implementing example
	The structure of an incoming array:
	<pre> SSDATA=array(); SSDATA['updateByTrack']='Tracking code of a previously created order'; SSDATA['order_id']='Ordre ID in IS'; SSDATA['PalletNumber']='Pallet number'; SSDATA['barcode']='Order barcode'; SSDATA['price']='Declared value'; SSDATA['payment_sum']='Sum to be paid'; SSDATA['delivery_sum']='Delivery cost'; SSDATA['vid']='Delivery type (1/2)'; SSDATA['shop']=array('name'=>'PUP code', 'name1'=>'Destination code'); SSDATA['customer']=array('fio'=>'recipient's name', 'phone'=>'Phone number', 'phone2'=>'Additional phone number', 'email'=>'E-mail for notifications', 'name'=>'Name of organization', 'address'=>'Address', 'inn'=>'TIN', 'kpp'=>'TRRC', 'r_s'=>'Current account', </pre>

	<pre> 'bank'=>'Name of the bank', 'kor_s'=>'Offset account', 'bik'=>'BIC'); \$\$DATA['kurdost'] = array('index' => 'Zipcode', 'citi' => 'City', 'address' => 'Recipient's address', 'timesfrom1' => 'Delivery period, from', 'timesto1' => 'Delivery period, to', 'timesfrom2' => 'Alternative time, from', 'timesto2' => 'Alternative time, to', 'timep' => 'The text format of delivery period', 'delivery_date' => 'The date of delivery is from +1 to +5 days from the current date (for Moscow and Saint Petersburg only)', 'comentk' => 'Comment'); \$\$DATA['items']=array(array('id'=>'Item number in the DB', 'Name'=>'Name of the item', 'UnitName'=>'Unit measure', 'nds'=>'VAT %', 'price'=>'Price per unit', 'quantity'=>'Quantity')); \$\$DATA['weights']=array('weight'=>'Weight of packing unit 1', 'barcode'=>'Barcode of the first unit', 'weight2'=>'Weight of packing unit 2', 'barcode2'=>'Barcode of the second unit', 'weight3'=>'Weight of packing unit 3', 'barcode3'=>'Barcode of the third unit', 'weight4'=>'Weight of packing unit 4', 'barcode4'=>'Barcode of the fourth unit', 'weight5'=>'Weight of packing unit 5', 'barcode5'=>'Barcode of the fifth unit', </pre>
JSON	<p>Integration example for PHP-version above 5.0:</p> <pre> // It is assumed that you have already created the \$\$DATA array using the example described above. // The array is sent to the Boxberry server via CURL. \$ch = curl_init(); curl_setopt(\$ch, CURLOPT_URL, 'https://api.boxberry.ru/json.php'); curl_setopt(\$ch, CURLOPT_POST, true); curl_setopt(\$ch, CURLOPT_POSTFIELDS, array('token'=>'YourToken', 'method'=>'ParseCreate', 'sdata'=>json_encode(\$\$DATA))); curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); \$data = json_decode(curl_exec(\$ch),1); if(\$data['err'] or count(\$data)<=0) { </pre>

	<pre> // if something went wrong and the response has not been received. echo \$data['err']; } else { // everything is correct, the response has been received and sent to \$data array. // (no label if the barcode is given). /* \$data=array('track'=>'XXXXXXXXX', // Tracking code for the parcel. 'label'=>'http://' // Link to download the PDF file with labels.); */ } </pre>
<p>SOAP</p>	<p>Integration example for PHP-version above 5.0</p> <pre> // It is assumed that you have already created the \$SDATA array using the example described above: ini_set("soap.wsdl_cache_enabled", "0"); // disable WSDL caching. try { \$clientS = new SoapClient('https://api.boxberry.ru/soap/1c_1c.php?wsdl'); } catch (SoapFault \$exception){ \$errS=\$exception; } try { \$get_data=\$clientS->ParselCreate(\$SDATA); } catch (SoapFault \$exception){ \$errS=\$exception; } if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { // INRESPOND: /* stdClass Object ([track] => XXXXXXXXXX // Tracking code for the parcel, [label] => http:// // Link to download the PDF file with labels.) */ } </pre> <p>Integration example for 1C: Enterprise:</p> <p>Procedure ParselCreateClick(Element)</p> <pre> Проху= WSCссылки.Parsel.СоздатьWSПрокси("api.boxberry.de", "BoxBerry", "BoxBerrySoap"); ТипСписок = Прокси.ФабрикаХДТО.Тип("api.boxberry.de", "ParselCreateQuery"); ТипShop = Прокси.ФабрикаХДТО.Тип("api.boxberry.de", "ParselCreateShopData"); ТипCustomer = Прокси.ФабрикаХДТО.Тип("api.boxberry.de", "ParselCreateCustomerData"); ТипWeights = Прокси.ФабрикаХДТО.Тип("api.boxberry.de", "ParselCreateWeightsData"); </pre>

```
// Fill in the parameter:

WSПараметрСписок = Прокси.ФабрикаХДТО.Создать(ТипСписок);
WSПараметрСписок.token = "Your API token";
WSПараметрСписок.order_id = "Order ID in IS";
WSПараметрСписок.PalletNumber = "Pallet number";
WSПараметрСписок.barcode = "barcode";
WSПараметрСписок.price = "Declared value";
WSПараметрСписок.payment_sum = "Sum to be paid";
WSПараметрСписок.delivery_sum = "Delivery cost";
WSПараметрСписок.vid = "Delivery type (1/2)";

СписокShop = Прокси.ФабрикаХДТО.Создать(ТипShop);

СписокShop.name = "PUP code";
СписокShop.name1 = "Destination code";
WSПараметрСписок.shop = СписокShop;

СписокCustomer = Прокси.ФабрикаХДТО.Создать(ТипCustomer);
СписокCustomer.fio = "Recipient's name";
СписокCustomer.phone = "Phone number";
СписокCustomer.phone2 = "Additional phone number";
СписокCustomer.email = "E-mail for notifications";
СписокCustomer.name = "Name of the organization";
СписокCustomer.address = "Address";
СписокCustomer.inn = "TIN";
СписокCustomer.kpp = "TRRC";
СписокCustomer.r_s = "Current account";
СписокCustomer.bank = "Name of the bank";
СписокCustomer.kor_s = "Offset account";
СписокCustomer.bik = "BIC";

WSПараметрСписок.customer = СписокCustomer;

СписокWeights = Прокси.ФабрикаХДТО.Создать(ТипWeights);

СписокWeights.weight = "Weight of packing unit 1";
СписокWeights.weight2 = "0";
СписокWeights.weight3 = "0";
СписокWeights.weight4 = "0";
СписокWeights.weight5 = "0";

WSПараметрСписок.weights = СписокWeights;

// Request:

Attempting

Result = Прокси.ParseCreate(WSПараметрСписок);
Notify("label: " + СокрЛП(Результат.label) + Символы.ПС + "track: " + СокрЛП(Результат.track));

Exception

Notify(ОписаниеОшибки());

КонецПопытки;

КонецПроцедуры
```

