boxberry

Description of web-services

Version 2.1

Contents

Introduction

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

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

You can receive relevant information about the cities of delivery, pickup points, delivery time, availability of courier delivery service by the receiver's ZIP code, accrued services, and also track the status of the parcel and transfer the parcel data to our information system.

Administrator toolbar:
http://api.boxberry.de/?act=auth

Location of web-services
JSON: http://api.boxberry.de/json.php
SOAP: http://api.boxberry.de/__soap/1c_public.php?wsdl
http://api.boxberry.de/__soap/1c_lc.php?wsdl

Sandbox

URL: http://test.api.boxberry.de

API-token: 13450.rbpqbaef

Contents

Methods used to receive data

JSON Url: http://api.boxberry.de/json.php

SOAP Url: http://api.boxberry.de/__soap/1c_public.php?wsdl

Datatable 1. JSON, SOAP protocol' available methods

Num	Method name	Description
1	ListCities	Method returns list of cities where Boxberry pickup points are located. Read more at page 5
2	ListPoints	Method returns information about all Boxberry pickup points. Read more at page 7
3	ListZips	Method returns list of ZIP codes where courier delivery is available. Read more at page 10
4	ZipCheck	Method allows to check courier delivery availability for specified ZIP code. Read more at page 12
5	GetStatusesForeign	Method returns parcel status information by specified tracking number via ImId parameter Read more at page 14
6	DeliveryCostsF	Method allows to calculate delivery price (to Boxberry pickup point or courier delivery). Read more at page 15
7	PointsByPostcode	Method returns list of pickup points codes by ZIP code. Read more at page 17
8	PointsDescription	Method returns detailed information about specified pickup point, Read more at page 18
9	ListCountry	Method returns information about all countries where Boxberry delivery service is available Read more at page 24

ListCities

ListCities method returns list of cities where Boxberry pickup points are located.

Datatable 2. Method ListCities

<table>
<tr><td>

JSON

</td><td>

Url: **http://api.boxberry.de/json.php?token= &method=ListCities&CountryCode=643**

</td></tr>
<tr><td>

Request GET method

</td><td>

```
<?php

        $url='http://api.boxberry.de/json.php?token= &method=ListCities&CountryCode=643';
        $handle = fopen($url, "rb");
        $contents = stream_get_contents($handle);
        fclose($handle);
        $data=json_decode($contents,true);
        if(count($data)<=0 or $data[0]['err'])
        {
           // if an error occurred and the response has not been received.
           echo $data[0]['err'];
        }
        else
        {
           // everything is fine, the response is received, now in the array $data
           // list of cities where boxberry service points are located:
           /*
           $data[0...n]=array(
              'Name'=>'City name',
              'Code'=>'Boxberry city code'
              'CountryCode' => 'Country code',
              'Prefix' => 'Prefix: г - City, п – Village, etc.',
              'ReceptionLaP' => 'Letters reception is available if 1',
              'DeliveryLaP' => 'Letters’ issue is available if 1',
              'Reception' => 'Reception for MiM is available if 1',
              'ForeignReceptionReturns' => 'International returns' reception is available if 1',
              'Terminal' => 'Presence of sorting terminal',
              'Kladr' => 'Kladr ID',
              'Region' => 'Region',
              'UniqName' => 'Unique name',
              'District' => 'District'
           );

           Example:
           echo $data[0]['Name'];
           echo $data[5]['Code'];
           echo $data[0]['CountryCode'];
```

</td></tr>
</table>

Contents

```
        */
      }
      ?>
```

SOAP Request

```
Array ( [token] => API-token, )
```

Response

```
stdClass Object
        (
          [result] => Array
          (
           [0...n] => stdClass Object
           (
             [Name]=>'City name',
             [Code]=>'Boxberry city code',
             [Prefix] => 'Prefix: г - City, п – Village, etc.', [ReceptionLaP] => 'Letters
             reception is available if 1', [DeliveryLaP] => 'Letters' issue is available if 1',,
             [Reception] => 'Reception for MiM is available if 1' [ForeignReceptionReturns]
             => 'International returns' reception is available if 1' [Terminal] => 'Presence of
             sorting terminal', [Kladr] => 'Kladr ID',


             [Region] => 'Region',
             [CountryCode] => 'Country code',
             [UniqName] => 'Unique name',
             [District] => 'District'
           )
          )
        )
```

Example for PHP 5.0 or later version

```
ini_set("soap.wsdl_cache_enabled", "0");
      $clientS = new SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl",
      array('features' => SOAP_SINGLE_ELEMENT_ARRAYS));

      $s=array();
      $s['token']='APItoken;

      try {
       $data=$clientS->ListCities($s);
       }
      catch (SoapFault $exception){
```

	$errS=$exception;
	}
	
	if(isset($errS)){
	echo $errS;
	} // if an error occurred and the response has not been received. else
	{
	// everything is fine, the response is received, now in the array $data. print_r($data);
	}

ListPoints

ListPoints method returns information about all Boxberry pickup points. By using additional parameter ("CityCode" - Boxberry city code) returns information about Boxberry pickup points by specified city. With parameter "CountryCode" (iso 3166-1 numeric) returns pickup points by specified country.

Important!

To get the list of all pickup points, please, use parameter "prepaid" with value 1.

Datatable 3. Method ListPoints

JSON	Url: **http://api.boxberry.de/json.php?token= &method=ListPoints** **http://api.boxberry.de/json.php?token= &method=ListPoints&CityCode=68&prepaid=1** **http://api.boxberry.de/json.php?token= &method=ListPoints&prepaid=1&CountryCode=643**

Request GET method	<?php
	
	$url='http://api.boxberry.de/json.php?token= &method=ListPoints';
	$handle = fopen($url, "rb");
	$contents = stream_get_contents($handle);
	fclose($handle);
	$data=json_decode($contents,true);
	if(count($data)<=0 or $data[0]['err'])
	{
	// if an error occurred and the response has not been received.
	echo $data[0]['err'];
	}
	else

```
    {
     // everything is fine, the response is received, now in the array $data
     // list of all boxberry service points in following format:
     /*
     $data[0...n]=array(
         'Code'=>'Service point code',
         'Name'=>'Service point name',
         'Address'=>'Full address',
         'Phone'=>'Phone number',
         'WorkSchedule'=>'Working hours',
         'TripDescription'=>'How to reach boxberry service point',
         'DeliveryPeriod'=>'Delivery time',
         'CityCode'=>'Boxberry city code',
         'CityName'=>'City name',
         'TariffZone'=>'Tariff zone',
         'Settlement'=>'Settlement',
         'Area'=>'Region',
         'Country'=>'Country',
         'GPS'=>'GPS coordinates',
         'OnlyPrepaidOrders'=>'If the value is "Yes" - a point only works with fully paid orders',
         'Acquiring'=>'If value "YES" - pickup point can accept payments with the Debit/Credit cards',
         'DigitalSignature'=>'If value "YES" - Signature of recipient will be stored in boxberry system digitally',
         'CountryCode' => 'Country code',
     );

     Example:
     echo $data[0]['Name'];
     echo $data[5]['Code'];
     echo $data[0]['CountryCode'];
     */
    }
    ?>
```