4.2. Restrictions on creating a shipment

№	Parameter	Check conditions	Error text
1	order_id Order number in IS	Valid character set: a-z(A-Z), 0-9, a-я(A-Я), ёЁ, dash(-), forward slash(/), dot(.), comma(,), underscore(), №, space	Order number contains forbidden characters.
		Length - 35 characters.	"Order number in IS" should contain 35 characters maximum.
		Order number availability.	«Order number in IS» is necessary to be filled in.
		Unicity in the terms of one PA of IS. For parcels sent via API it is checked on parcels "in the act". If the parcel is not "in the act", the data is rerecorded, and the tracking number does not change.	Value «.....» for «Order number in IS» is already filled.
2	name Pick-up point	The actual PUP code is entered (sign of a section IssuanceBoxberry=1). It is possible to send the code to a closed PUP if it is available in the IS PA base.	The PUP code is invalid.
		PUP code availability.	«Pick-up point» field should be filled.
3	name1 Parcel depositary	The code of an actual PUP is entered (sign of a section Reception=1), the department shall be ready for service. <ul style="list-style-type: none"> — The default PUP for "My profile" settings are used, if not specified, - "Access". — If the user settings do not have the default PUP, the one should be used that is present in "My profile" IS settings - "IS data" - "Parcel depositary". 	The Parcel depositary code is invalid.
4	e-mail	In the case of the default options: the strings "E-Mai notifications" and "Additional E-Mai notifications" shall be filled in.	"Resepient's e-mail" shall be filled in.
5	fio Recipient's name	Name available. At least 2 words with a space (manual ordering). Cyrillic text (manual ordering).	The field «Surname» shall be filled in. The field «Name» shall be filled in. The field «Middle name» shall be filled in. Format attribute «Surname» shall be written in Cyrillic. Format attribute «Name» shall be written in Cyrillic. Format attribute «Middle name» shall be written in Cyrillic.
6	phone The recipient's phone number	The phone number must contain 10 figures, otherwise the amount of characters is reduced and only 10 figures from the end are used.	«Recipient's phone number» shall contain 10 characters.
7	price Declared value of the parcel	Declared value of the parcel should be within the range from 0 to 300,000.	Declared value cannot be higher than 300,000 rub. Declared value cannot have negative value.
		Declared value must have positive value.	Declared value must be a number.
			Declared value cannot have negative value.
8	delivery_sum Delivery cost	Delivery cost must have positive value.	Delivery cost must be a number. Delivery cost cannot have negative value.
9	payment_sum Sum to be paid	Sum to be paid must have positive value.	Sum to be paid must be a number. Sum to be paid cannot have negative value.
		Sum to be paid in the range, defined in IS PA settings (administration section) - "Minimum sum to be paid" and "Maximum sum to be paid"	Sum to be paid cannot be lower than <minimum sum to be paid> and higher than <maximum sum to be paid>.

10	addressp The address for Courier delivery	Fields "city" and "address" shall be filled in. Fields "city" and "address" shall be filled in in Cyrillic. When creating packages, the zipcode given is ignored, "the city + address" is sent to DaData.ru, and the received zipcode is checked for availability of CD. The address with an area and a district shall be written as an area, a districts, a settlement. <i>E.g.</i> <city>Sverdlovskaya area, Kushva town, Baranchinsky village</city> <addressp>Respubliki street, 4</addressp> or without specifying: <city>Sverdlovskaya, Kushva, Baranchinsky</city> <addressp>Respubliki, 4</addressp>	The zipcode cannot be defined, check the address. Courier delivery is not available using the zipcode <index>. «Recipient's city» is necessary to be filled in. «Recipient's address» is necessary to be filled in. Format attribute «Recipient's city» shall be written in Cyrillic. Format attribute «Recipient's address» shall be written in Cyrillic.
11	timesfrom1 timesto1 Courier delivery intervals (destination is defined by the parcel data, by the values "parcel depository" and "pick-up point")	For destinations: Moscow - Moscow Moscow - Saint Petersburg Saint Petersburg - Saint Petersburg Saint Petersburg - Moscow ===== The interval shall be within time limits from 10:00 to 22:00 and shall last at least 3 hours: — If the interval failing to meet the requirements is given, the interval is selected according the algorithm. — If the interval is not given, the default value is set - 10:00-18:00 For other destinations: the interval is ignored.	No errors.
	delivery_date The date of delivery	For destinations: Moscow - Moscow Moscow - Saint Petersburg Saint Petersburg - Saint Petersburg Saint Petersburg - Moscow ===== The date of delivery should be in the interval from +1 to +5 days from the current date. If the date is not given or does not comply with terms, the default delivery date will be the current date + 1 day. For other destinations: the date of delivery is ignored.	No errors.
12	items Goods contents	In case of "partial issuance" all attachments tests of goods must succeed, or the user gets an error notification. In case of issuance "with unpacking" or "without unpacking" , if all tests are conducted, the goods contents must be recorded, if one of the conditions is not fulfilled, the goods are not recorded and the user does not get any errors. Goods contents presence. Cost is a numerical value higher than 0. Quantity is a numerical value higher than 0. Goods contents name presence. VAT is a figure from 0 to 20. If the string is empty, it means 0. At least one good must have the positive price (all goods cannot cost 0 rub.) Goods value check: Sum to be paid = Goods value + delivery cost, or Sum to be paid = 0 (prepaid order).	It is necessary to fill in the attachments for your set of services. Goods are out of stock. The cost of the attachment No.1...n cannot have a negative value The cost of the attachment No.1... is inaccurate The quantity shall be more than 0 for the attachment No.1...n The quantity shall be defined as a whole number No.1...n The name of the attachment No.1... is not defined The VAT of the attachment No.1...n cannot be lower than 0 The VAT of the attachment No.1...n cannot be higher than 20 The VAT of the attachment No.1...n is inaccurate The total sum of all goods shall be higher than 0. Goods value (<sum>) + delivery cost (<sum>) shall equal the sum to be paid (<sum>).
13	weight The weight of the packing unit	The weight of the 1 packing unit must be filled. It is allowable to send only one packing unit for RP parcels. Weight is a numerical value.	«Weight of the box (kg)» must be filled in for No.1...n No packing units available. The weight must be written as a number next to a packing unit

			No.1...n
		The weight of a packing unit cannot exceed the weight limit of the department to get the order.	«Weight of the box (kg)» cannot exceed 15 (31) kg for a chosen PUP next to No.1...n.
		Weight fo CD cannot exceed 25 kg.	The weight of the box cannot exceed 25 kg next to No.1...n.
14	barcode Parcel barcode	IS barcode cannot have 13 characters with a leading 0.	Barcode cannot have 13 characters with a leading 0.
		If the barcode is given, it is not generated and the labelling is not available.	-
15	barcode Packing units barcodes	If at least one packing units has a barcode, the others are mandatory to fill in.	A barcode for <1...n> unit must be filled in.
		If there are packing units barcodes, the parcel barcode is ignored.	-
16 Inspections during Russian Post delivery.			
17	vid issuance type given in the IS PA settings is not analyzed	The availability of Russian Post delivery is defined by the city-sender. The feature "russianpostDelivery" in the table of cities = 1.	The Parcel depositary code is invalid. Russian Post delivery is not available for a PUP chosen
18	fio Recipient's name	Name shall be written in Cyrillic only.	Format attribute «Surname» shall be written in Cyrillic. Format attribute «Name» shall be written in Cyrillic. Format attribute «Middle name» shall be written in Cyrillic.
19	weight Meta-weight	The first and the only one packing unit weight is checked and used! The weight of the further units is ignored.	Only one box must be defined for Russian Post.
	x* y* z Unit dimensions	Unit dimensions are mandatory.	«Width» No.1 must be filled in. «Height» No.1 must be filled in. «Depth» No.1 must be filled in.
20	addressp Recipient's address	Address shall be written in Cyrillic. The address is checked whether it is possible to deliver by Russian Post (the same process as during CD: the zipcode given is ignored, we get the data from <city> and <addressp> and get the zipcode via DaData.ru). The address with an area and a district shall be written as an area, a districts, a settlement. <i>E.g.</i> <city>Sverdlovskaya area, Kushva town, Baranchinsky village</city> <addressp>Respubliki street, 4</addressp> or without specifying: <city>Sverdlovskaya, Kushva, Baranchinsky</city> <addressp>Respubliki, 4</addressp>	The zipcode cannot be defined, check the address Format attribute «Recipient's city» shall be written in Cyrillic. Format attribute «Recipient's address» shall be written in Cyrillic.
21	payment_sum Sum to be paid	Cash on delivery (if the sum to be paid > 0): — Handover of the goods is mandatory. — Sums shall be checked: "sum to be paid = goods value + delivery cost", declared value = sum to be paid. While manual ordering: — Delivery cost shall be counted via Axiomus, cannot be edited. — Declared value (cannot be edited) = sum to be paid.	It is necessary to fill in the attachments for the delivery type chosen with cash on delivery. Goods value (<sum>) + delivery cost (<sum>) shall equal the sum to be paid (<sum>). Declared value shall equal the sum to be paid.
22	payment_sum Sum to be paid	Prepaid order (if the sum to be paid = 0): — Handover of the goods is optional (if handed, then kept). — Declared value is not checked. While manual ordering: — The default declared value is 0, can be edited.	