SOAP Request

```
Array ( [token] =>, [CityCode] => 0, [prepaid] => 0 )
```

Response

```
stdClass Object
    (
      [result] => Array
      (
        [0...n] => stdClass Object
        (
          [Code] => Service point code,
          [Name] => Service point name,
          [Address] => Full address,
          [Phone] => Phone number,
          [WorkSchedule] => Working hours,
          [TripDescription] => How to reach boxberry service point,
```

Contents

```
                [DeliveryPeriod] => Delivery time,
                [CityCode] => Boxberry city code,
                [CityName] => City name,
                [TariffZone] => Tariff zone,
                [Settlement] => Settlement,
                [Area] => Region,
                [Country] => Country,
                [GPS] => GPS coordinates,
                [OnlyPrepaidOrders] => If the value is "Yes" - a point only works with fully paid orders,
                [Acquiring] => If value "YES" - pickup point can accept payments with the Debit/Credit cards,
                [DigitalSignature] => If value "YES" - Signature of recipient will be stored in boxberry system
digitally,
                [TypeOfOffice] => PVZ / SPVZ - the type of point of delivery,
                [NalKD] => Yes / No - the presence of express delivery,
                [Metro] => The nearest metro station, [CountryCode] =>
                Country code,
              )
            )
          )
```

Example for PHP 5.0 or later version

```
ini_set("soap.wsdl_cache_enabled", "0");
        $clientS = new SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl",
                    array('features' => SOAP_SINGLE_ELEMENT_ARRAYS));

        $s=array();
        $s['token']='APItoken';
        $s['code']=0;
        $s['prepaid']=0;

        try {
         $data=$clientS->ListPoints($s);
         }
        catch (SoapFault $exception){
         $errS=$exception;
         }

        if(isset($errS)){
         echo $errS;
         } // if an error occurred and the response has not been
        received. else
        {
          print_r($data);
        }
```

ListZips

ListZips method returns list of ZIP-codes where courier delivery is available.

Important!

List of ZIP codes is about 14 MB.

Datatable 4. Method ListZips

<table>
<tr><td><b><i>JSON</i></b></td><td>Url: http://api.boxberry.de/json.php?token=YourAPItoken&method=ListZips</td></tr>
<tr><td><b><i>Request
GET
method</i></b></td><td>

```
<?php

    $url='http://api.boxberry.de/json.php?token=YourAPItoken&method=ListZips';
    $handle = fopen($url, "rb");
    $contents = stream_get_contents($handle);
    fclose($handle);
    $data=json_decode($contents,true);
    if(count($data)<=0 or $data[0]['err'])
    {
      // if an error occurred and the response has not been received.
      echo $data[0]['err'];
    }
    else
    {
      // everything is fine, the response is received, now in the array $data
      // list of all postal codes in following format:
      /*
      $data[0...n]=array(
          'Zip'=>'Postal code',
          'City'=>'City',
          'Region'=>'Region',
          'Area'=>'Area',
          'ZoneExpressDelivery'=>'Tariff zone'
      );

      Example:
      echo $data[0]['Zip'];
      echo $data[5]['City'];
      */
    }
    ?>
```

</td></tr>
</table>

SOAP Request

```
Array ( [token] => YourAPItoken, )
```

Response

```
stdClass Object
          (
            [result] => Array
            (
              [0...n] => stdClass Object
              (
                [Zip] => Postal code,
                [City] => City,
                [Region] => Region,
                [Area] => Area,
                [ZoneExpressDelivery] => Tariff zone,
              )
            )
          )
```

Example for PHP 5.0 or later version

```
ini_set("soap.wsdl_cache_enabled", "0");
        $clientS = new SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl",
                    array('features' => SOAP_SINGLE_ELEMENT_ARRAYS));

        $s=array();
        $s['token']='YourAPItoken';

        try {
         $data=$clientS->ListZips($s);
         }
        catch (SoapFault $exception){
         $errS=$exception;
         }

        if(isset($errS)){
         echo $errS;
         } // if an error occurred and the response has not been received.
        else
        {
          print_r($data);
        }
```

ZipCheck

ZipCheck method allows to check availability of courier delivery for specified ZIP code. Required parameter ("zip" - ZIP code to be checked)

Datatable 5. Method ZipCheck

JSON	Url: **http://api.boxberry.de/json.php?token= &method=ZipCheck&Zip=101000&CountryCode=643**

Request GET method

```
$url='http://api.boxberry.de/json.php?token=&method=ZipCheck&Zip=101000&CountryCode=643';
    $handle = fopen($url, "rb");
    $contents = stream_get_contents($handle);
    fclose($handle);
    $data=json_decode($contents,true);
    if(count($data)<=0 or $data[0]['err'])
    {
      // if an error occurred and the response has not been received.
      echo $data[0]['err'];
    }
    else
    {
      // everything is fine, the response is received, now in the array $data
      // Element with the following data:
      /*
      $data[0]=array(
       'ExpressDelivery'=>'1 - Delivery available, 0 - Delivery unavailable',
       'ZoneExpressDelivery'=>'Tariff zone'
      );

      Example:
      echo $data[0]['ExpressDelivery'];
      */
    }
```

Contents

SOAP Request	`Array ( [token] => YourAPItoken, [zip] => XXXXXX )`
Response	stdClass Object ([result] => stdClass Object ([ExpressDelivery] => 1 - Delivery available, 0 - Delivery unavailable, [ZoneExpressDelivery] => Tariff zone.)) ini_set("soap.wsdl_cache_enabled", "0"); $clientS = new SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)); $s=array(); $s['token']='YourAPItoken';

Example for PHP 5.0 or later version	ini_set("soap.wsdl_cache_enabled", "0");
	$clientS = new SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl", array('features' => SOAP_SINGLE_ELEMENT_ARRAYS));
	
	$s=array();
	$s['token']='YourAPItoken';
	$s['zip']=XXXXXX;
	
	try {
	$data=$clientS->ZipCheck($s);
	}
	catch (SoapFault $exception){
	$errS=$exception;
	}
	
	if(isset($errS)){
	echo $errS;
	} // if an error occurred and the response has not been received. else
	{
	print_r($data);
	}

GetStatusesForeign

ListStatuses method returns parcel status information by specified tracking number via ImId parameter.