23	price Declared value	The default declared value is 0, can be edited.	
24	type Parcel type	Checked for correctness (possible values: 0 - Parcel, 2 - Courier online, 3 - Parcel online, 5 - Parcel of the 1st class). Default value is 0, if not specified.	«Parcel type» is invalid.
25	strong Strong type	Checked for correctness (possible values are 0 and 1). Default value is 0, if not specified.	«Strong type» value must be «1» or «0».
26	fragile Fragile goods	Checked for correctness (possible values are 0 and 1). Default value is 0, if not specified.	«"Handle with care" label for fragile goods» value must be «1» or «0».
27	optimize Tariff optimization	Checked for correctness (possible values are 0 and 1). Default value is 1, if not specified.	«Tariff optimization» value must be «1» or «0».

4.3. Labelling requirements for the parcel.

Each parcel shall be labelled (with a tag/label).

Label (tag) shall have the following data:

1. Order number shall conform with ID number in the order parameters in the 'order_id' field
2. Name of the sender (IS) - the name defined in the PA, not the legal name.
3. Recipient's name
4. City of delivery
5. Barcode.
6. Barcode decoding.
7. Number of the packing place and total amount of places in the order (if the order has only one place, "place 1 out of 1" shall be defined. If the order has two places, "place 1 out of 2" and "place 2 out of 2" shall be defined.

CO	boxberry	Tel. 8 (800) 222-80-01		
Sender: Ltd "SMART LOGISTIC", Moscow				
Recipient: Moscow. Self-employed entrepreneur Sysoev Mikhail Pavlovich				
Moscow	Moscow			



99
Testing stand
Petrov Ivan

119002, Moscow, Karmanitsky lane, 9

000030700133100001 1/1 IS
(boxberry)

Labelling requirements:

1. Barcode EAN 128 is recommended.
2. Printing definition: 360 dpi (label shall be clearly printed, not fuzzy, shall not have any printing skips).
3. To read the code symbol properly, it should contain the digital values and **SHOULD NOT start from**
 - 0... with the total amount of characters of 13
 - Latin B... with the total amount of characters of 10
4. The barcode on the label must conform with the barcode number indicated in the 'barcode' field in the order parameters, only in this case the parcel in the warehouse will be identified and correctly processed.

Readme! If the 'barcode' parameter (barcode of order or place) is given, then printing the standard Boxberry label will not be available.

4.4. ParseSend. Act of acceptance of delivery formation.

It forms the Act of acceptance of delivery in Boxberry (uploading the orders to the Boxberry Database is triggered at the same time). The act is formed for both one parcel and a group of parcels.

The <string for tracking codes> ImIds parameter is mandatory. It is important to note that the ImIds parameter does not send your order codes, you send your tracking codes returned via [ParseCreate](#) method.

If there is a request to form the Act for a group of orders uploaded previously (and included to other acts), you will get a response with tracking codes for these orders.

In examples XXXXXX,XXXXXX,XXXXXX mean the list of tracking codes.

Attention! The service works with any parcels.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
ImIds	Yes	string	Tracking code

Response data:

Parameter	Type	Description
id	string	Act number
label	string	Link to an act print form
sticker	string	Link to a label print form

Table 17. ParseSend method

Protocol	Order format / implementing example
JSON	<p>Request:</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParseSend&ImIds=XXXXXX,XXXXXX,XXXXXX</p> <p>Integration example for PHP-version above 5.0:</p> <pre><? \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ParseSend&ImIds=XXXXXX,XXXXXX'; ; // XXXXXX - tracking code: \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data['err']){ // if something went wrong and the response has not been received: echo \$data['err']; }else{ // everything is correct, the response has been received and sent to \$data array: /* \$data=array('id' => Act number, 'upload_id' => Act number, 'label' => http://, 'sticker' => http://); */ } ?></pre>
SOAP	<p>Integration example for PHP-version above 5.0</p> <pre><? ini_set("soap.wsdl_cache_enabled", "0");// disable WSDL caching.</pre>

```

try {
    $clientS = new SoapClient('https://api.boxberry.ru/__soap/1c_1c.php?wsdl');
} catch (SoapFault $exception){
    $errS=$exception;
}

SSDATA=array();
SSDATA['token']='YourToken';
SSDATA['ImlDs']='XXXXXXX,XXXXXXX,XXXXXXX';

try {
    $get_data=$clientS->ParselSend(SSDATA);
} catch (SoapFault $exception){
    $errS=$exception;
}
if(@$errS){
    echo $errS;
} // if something went wrong and the response has not been received.
else
{
    // INRESPOND:
    /*

    stdClass Object
    (
        [result] => stdClass Object
        (
            [id] => Act number,
            [label] => http://,
            [sticker] => http://
        )
    )

    */
}
?>