Datatable 6. Method GetStatusesForeign

<table>
<tr><td><b><i>JSON</i></b></td><td>Url: <b>http://api.boxberry.de/json.php?token=YourAPItoken&method=GetStatusesForeign&ImId=XXXXXX</b></td></tr>
<tr><td><b><i>Request GET Method</i></b></td><td>

```
   <?php
$url='http://api.boxberry.de/json.php?token=YourAPItoken&method=GetStatusesForeign&ImId=XXXXXX';
        // XXXXXX - Tracking number.
        $handle = fopen($url, "rb");
        $contents = stream_get_contents($handle);
        fclose($handle);
        $data=json_decode($contents,true);
        if(count($data)<=0 or $data[0]['err'])
        {
          // if an error occurred and the response has not been received.
          echo $data[0]['err'];
        }
        else
        {
          // everything is fine, the response is received, now in the array $data
          // all parcel statuses in following format:
          /*
          $data[0...n]=array(
            'Status'=>'Status code',
            'Name'=>'Status name',
            'Date'=>'Date of status in following format DD/MM/YY Hours:Minutes:Seconds PM/AM',
            'Comment'=>'Comment to status'
            );

            Example:
            echo $data[0]['Date'];
            echo $data[0]['Name'];
            */
          }
          ?>
```

</td></tr>
</table>

DeliveryCostsF

DeliveryCostsF method allows to calculate delivery cost (to Boxberry pickup point or for courier delivery).

Datatable 7. Method DeliveryCostsF

JSON	Url: http://api.boxberry.de/json.php?token=&method=DeliveryCostsF&weight=500&type=1&target=010&ordersum=0&insurance=0

Request GET method	<?php
	/*
	Incoming parameters:
	weight - Parcel weight in grams,
	type - Delivery type (1 - boxberry pickup point, 2 - Courier delivery (KD)),
	target - pickup point code or zip code for courier delivery type=2,
	ordersum - Parcel value (0 if empty),
	insurance - Additional services (if insurance applied).
	*/
	$url='http://api.boxberry.de/json.php?token=
	&method=DeliveryCostsF&weight=500&type=1&target=010&ordersum=0&insurance=0';
	
	$handle = fopen($url, "rb");
	$contents = stream_get_contents($handle);
	fclose($handle);
	$data=json_decode($contents,true);
	if(count($data)<=0 or $data[0]['err'])
	{
	// if an error occurred and the response has not been received.
	echo $data[0]['err'];
	}
	else
	{
	// everything is fine, the response is received, now in the array $data
	/*
	array(
	['price'] => 'Total delivery price incl. additional services:;
	['price_base'] => 'Delivery price';
	['price_service'] => 'Insurance, by wish of client (1 - yes, 0 - no (0 is empty))';
	['tariff_zone'] => 'Tariff zone';
	)
	*/
	}
	?>

SOAP Request

```
             Array
             (
               [token] => YourAPItoken,
               [weight] => Parcel weight in grams,
               [type] => Delivery type (1 - boxberry pickup point, 2 - Courier delivery
(KD)),
               [target] => pickup point code or zip code for courier delivery type=2,
               [ordersum] => Parcel value (0 if empty),
               [insurance] => Additional services (if insurance applied).
             )
```

Response

```
stdClass Object
     (
      [result] => stdClass Object
      (
       [price] => Total delivery price incl. additional services:,
       [price_base] => Delivery price,
       [price_service] => Insurance, by wish of client (1 - yes, 0 - no (0 is
       empty)), [tariff_zone] => Tariff zone,
      )
     )
```

Example for PHP 5.0 or later version

```
ini_set("soap.wsdl_cache_enabled", "0");
     $clientS = new SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl",
            array('features' => SOAP_SINGLE_ELEMENT_ARRAYS));

     $s=array();
     $s['token']='YourAPItoken';
     $s['weight']=500;
     $s['type']=1;
     $s['target']='010';
     $s[ordersum] => 0;
     $s[insurance] => 0;

     try {
      $data=$clientS->DeliveryCostsF($s);
      }
     catch (SoapFault $exception){
      $errS=$exception;
      }

     if(isset($errS)){
      echo $errS;
      } // if an error occurred and the response has not been received.
     else
     {
       print_r($data);
     }
```

PointsByPostcode (legacy)

PointsByBostCode method returns list of pickup points’ codes by ZIP code.

Datatable 8. Method PointsByPostcode

JSON	Url: http://api.boxberry.de/json.php?token=YourAPItoken&method=PointsByPostCode&zip=XXXXXX

Request GET method	<?php
	$url =
	'http://api.boxberry.de/json.php?token=YourAPItoken&method=PointsByPostCode&zip=XXXXXX';
	// XXXXXX - Postal 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 an error occurred and the response has not been received.
	echo $data['err'];
	} else {
	// everything is fine, the response is received, now in the array $data
	// Service point code:
	/*
	$data[Code] => "Service point code"
	*/
	}
	?>

SOAP Request	`Array ( [token] => YourAPItoken, [zip] => XXXXXX )`
Response	Array ([result] => Array ([code] => XXXX))
	ini_set("soap.wsdl_cache_enabled", "0"); $clientS = new SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl",

Example for PHP 5.0 or later version

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

    $s = array();
    $s['token'] = 'YourAPItoken';
    $s['zip'] = 'XXXX';

    $data = array();

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

    if (@$errS) {
        echo $errS;
    } // if an error occurred and the response has not been
    received. else {
        /*
          print_r($data);
        */
        // everything is fine, the response is received, now in the array $data
        // Service point code:
        /*
        Array(
          'result' => Array(
             "Code" => "Service point code"
          )
        )
        */
    }
```

PointsDescription

PointsDescription method returns information about all Boxberry pickup points including photos.

Datatable 9. Method PointsDescription

JSON	Url: http://api.boxberry.de/json.php?token=&method=PointsDescription&code=XXXXX&photo=1

Request GET method

```
   <?php
       $url =
'http://api.boxberry.de/json.php?token=YourAPItoken&method=PointsDescription&code=XXXXX&photo
=1';
       // token – Your API token,
       // code – Service point code,
       // photo – Get photo, if specified..
```

Contents

```
      $handle = fopen($url, "rb");
      $contents = stream_get_contents($handle);
      fclose($handle);
      $data = json_decode($contents, true);
      if (count($data) <= 0 or $data['err']) {
        // if an error occurred and the response has not been received.
        echo $data['err'];
      } else {
        /*
        print_r($data);
        */
        // everything is fine, the response is received, now in the array $data
        // detailed information about issuing point:
        /*
          Array(
              [Name] => Service point code,
              [Organization] => Organization,
              [ZipCode] => Zip,
              [Country] => Country,
              [Area] => Region,
              [CityCode] => Boxberry city code,
              [CityName] => City,
              [Settlement] => Inhabited locality,
              [Metro] => Metro Station,
              [Street] => Street,
              [House] => House number,
              [Structure] => Structure,
              [Housing] => House,
              [Apartment] => Apartment/Office,
              [Address] => Full address,
              [AddressReduce] => Short address,
              [GPS] => GPS coordinates,
              [TripDescription] => Information about the location,
              [Phone] => Phone number,
              [ForeignOnlineStoresOnly] => Only for international Internet shop,
              [PrepaidOrdersOnly] => Issuance of prepaid orders,
              [Acquiring] => Acquiring (the ability to receive credit cards, the presence of the terminal),
              [DigitalSignature] => Digital signature,
              [TypeOfOffice] => Type Of Issuing point: 1 - The point of issue orders, 2 - Third-party the
point of issue orders,
              [CourierDelivery] => Courier delivery,
              [ReceptionLaP] => Reception of letters,
              [DeliveryLaP] => Issuing of letters,
              [LoadLimit] => Limit weight, kg,
              [VolumeLimit] => Volume limit,
              [EnablePartialDelivery] => Enable partial delivery,
              [EnableFitting] => Enable fitting,
              [WorkShedule] => Work schedule,
              [WorkMoBegin] => Monday, the start of the working day,
              [WorkMoEnd] => Monday, the end of the working day,
              [WorkTuBegin] => Tuesday, the start of the working day,
              [WorkTuEnd] => Tuesday, the end of the working day,
              [WorkWeBegin] => Wednesday, the start of the working day,
              [WorkWeEnd] => Wednesday, the end of the working day,
                [WorkThBegin] => Thursday, the start of the working day,
                [WorkThEnd] => Thursday, the end of the working day,
                [WorkFrBegin] => Friday, the start of the working day,
                [WorkFrEnd] => Friday, the end of the working day,
                [WorkSaBegin] => Saturday, the start of the working day,
                [WorkSaEnd] => Saturday, the end of the working day,
                [WorkSuBegin] => Sunday, the start of the working day,
                [WorkSuEnd] => Sunday, the end of the working 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,
                [Photos] => Array with photos base64,
                // Data terminal, which is subject to separation.,
                // E.g. for Kamensk-Uralsky - This is the "Yekaterinburg (terminal)",
                // Data by analogy with the details of the point: Code, Name ... prefix Terminal.
                [TerminalCode] => Code of terminal,
                [TerminalName] => Name of terminal,
                [TerminalOrganization] => Terminal Organization’s name,
                [TerminalCityCode] => Code of terminal’s city,
                [TerminalCityName] => Name of terminal’s city,
                [TerminalAddress] => Terminal’s address,
                [TerminalPhone] => Terminal’s phone,
                [Reception] => Availability of parcel’s reception.
             )
         */
      }
      ?>
```

SOAP Request	Array ([token] => YourAPItoken, [code] => XXXXXX, [photo] => 1)

Response

```
Array
    (
      [result] => Array
      (
           [Name] => Service point code,
           [Organization] => Organization,
           [ZipCode] => Zip,
           [Country] => Country,
```

Contents

	[Area] => Region, [CityCode] => Boxberry city code, [CityName] => City, [Settlement] => Inhabited locality, [Metro] => Metro Station, [Street] => Street, [House] => House number, [Structure] => Structure, [Housing] => House, [Apartment] => Apartment/Office, [Address] => Full address, [AddressReduce] => Short address, [GPS] => GPS coordinates, [TripDescription] => Information about the location, [Phone] => Phone number, [ForeignOnlineStoresOnly] => Only for international Internet shop, [PrepaidOrdersOnly] => Issuance of prepaid orders, [Acquiring] => Acquiring (the ability to receive credit cards, the presence of the terminal), [DigitalSignature] => Digital signature, [TypeOfOffice] => Type Of Issuing point: 1 - The point of issue orders, 2 - Third-party the point of issue orders, [CourierDelivery] => Courier delivery, [ReceptionLaP] => Reception of letters, [DeliveryLaP] => Issuing of letters, [LoadLimit] => Limit weight, kg, [VolumeLimit] => Volume limit, [EnablePartialDelivery] => Enable partial delivery, [EnableFitting] => Enable fitting, [WorkShedule] => Work schedule, [WorkMoBegin] => Monday, the start of the working day, [WorkMoEnd] => Monday, the end of the working day, [WorkTuBegin] => Tuesday, the start of the working day, [WorkTuEnd] => Tuesday, the end of the working day, [WorkWeBegin] => Wednesday, the start of the working day, [WorkWeEnd] => Wednesday, the end of the working day, [WorkThBegin] => Thursday, the start of the working day, [WorkThEnd] => Thursday, the end of the working day, [WorkFrBegin] => Friday, the start of the working day, [WorkFrEnd] => Friday, the end of the working day, [WorkSaBegin] => Saturday, the start of the working day, [WorkSaEnd] => Saturday, the end of the working day, [WorkSuBegin] => Sunday, the start of the working day, [WorkSuEnd] => Sunday, the end of the working 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,
                [Photos] => Array with photos base64,
                // Data terminal, which is subject to separation.,
                // E.g. for Kamensk-Uralsky - This is the "Yekaterinburg (terminal)",
                // Data by analogy with the details of the point: Code, Name ... prefix Terminal.
                [TerminalCode] => Code of terminal,
                [TerminalName] => Name of terminal,
                [TerminalOrganization] => Terminal Organization’s name,
                [TerminalCityCode] => Code of terminal’s city,
                [TerminalCityName] => Name of terminal’s city,
                [TerminalAddress] => Terminal’s address,
                [TerminalPhone] => Terminal’s phone,
                [Reception] => Availability of parcel’s reception.
          )
        )
```