```

Integration example for 1C: Enterprise

Procedure ParselCreateClick(Element)

```

Proxy = WSCссылки.Parsel.СоздатьWSПрокси("https://api.boxberry.ru", "BoxBerry", "BoxBerrySoap");

// Request:
Attempting
Result = Прокси.ParselSend("Your API token", "XXXXXXX");
Сообщить(СокрЛП(Результат));
Exception
Notify(ОписаниеОшибки());
КонецПопытки;

КонецПроцедуры

```

4.5. ParselSend. Error reference.

№	Error text	Error description
1	"id": "U-XXXXXX", "upload_id": XXXXXX, "label": "https://" "sticker": "https://"	Query evaluation successful. If you try to register the act again within 72 hours, a link to the previously created act is returned.
2	Not all of these parcels can be placed in the act: (tracking code (list) of the order) "	You are trying to send the tracking number of an existing order from the other act or an order not available in PA. If you are trying to register the act again after 72 hours.
3	"Only the parcels with the same pick-up point can be formed into an act."	You are trying to add parcels with different PUPs into one act (ParcelCreate method, "name1" parameter).
4	"No parcel data"	You are trying to send a tracking number which is invalid or unavailable in PA.

4.6. ParselDel. Delete the order.

It allows to delete the order created in Boxberry PA, but not included in any Act of acceptance of delivery.

In examples XXXXXX means the tracking code.

Attention! The service works with any parcels.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
ImId	Yes	string	Tracking code

Response data:

Parameter	Type	Description
ImIds	string	Output

Table 18. ParselDel method

Protocol	Order format / implementing example
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ParselDel&ImId=XXXXXX</p> <p>Integration example for PHP-version above 5.0:</p> <pre><? \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ParselDel&ImId=XXXXXX'; // XXXXXX - tracking code: \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data['err']) { // if something went wrong and the response has not been received: echo \$data['err']; } else { // everything is correct, the response has been received and sent to \$data array: /* \$data=array('text'=>'ok'); */ } ?></pre>
SOAP	<p>Integration example for PHP-version above 5.0</p> <pre><? ini_set("soap.wsdl_cache_enabled", "0"); // disable WSDL caching. try {\$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_lc.php?wsdl");} catch (SoapFault \$exception){\$errS=\$exception;} \$\$DATA=array(); \$\$DATA['token']='YourToken'; \$\$DATA['ImId']='XXXXXX'; try { \$get_data=\$clientS->ParselDel(\$\$DATA); } catch (SoapFault \$exception){ \$errS=\$exception; } if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received.</pre>

	<pre>else { // INRESPOND: /* stdClass Object ([text] => ok) */ }</pre> <p>?></p>
	<p>Integration example for 1C: Enterprise</p> <pre>Procedure ParselDelClick(Element) Proxy = WSCсылки.Parsel.СоздатьWSПрокси("https://api.boxberry.ru", "BoxBerry", "BoxBerrySoap"); // Request: Attempting Result = Прокси.ParselDel("Your API token", "XXXXXX"); Сообщить(СокрЛП(Результат)); Exception Notify(ОписаниеОшибки()); КонецПопытки; КонецПроцедуры</pre>

4.7. CreateIntake. A request order to issue the parcels by a courier.

It allows to create a request order to issue the parcels.

It is important to note that creating a request order to issue the parcels in PA is temporarily available for IS based in Moscow only. The availability of creating a request order to issue the parcel is defined by the zipcode, correct zipcode must be given.

If you need to issue a parcel in other cities, you can consult your account manager.

Input parameters:

Parameter	Type	Description
token	string	Token for the method access
zip	string	Zipcode
city	string	City
street	string	Street
house	string	House
building	string	Building
housing	string	Housing
flat	string	Flat/office
contact_person	string	Contact person
contact_phone	string	Phone number
taking_date	string	Taking date
taking_time_from	string	Taking time "from"
taking_time_to	string	Taking time "to"
seats_count	integer	Number of units
volume	float	Volume (m ³)
weight	float	Weight (kg)
comment	string	Comment

Response data:

Parameter	Type	Description
message	string	Output

Table 19. CreateIntake method

Protocol	Order format / implementing example
	<p style="text-align: center;">The structure of an incoming array:</p> <pre> \$query = array(); \$query['token'] = "YourToken"; // Your API token; \$query['method'] = "CreateIntake"; // Method name; \$query['zip'] = "105005"; // Zipcode; \$query['city'] = "Moscow"; // City; \$query['street'] = "Baumanckaya St."; // Street; \$query['house'] = "56"; // House; \$query['building'] = "B"; // Building.; \$query['housing'] = "5"; // Housing; \$query['flat'] = "4"; // Flat/office; \$query['contact_person'] = "Name"; // Contact person; \$query['contact_phone'] = "+79089999999"; // Phone number; \$query['taking_date'] = "5.05.2016"; // Taking date; \$query['taking_time_from'] = "10:00"; // Taking time from; \$query['taking_time_to'] = "18:00"; // Taking time to; \$query['seats_count'] = "2"; // Number of units; \$query['volume'] = "10"; // Volume (m3); </pre>