Example for PHP 5.0 or later version

```
<?php

    ini_set("soap.wsdl_cache_enabled", "0");
    $clientS = new
      SoapClient("http://api.boxberry.de/__soap/1c_public.php?wsdl",
      array('features' => SOAP_SINGLE_ELEMENT_ARRAYS)
    );

    $s = array();
    $s['token'] = 'YourAPItoken';
    $s['code'] = 'XXXX';
    $s['photo'] = '1';
    // token – Your API token.
    // code – Service point code,
    // photo – Get photo, if specified..

    $data = array();

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

    if (@$errS) {
      echo $errS;
    } // if an error occurred and the response has not been
    received. else {
      /*
        print_r($data);
      */
      // everything is fine, the response is received, now in the array $data
      // detailed information about issuing point:
      /*
        Array(
          [result] => Array(
```

[Name] => Service point code,
[Organization] => Organization,
[ZipCode] => Zip,
[Country] => Country,
[Area] => Region,
[CityCode] => Boxberry city code,
[CityName] => City,
[Settlement] => Inhabited locality,
[Metro] => Metro Station,
[Street] => Street,
[House] => House number,
[Structure] => Structure,
[Housing] => House,
[Apartment] => Apartment/Office,
[Address] => Full address,
[AddressReduce] => Short address,
[GPS] => GPS coordinates,
[TripDescription] => Information about the location,
[Phone] => Phone number,
[ForeignOnlineStoresOnly] => Only for international Internet shop,
[PrepaidOrdersOnly] => Issuance of prepaid orders,
[Acquiring] => Acquiring (the ability to receive credit cards, the presence of the terminal),
[DigitalSignature] => Digital signature,
[TypeOfOffice] => Type Of Issuing point: 1 - The point of issue orders, 2 - Third-party the point of issue orders,
[CourierDelivery] => Courier delivery,
[ReceptionLaP] => Reception of letters,
[DeliveryLaP] => Issuing of letters,
[LoadLimit] => Limit weight, kg,
[VolumeLimit] => Volume limit,
[EnablePartialDelivery] => Enable partial delivery,
[EnableFitting] => Enable fitting,
[WorkShedule] => Work schedule,
[WorkMoBegin] => Monday, the start of the working day,
[WorkMoEnd] => Monday, the end of the working day,
[WorkTuBegin] => Tuesday, the start of the working day,
[WorkTuEnd] => Tuesday, the end of the working day,
[WorkWeBegin] => Wednesday, the start of the working day,
[WorkWeEnd] => Wednesday, the end of the working day,
[WorkThBegin] => Thursday, the start of the working day,
[WorkThEnd] => Thursday, the end of the working day,
[WorkFrBegin] => Friday, the start of the working day,
[WorkFrEnd] => Friday, the end of the working day,
[WorkSaBegin] => Saturday, the start of the working day,
[WorkSaEnd] => Saturday, the end of the working day,
[WorkSuBegin] => Sunday, the start of the working day,
[WorkSuEnd] => Sunday, the end of the working 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,
            [Photos] => Array with photos base64,
            // Data terminal, which is subject to separation.,
            // E.g. for Kamensk-Uralsky - This is the "Yekaterinburg (terminal)",
            // Data by analogy with the details of the point: Code, Name ... prefix Terminal.
            [TerminalCode] => Code of terminal,
            [TerminalName] => Name of terminal,
            [TerminalOrganization] => Terminal Organization’s name,
            [TerminalCityCode] => Code of terminal’s city,
            [TerminalCityName] => Name of terminal’s city,
            [TerminalAddress] => Terminal’s address,
            [TerminalPhone] => Terminal’s phone,
            [Reception] => Availability of parcel’s reception.
             )
      )
   */
}
?>
```

ListCountry

ListCountry method allows to receive list of countries where Boxberry delivers to.

Datatable 10. Method ListCountry

JSON	Url: **http://api.boxberry.de/json.php?token=YourAPItoken&method=ListCountry**
Request GET method	<?php $url = "http://api.boxberry.de/json.php";

```
$query = [];
$query['token'] = 'YourAPItoken'; // Your API token,
$query['method'] = 'ListCountry';  // Name of method
$data = file_get_contents($url."?".http_build_query($query));

$data = json_decode($data,true);

if(!empty($data)){
   /*
   print_r($data);

   // everything is fine, the response is received, now in the array $data.
   Array
      (
         [0] => Array
            (
               [CountryCode] => 398    // Country code,
               [LangRu] => Казахстан   // Name in Russian,
               [LangEng] => Kazakhstan // Name in English.
            )

         [1] => Array
            (
               [CountryCode] => 643 [LangRu]
               => Россия [LangEng] => Russian
               Federation
            )

         [2] => Array
            (
               [CountryCode] => 804
               [LangRu] => Украина
               [LangEng] => Ukraine
            )
      )
   */

}
?>
```

SOAP

Example for PHP 5.0 or later version

```
<?php

        $url = "http://api.boxberry.de/__soap/1c_public.php?wsdl";

        $query = [];
        $query['token'] = 'YourAPItoken'; // Your API token.

        $data = array();

        try{
           $client = new SoapClient($url);
```

```
  $data = json_encode($client->ListCountry($query));
} catch (SoapFault $ex){
  //print $ex->getMessage();
}

$data = json_decode($data,true);

if(!empty($data)){
  /*
  print_r($data);

  // everything is fine, the response is received, now in the array
  $data. Array
  (
    [0] => Array
    (
      [CountryCode] => 398    // Country code,
      [LangRu] => Казахстан   // Name in Russian,
      [LangEng] => Kazakhstan // Name in English.
    )

    [1] => Array
    (
      [CountryCode] => 643
      [LangRu] => Россия
      [LangEng] => Russian Federation
    )

    [2] => Array
    (
      [CountryCode] => 804
      [LangRu] => Украина
      [LangEng] => Ukraine
    )
  )
  */
}
?>       }
```

Methods used to transfer data

JSON Url: http://api.boxberry.de/json.php

SOAP Url: http://api.boxberry.de/__soap/1c_lc.php?wsdl

Datatable 11. Methods used to transfer data

Num	Method name	Description
1	ParcelCreateForeign	Method creates parcel in Boxberry system. Read more at page 29
2	ParselCheck	Method allows to get the link to the file for printing labels, the list of barcodes in parcel separated with commas (,), the list of barcodes for boxes in parcel separated with comma (,). Parameter with tracking code required. Read more at page 34
3	ParselSend	Method uploads parcels' data to the Boxberry system. Read more at page 35
4	ParselReadiness	Method informs if all the necessary parcel data is collected or not. Read more at page 37
5	ParselDel	Method deletes parcels' data, created via API (before request to ParselSend). Read more at page 38
6	ParselSendForeign	Method uploads parcels' data to Boxberry system. It's necessary to send encoded JSON array by POST. Description of the array please find below. In case of successful result the method will return JSON-encoded array with identification number. Read more at page 40
7	ParcelUpdateForeign	Method updates parcel, previously created via api.boxberry.de Read more at page 40

ParcelCreateForeign

ParcelCreateForeign method creates parcels' data in Boxberry system. It is necessary to transfer data to array by POST method. Description of array is below. Method will return JSON encoded array with Identification number (tracking number), link to the labels and the list of boxes barcodes in the parcel separated with commas(,).

Datatable 12. Method ParselCreateForeign

JSON	Url: **http://api.boxberry.de/json.php**
Request POST method	<?php $SDATA = []; $SDATA = array('method' => 'ParcelCreateForeign', // Name of method (required) 'token' => 'YourAPItoken', // Your API token (required) 'u_name' => 'Иван', // Name (required) 'u_surname' => 'Иванов', // Last name (required) 'u_middlename' => 'Иванович', // Middle name (optional) 'u_email' => 'air@email.com', // EMAIL (required) 'u_phone' => '+7 XXX XXX XX XX', // Receiver's phone number (required) 'u_country_code' => 643, // Code of issuing state. By default 643 - Russia. (required) 'u_pasport' => 'XXXX XXXXXX', // Number of identification document (required) 'u_passportIssued'=>'2017-01-01', // Identification document's date of issue (required) 'u_passportIssuedBy'=>'', // Identification document issued by (required) 'u_destination_country_code' => '', // The code of the destination country (required) 'u_city' => "", // Receiver's city (required) 'u_take' => 'kd', // kd if courier delivery or Pick-up point code (required) 'u_post_code' => 'XXXXXX', // XXXXXX - Postal code (optional, if delivery type – pick-up point) 'u_street' => '5-я Курская, д.8 кв.16', // Address of the receiver (optional, if delivery type – pick-up point) 'Order' => '123456789', // Order number in the Internet shop (optional, Sender tracking is assigned if empty) 'sender_tracking' => '', // Tracking number in the Internet shop (optional, Boxberry tracking is assigned if empty) 'barcode_pallets' => 'XXXXXXXXXXX', // Field for transmitting information on barcode pallets (optional) 'barcode_bigbox' => 'XXXXXXXXXXX', // Field for transmitting information on barcode boxes (big with small packets inside) (optional)

```
            'extraPassData' => 'XXXXXXXXXXX',      // Additional data if necessary (e.g. INN, subdivision code)
(optional)
            'box' => array(                   // Array of boxes with goods
               array(                    // Array of one box, not more than 5 boxes in one parcel
                  'x' => '10',               // Dimensions of the box, cm
                  'y' => '20',
                  'z' => '40',
                  'weight_bruto' => '515',       // Gross weight of the box in grams (required)
                  'add_tracking_code' => '',      // Additional tracking code to the sender's personal needs
(optional)
                  'add_barcode128' => 'XXXXXXXXX',// Barcode in the Internet shop for printing on the label
(Derivation on the label – EAN, optional)
                           'seller' => array(           // For Type IM = Mediator
                     'exporter_number' => 'exporter_number',      // Number for export clearance (required)
                     'legal_name' => 'legal_name',               // Legal name of the seller (required)
                     'web_address' => 'web_address',             // Seller's common name (optional)
                     'address' => 'Address',                  // Seller's address (required)
                     'post_code' => 'post_code',                 // Seller's Zip-code (required)
                     'city' => 'city',                     // Seller's city (required)
                     'country_code' => 643,                   // Seller's country code (required)
                  ),

                  'items' => array(                        // Array of goods in the box
                     array(                           // Array of one unit of goods, can be any amount
                        'articul' => '1232212',              // Article in the shop (required)
                        'Manufacturer' => 'Hitachi',           // Manufacturer (optional, No name if empty)
                        'model' => 'DLS-118',               // Item name (required)
                        'quantity' => '1',                 // Quantity (required)
                        'bruto' => '500',                  // Gross weight (gr) (required)
                        'item_price' => '200',               // Price in supplier's currency (required)
                        'currency_price' => 'EUR',             // Supplier's currency (EUR GBP USD RUB). If incorrect
or field is empty, than EUR (required)
                        'web_address' => ''                // Link to the manufacturer
                        'link_web' => '',                  // Link to the item of goods in the internet (required)
                        'link_foto' => '',                 // Link to the item of goods' photo in the Internet (optional)
                        'country_of_origin' => '',            // Country of origin (optional)

                        'invoice' => 'XXXXXXXXXXXX',            // Invoice number (required)
                        'descr_rus' => 'Портативный аудио плеер', // Short description in Russian (optional)
                        'descr_alt' => 'Funny audio',          // Short description in supplier's or English language
(required)
                        'descr_alt_eng' => true,              // If description is in English, pass true
                        'noti_num' => 'RU0000005470',            // Notification's number (optional)
                        'noti_date' => '01.01.2017',           // Notification's expiration date (optional)
                        'noti_code' => '01092',              // Notification's code (optional)
                     ),
                  ),
               ),

            ),
         );

         $ch = curl_init();
```

```
        curl_setopt($ch, CURLOPT_URL, 'http://api.boxberry.de/json.php');
        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($SDATA));
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); $data =
        json_decode(curl_exec($ch), 1);

        if(isset($data['err'])){

          // if an error occurred and the response has not been received
          print ($data['err']);

        } else if(isset($data['result'])){

          /*
            // everything is fine, the response is received, now in the array $data
            print_r($data);
            Array
            (
              [result] => Array
                (
                      [track] =>
                      XXXXXXXXX [label] =>
http://api.boxberry.de/?act=build&track=XXXXXXXXX&token=YourAPItoken
                      [box] => 1XXXXXXXXXX
                )