	<pre>\$query['weight'] = "4"; // Weight (kg); \$query['comment'] = "Comment"; // Comment.</pre>
	<p>Response data:</p> <pre>Array([Result] => Array([message] => 1632 // Created request number.))</pre>
JSON	<p>Integration example for PHP-version above 5.0</p> <pre>\$url = 'https://api.boxberry.ru/json.php'; \$data = file_get_contents(\$url."?". http_build_query(\$query)); \$data = json_decode(\$data,1); if(isset(\$data['err'])){ // if something went wrong and the response has not been received: print \$data['err']; } else { print_r(\$data); /* Array ([message] => 1635 // Created request number.) */ } ?></pre>
SOAP	<p>Integration example for PHP-version above 5.0</p> <pre><? \$xml = simplexml_load_string(\$response); \$url = "http://dev.api/__soap/1c_lc.php?wsdl"; try{ ini_set("soap.wsdl_cache_enabled", "0"); // disable WSDL caching. \$client = new SoapClient(\$url); \$data = json_decode(json_encode(\$client->CreateIntake(\$query)),1); // everything is correct, the response has been received and sent to \$data array: print_r(\$data); } catch (SoapFault \$e){ // if something went wrong and the response has not been received: print \$e->getMessage(); } ?></pre>

4.8. ParselCheck. Link to a label print form.

It allows to get a tag printing file reference according a specified order.

In examples XXXXXX means the tracking code.

Attention! The service works only for orders created via API.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
ImId	Yes	string	Parcel tracking number

Response data:

Parameter	Type	Description
label	string	A tag printing file link (is generated if the IS barcode is not given)

Table 20. ParselCheck method

Protocol	Order format / implementing example
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ParaselCheck&ImId=XXXXXX</p> <p>Integration example for PHP-version above 5.0:</p> <pre><? \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ParaselCheck&ImId=XXXXXX'; // XXXXXX - Tracking code. \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data['err']){ // if something went wrong and the response has not been received. echo \$data['err']; }else{ // everything is correct, the response has been received and sent to \$data array: /* \$data=array('label'=>'Link to download the PDF file with labels'); */ } ?></pre>
SOAP	<p>Integration example for PHP-version above 5.0</p> <pre><? ini_set("soap.wsdl_cache_enabled", "0"); // disable WSDL caching. try { \$clientS = new SoapClient('https://api.boxberry.ru/__soap/1c_1c.php?wsdl'); } catch (SoapFault \$exception){ \$error=\$exception; } \$SDATA=array(); \$SDATA['token']='YourToken'; \$SDATA['ImId']='XXXXXX'; try { \$get_data=\$clientS->ParaselCheck(\$SDATA); } catch (SoapFault \$exception){ \$error=\$exception; } if(@\$error){</pre>

```
        echo $errS; // if something went wrong and the response has not been received.
    }else{
        // INRESPOND:
        /*
        stdClass Object(
            [label] => http:// // Link to download the PDF file with labels.
            [box] => // A list of boxes barcodes in the parcel ;
            [box_upload] => // A list of uploaded boxes barcodes in the parcel.
        )
        */
    }
?>
```

Integration example for 1C: Enterprise

Procedure **ParselCheckClick**(Element)

```
Proxy = WSCсылки.Parsel.СоздатьWSПрокси("https://api.boxberry.ru", "BoxBerry", "BoxBerrySoap");

// Request: Attempting
Result = Прокси.ParselCheck("Your API token", "XXXXXX"); Сообщить(СокрЛП(Результат));
Exception
Notify(ОписаниеОшибки());
КонецПопытки;
КонецПроцедуры
```

4.9. ParseList. List of tracking codes.

It allows to get a list of all tracking codes that were created in the Personal account but not included in any Act of acceptance of delivery.

The string immediately takes the form necessary for integration into the `ParseSend` method.

In examples XXXXXX means the tracking code.

Attention! The service works only for orders created via API.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token

Response data:

Parameter	Type	Description
ImIds	string	The list of tracking numbers as a string, separated by commas

Table 21. ParseList method

Protocol	Order format / implementing example
JSON	<p>Request: https://api.boxberry.ru/json.php?token=YourToken&method=ParseList</p> <p>Integration example for PHP-version above 5.0:</p> <pre><? \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ParseList'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data['err']){ // if something went wrong and the response has not been received: echo \$data['err']; }else{ // everything is correct, the response has been received and sent to \$data array: /* \$data=array('ImIds'=>'XXXXXXXX,XXXXXXXX,XXXXXXXX'); */ }</pre> <p>?></p>
SOAP	<p>Integration example for PHP-version above 5.0</p> <pre><? ini_set("soap.wsdl_cache_enabled", "0");// disable WSDL caching. try { \$clientS = new SoapClient('https://api.boxberry.ru/__soap/1c_lc.php?wsdl'); } catch (SoapFault \$exception){ \$errS=\$exception; } \$SDATA=array(); \$SDATA['token']='YourToken'; try { \$get_data=\$clientS->ParseList(\$SDATA); } catch (SoapFault \$exception){ \$errS=\$exception; } if(@\$errS){ echo \$errS; // if something went wrong and the response has not been received. }else{</pre>

```
// in response
/*
  stdClass Object(
    [ImIds] => XXXXXXX,XXXXXXX,XXXXXXX,XXXXXXX
  )
*/
}
```