            )
          */
        }

        ?>
```

SOAP

Example for PHP 5.0 or later version

```
<?php
        $SDATA = array(
                  'method' => "ParcelCreateForeign",   // Name of method (required)
        'token' => 'YourAPItoken',          // Your API token (required)

        'u_name' => 'Иван',                 // Name (required)
        'u_surname' => 'Иванов',             // Last name (required)
        'u_middlename' => 'Иванович',         // Middle name (optional)
        'u_email' => 'air@email.com',        // EMAIL (required)
        'u_phone' => '+7 XXX XXX XX XX',      // Receiver’s phone number (required)
        'u_country_code' => 643,             // Code of issuing state. By default 643 - Russia. (required)
        'u_pasport' => 'XXXX XXXXXX',         // Number of identification document (required)
        'u_passportIssued'=>'2017-01-01',     // Identification document’s date of issue (required)
        'u_passportIssuedBy'=>'',            // Identification document issued by (required)
        'u_destination_country_code' => '',   // The code of the destination country (required)
        'u_city' => "",                     // Receiver’s city (required)
        'u_take' => 'kd',                    // kd if courier delivery or Pick-up point code (required)
        'u_post_code' => 'XXXXXX',            // XXXXXX - Postal code (optional, if delivery type – pick-up
point)
```

```
        'u_street' => '5-я Курская, д.8 кв.16', // Address of the receiver (optional, if delivery type – pick-up point)
        'Order' => '123456789',          // Order number in the Internet shop (optional, Sender tracking is assigned if empty)
        'sender_tracking' => '',         // Tracking number in the Internet shop (optional, Boxberry tracking is assigned if empty)
        'barcode_pallets' => 'XXXXXXXXXXX',    // Field for transmitting information on barcode pallets (optional)
        'barcode_bigbox' => 'XXXXXXXXXXX',     // Field for transmitting information on barcode boxes (big with small packets inside) (optional)
        'extraPassData' => 'XXXXXXXXXXX',      // Additional data if necessary (e.g. INN, subdivision code) (optional)
        'box' => array(                  // Array of boxes with goods
          array(                      // Array of one box, not more than 5 boxes in one parcel
            'x' => '10',              // Dimensions of the box, cm
            'y' => '20',
            'z' => '40',
            'weight_bruto' => '515',      // Gross weight of the box in grams (required)
            'add_tracking_code' => '',     // Additional tracking code to the sender's personal needs (optional)
            'add_barcode128' => 'XXXXXXXXX',// Barcode in the Internet shop for printing on the label (Derivation on the label – EAN, optional)
                  'seller' => array(          // For Type IM = Mediator
              'exporter_number' => 'exporter_number',       // Number for export clearance (required)
              'legal_name' => 'legal_name',               // Legal name of the seller (required)
              'web_address' => 'web_address',              // Seller’s common name (optional)
              'address' => 'Address',                  // Seller’s address (required)
              'post_code' => 'post_code',                 // Seller’s Zip-code (required)
              'city' => 'city',                      // Seller’s city (required)
              'country_code' => 643,                    // Seller’s country code (required)
            ),

            'items' => array(                         // Array of goods in the box
              array(                           // Array of one unit of goods, can be any amount
                'articul' => '1232212',               // Article in the shop (required)
                'Manufacturer' => 'Hitachi',             // Manufacturer (optional, No name if empty)
                'model' => 'DLS-118',                // Item name (required)
                'quantity' => '1',                  // Quantity (required)
                'bruto' => '500',                   // Gross weight (gr) (required)
                'item_price' => '200',               // Price in supplier’s currency (required)
                'currency_price' => 'EUR',             // Supplier’s currency (EUR GBP USD RUB). If incorrect or field is empty, than EUR (required)
                'web_address' => ''                 // Link to the manufacturer
                'link_web' => '',                  // Link to the item of goods in the internet (required)
                'link_foto' => '',                 // Link to the item of goods’ photo in the Internet (optional)
                'country_of_origin' => '',            // Country of origin (optional)

                'invoice' => 'XXXXXXXXXXXX',            // Invoice number (required)
                'descr_rus' => 'Портативный аудио плеер', // Short description in Russian (optional)
                'descr_alt' => 'Funny audio',            // Short description in supplier’s or English language (required)
                'descr_alt_eng' => true,               // If description is in English, pass true
```

```
                    'noti_num' => 'RU0000005470',          // Notification’s number (optional)
                    'noti_date' => '01.01.2017',         // Notification’s expiration date (optional)
                    'noti_code' => '01092', ),         // Notification’s code (optional)

                        ),
                    ),

                ),
            );
            ?>
```

```
<?php
        $client = null;

        try {
           ini_set("soap.wsdl_cache_enabled", "0");

           $client = new SoapClient("http://api.boxberry.de/__soap/1c_lc.php?wsdl");

           $data = json_decode(json_encode($client->ParcelCreateForeign($SDATA)), true);

        } catch (SoapFault $ex) {

           // print $ex->getMessage();

        }

        if (isset($data['result'])) {

        /*
        print_r($data);
           Array
              (
                 [result] => Array
                    (
                       [track] => XXXXXXXXX
                       [label] => http://api.boxberry.de/?act=build&track=XXXXXXXXX&token=YourAPItoken
                       [box] => 1XXXXXXXXXX
                    )
              )
        */
        }
        ?>
```

ParselCheck

xxxxxx- tracking number
ParselCheck method allows to get the link to the file for printing labels, the list of barcodes in parcel separated with commas (,), the list of barcodes for boxes in parcel separated with comma (,).
Necessary to have parameter (tracking code).

Important!

Service works only with parcels created at api.boxberry.de

Datatable 13. Method ParselCheck

JSON	Url: http://api.boxberry.de/json.php?token=YourAPItoken&method=ParselCheck&ImId=XXXXXX

Request POST method

```
<?php
        $url='http://api.boxberry.de/json.php?token=YourAPItoken&method=ParselCheck&ImId=XXXXXX';
        // XXXXXX - Tracking number
        $handle = fopen($url, "rb");
        $contents = stream_get_contents($handle);
        fclose($handle);
        $data=json_decode($contents,true);
        if(count($data)<=0 or !empty($data['err'])){
          // if an error occurred and the response has not been received
          echo $data['err'];
        }else{
          // everything is fine, the response is received, now in the array $data
          /*
            $data=array(
                'label'=>'Link to the PDF file with stickers'
                'box'=>'Barcode list of boxes in the parcel',
                'box_upload'=>'Barcodes list of uploaded boxes in parcel'
            );
          */
        }
        ?>
```

SOAP

Example for PHP 5.0 or later version