Integration example for 1C: Enterprise

Procedure ParseListClick(Element)

```
Proxy = WSCсылки.ParseList.СоздатьWSПрокси("https://api.boxberry.ru", "BoxBerry", "BoxBerrySoap");

// Request Attempting
Result = Прокси.ParseList("Your API token");
Сообщить(СокрЛП(Результат));
Exception
  Notify(ОписаниеОшибки());
КонецПопытки;
```

КонецПроцедуры

4.10. ParselStory

It allows to get the list of parcels created via API. If you do not specify date intervals, the last created order will be returned.

Use parameters from (period from) and to (period to) as year-month-day to define the date range. Attention! The service works only for orders created via PA API.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
from	No	integer	Period "from" as YYYYMMDD
to	No	integer	Period "from" as YYYYMMDD

Response data:

Parameter	Type	Description
Shipment array		
track	string	Tracking code of the parcel
label	string	Link to download the PDF file with labels
date	string	To create a parcel as year-month-day hour-minute-second
send	bool	Sign of sending the order to the act
barcode	string	Parcel barcode
imid	string	The order number given by IS

Table 22. ParselStory method

Protocol	Order format / implementing example
JSON	<p>Request:</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselStory – the latest created order</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselStory&from=YYYYMMDD – orders created from the date indicated</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselStory&to=YYYYMMDD – orders created after the date indicated</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselStory&from=YYYYMMDD&ParselStory_&to=YYYYMMDD – orders created in the specified date range</p> <p>Integration example for PHP-version above 5.0:</p> <pre><? \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ParselStory'; \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$content,true); if(count(\$data)<=0 or \$data['err']) { // if something went wrong and the response has not been received: echo \$data['err']; } else { // everything is correct, the response has been received and sent to \$data array, // (no label if the barcode is given): /* \$data[0..n]=array('track'=>'XXXXXXXXXX', // Tracking code for the parcel, 'label'=>'http://', // Link to download the PDF file with labels, 'date'=>'2015.12.24', // date of the order creation as year-month-day); */ } </pre> <p>HH:MM:SS.</p>

	<pre> 'send'=>'true / false', // a sign of sending the order to the act, 'barcode'=>'XXXXXXXXXX', // parcel barcode, 'imid'=>'XXXXXXXXXX' // The order number given by IS.); */ } ?> </pre>
<p>SOAP</p>	<p>Integration example for PHP-version above 5.0</p> <pre> <? ini_set("soap.wsdl_cache_enabled", "0"); // disable WSDL caching. try { \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_1c.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); } catch (SoapFault \$exception){ \$errS=\$exception; } \$SDATA=array(); \$SDATA['token']='YourToken'; \$SDATA['from']=YYYYMMDD; \$SDATA['to']= YYYYMMDD; try { \$get_data=\$clientS->ParselStory(\$SDATA); } catch (SoapFault \$exception){ \$errS=\$exception; } if(@\$errS){ echo \$errS; } // if something went wrong and the response has not been received. else { // INRESPOND: /* (no label if the barcode is given). stdClass Object[result][0..n]=> { stdClass Object ([track] => Tracking code for the parcel, [label] => Link to download the PDF file with labels, [date] => date of the order creation as year-month-day hour-minute-second, [send] => sign of sending the parcel to the act, [barcode] => parcel barcode, [imid] => The order number given by IS,) } */ } ?> </pre>

4.11. ParselSendStory

It allows to get a list of acts of acceptance of delivery created. If you do not specify date intervals, the last created act will be returned.

Use parameters from (period from) and to (period to) as year-month-day to define the date range. Attention! The service works with any parcels.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	API-service access token
from	No	string	Period "from" as YYYYMMDD
to	No	string	Period "to" as year-month-day

Response data:

Parameter	Type	Description
Acts of acceptance of delivery data array		
track	string	A list of tracking codes of the parcels in the act
label	string	A link to download the act, if available
data	string	Date of the act creation as year-month-day hour-minute-second

Table 23. ParselSendStory method

Protocol	Order format / implementing example
JSON	<p>Request:</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselSendStory – the latest act created</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselSendStory&from=YYYYMMDD – acts created from the date indicated</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselSendStory&to=YYYYMMDD - acts created after the date indicated</p> <p>https://api.boxberry.ru/json.php?token=YourToken&method=ParselSendStory&from=YYYYMMDD&=ParselSendStory&to=YYYYMMDD – acts created in the specified date range</p> <pre> <? \$url='https://api.boxberry.ru/json.php?token=YourToken&method=ParselSendStory'; \$handle = fopen(\$url, "rb"); \$contents = stream_get_contents(\$handle); fclose(\$handle); \$data=json_decode(\$contents,true); if(count(\$data)<=0 or \$data['err']){ // if something went wrong and the response has not been received: echo \$data['err']; }else{ // everything is correct, the response has been received and sent to \$data array: /* \$data[0..n]=array('track'=>'XXXXXXX,XXXXXXX,XXXXXXX', // a list of tracking codes of the parcels in the act, 'label'=>'http://', // a link to download the act, if available, 'date'=>'2015.11.25' // date of the order creation as year-month-day hour-minute-second.); */ } ?> </pre>
SOAP	<p>Integration example for PHP-version above 5.0</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); // disable WSDL caching try { </pre>

```

        $clientS = new SoapClient('https://api.boxberry.ru/ soap/1c_1c.php?wsdl', array('features' =>
SOAP_SINGLE_ELEMENT_ARRAYS));
    } catch (SoapFault $exception){
        $errS=$exception;
    }

    $SSDATA=array();
    $SSDATA['token']='YourToken';
    $SSDATA['from']=YYYYMMDD;
    $SSDATA['to']=YYYYMMDD;

    try {
        $get_data=$clientS->ParseSendStory($SSDATA);
    } catch (SoapFault $exception){
        $errS=$exception;
    }
    if(@$errS){
        echo $errS;
    } // if something went wrong and the response has not been received.
    else
    {
        // INRESPOND:
        /*

        stdClass Object[result][0..n]=>
        {
            stdClass Object
            (
                [track] => a list of tracking codes of the parcels in the act,
                [label] => a link to download the act, if available,
                [date] => date of the order creation as year-month-day hour-minute-second,

            )
        }
        */
    }
}

```

4.12. OrdersBalance

It allows to get information about the orders that have actually been sent to Boxberry to be delivered, but have not been delivered to a client.

Attention! The service works only with acts created via PA API.

Input parameters:

Parameter	Mandatory	Type	Description
token	Yes	string	Token for the method access
OnlyPostpaid	No	integer	A list of postpaid orders only is returned. Possible values: 0 - a list of prepaid and postpaid orders is returned. 1 - a list of postpaid orders only is returned.

Response data:

Description	Type	Description
result	array	Array
ID	string	The order number given by IS
Status	string	Order status
Price	string	Declared value
Delivery_sum	string	Delivery cost
Payment_sum	string	Sum to be paid

Table 24. OrdersBalance method

Protocol	Order format / implementing example
	The structure of an incoming array:
	<pre> Array([token] => YourToken) </pre>
	<p>Response data:</p> <pre> Array([Result] => Array([0] => Array([ID] => The order number given by IS, [Status] => Order status, [Price] => Goods price, [Delivery_sum] => Delivery cost, [Payment_sum] => Sum to be paid,))) </pre>
JSON	<p>Integration example for PHP-version above 5.0</p> <pre> <? \$url = 'https://api.boxberry.ru/json.php?token=YourToken&method=OrdersBalance'; // token - Your API token: \$handle = fopen(\$url, "rb"); \$content = stream_get_contents(\$handle); fclose(\$handle); \$data = json_decode(\$content, true); if (count(\$data) <= 0 or \$data['err'] { // if something went wrong and the response has not been received: echo \$data['err']; } else { /* * print_r(\$data); */ // everything is correct, the response has been received and sent to \$data array, // information about order balance: /* </pre>

	<pre> Array([0..n] => Array([ID] => The order number given by IS, [Status] => Order status, [Price] => Goods price, [Delivery_sum] => Delivery cost, [Payment_sum] => Sum to be paid.),) */ } ?> </pre>
<p>SOAP</p>	<p>Integration example for PHP-version above 5.0</p> <pre> ini_set("soap.wsdl_cache_enabled", "0"); \$clientS = new SoapClient("https://api.boxberry.ru/__soap/1c_lc.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); \$s = array(); \$s['token'] = 'YourToken'; \$data = array(); try { \$data = json_decode(json_encode(\$clientS->OrdersBalance(\$s)), true); } catch (SoapFault \$exception) { \$errS = \$exception; } if (@\$errS) { echo \$errS; } // if something went wrong and the response has not been received. else { /* print_r(\$data); */ // everything is correct, the response has been received and sent to \$data array, // information about order balance: /* array('result' => array([0..n] => Array([ID] => The order number given by IS, [Status] => Order status, [Price] => Goods price, [Delivery_sum] => Delivery cost, [Payment_sum] => Sum to be paid,),)) */ } </pre>

5. Frequently asked questions

For the most frequently asked questions, please click on [the following link](#).

6. Additionally

If you encounter encoding problems when using the JSON service:

You need to parse the system response as JSON. E.g. in php `json_decode()` and `json_encode()`, which convert JSON data types into the corresponding PHP types and vice versa.

The sequence `\uXXXX` is a character that has the code `XXXX` encoded in UTF-8. Russian characters are encoded in JSON format this way (this service works in UTF-8 encoding only). This is solved at the stage of parsing JSON - for example, using `.parseJSON` (JQUERY) or `json_decode` (PHP) methods.

If you receive an authorization error when accessing the service, we recommend to use Content-Type: `application/x-www-form-urlencoded`.