```
<?php

          $url = 'http://api.boxberry.de/__soap/1c_lc.php?wsdl';

          $query = [];
          $query['token'] = 'YourAPItoken';
          $query['ImId'] = 'XXXXXX';
          // XXXXXX - Tracking number.

          try{
```

Contents

```
    ini_set("soap.wsdl_cache_enabled", "0");
    $client = new SoapClient($url);
    $data = json_decode(json_encode($client->ParselCheck($query)), true);
} catch (SoapFault $ex){
    // if an error occurred and the response has not been received.
    $data = [];
    //print $ex->getMessage();
}

if(!empty($data))
{
    // everything is fine, the response is received, now in the array $data:
    /*
    $data=array(
        'label'=>'Link to the PDF file with stickers', 'box'=>'Barcode
        list of boxes in the parcel', 'box_upload'=>'Barcodes list of
        uploaded boxes in parcel'
    );
    */
}
?>
```

ParselSend

ParselSend method uploads parcels' data to the Boxberry system. It is necessary to send encoded JSON array by POST. Description of the array can be found below. Method returns JSON encoded array with identification number.

Datatable 14. Method ParselSend

JSON	Url: http://api.boxberry.de/json.php?token=YourAPItoken&method=ParcelCreate

Request POST method

```
<?php
        $SDATA=array();
        $SDATA['control']='on';     // Control of box's conformance to parcel, is automatically switch ON,
press off to switch OFF
        $SDATA['pallet']=array(     // Array of pallets
             array( // Array of one pallet, pallets might be few
                  'barcode'=>'',                  // Barcode of pallet. 0 or empty if absence
                  'box'=>'B0000000000,B0000000000', // Barcodes of boxes in pallets, cross commas(,)
```

```
            ),
        );
        ?>
```

```
<?php
        // Suppose, you've already create array $SDATA by the description above
        // Sending the array to boxberry server by CURL
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, 'http://api.boxberry.de/json.php');
        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_POSTFIELDS, array(
            'token'=>'YourAPItoken',
            'method'=>'ParselSend',
            'sdata'=>json_encode($SDATA)
        ));
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        $data = json_decode(curl_exec($ch),1);
        if(!empty($data['err']) or count($data) == 0 )
        {
            // if an error occurred and the response has not been
            received echo $data['err'];
        }
        else
        {
        // everything is fine, the response is received, now in the array $data
        /*
          $data=array(
              'id'=>'XXXXXXXX' // Upload ID in boxberry system
          );
        */
        }
        ?>
```

ParselReadiness

ParselReadiness method informs if all the necessary parcel data is collected or not.

Datatable 15. Method ParselReadiness

Request POST method

```
<?php
        header("Content-type: text/plain; charset=utf-8");
        $gdata = array();
        $gdata['token'] = 'YourAPItoken'; // token
        $gdata['method']='ParselReadiness'; // method name
        // Send one or all of the arguments.
        $gdata['track'] = ''; // Tracking number
        $gdata['dop_code'] = ''; // Reference number for personal needs of sender
        $gdata['dop_barcode128']=''; // Barcode data for the personal needs of the sender (Output on the
label - EAN 128)

        // generation of query string
        $query = http_build_query($gdata);

        // Accessing the service JSON
        $data = json_decode(file_get_contents('http://api.boxberry.de/json.php?'. $query), true);

        // if an error occurred and the response has not been received
        if(isset($data['err'])){
          print $data['err'];
        } else
        // everything is fine, the response is received, now in the array $data
        if(isset($data['pre'])){
          // The method ensures that the order is ready for shipment or not.
          // 0 - ready, 1 - no
          print_r($data);
        }
        ?>
```

SOAP

Example for PHP 5.0 or later version

```
<?php

        $query = [];
        $query['token'] = 'YourAPItoken'; // Your API token,
        // Send one or all of the arguments.
        $query['track'] = 'XXXXXXXX'; // XXXXXX - Tracking number,
        $query['dop_code'] = ''; // Reference number for personal needs of sender,
        $query['dop_barcode128']=''; // Barcode data for the personal needs of the sender (Output on
the label - EAN 128).

        $url = "http://api.boxberry.de/__soap/1c_lc.php?wsdl";

        try{
```

```
    $client = new SoapClient($url);
    $data = json_decode(json_encode($client->ParselReadiness($query)),true);
} catch (SoapFault $ex){
    // if an error occurred and the response has not been received.
    $data = [];
    print $ex->getMessage();
}
if(!empty($data)) {
    /*
    // everything is fine, the response is received, now in the array
    $data: print_r($data);
    Array
        (
            [result] => Array
                (
                    [pre] => 0
                )
        )
    */
}
?>
```

ParselDel

ParselDel method deletes parcels’ data created via API (before request to ParselSend).

Datatable 16. Method ParselDel

JSON	Url: http://api.boxberry.de/json.php?token=YourAPItoken&method=ParcelDel&parcelId=XXXXXX

Request POST method	<?php
	$url='http://api.boxberry.de/json.php?token=YourAPItoken&method=ParcelDel&parcelId=XXXXXX';
	// XXXXXX - Tracking number:
	$handle = fopen($url, "rb");
	$contents = stream_get_contents($handle);
	fclose($handle);
	$data=json_decode($contents,true);
	if(count($data)<=0 or $data['err'])
	{
	// if an error occurred and the response has not been received:

```
          echo $data['err'];
      }
      else
      {
          // everything is fine, the response is received, now in the array $data:
          /*
          $data=array( 'message'=>'...'


          );
          */
      }
      ?>
```

SOAP

Example for PHP 5.0 or later version

```
<?php
      ini_set("soap.wsdl_cache_enabled", "0"); // disable caching of WSDL.
      try {
         $clientS = new SoapClient('http://api.boxberry.de/__soap/1c_lc.php?wsdl');
      } catch (SoapFault $exception){
         $errS=$exception;
      }

      $SDATA=array();
      $SDATA['token']='YourAPItoken';
      $SDATA['parcelId']='XXXXXX';

      try {
         $get_data=$clientS->ParcelDel($SDATA);
      } catch (SoapFault $exception){
         $errS=$exception;
      }
      if(!empty($errS)){
         echo $errS;
      } // if an error occurred and the response has not been received.
      else
      {
         // in response:
         /*
         stdClass Object
         (
             [message] => ...
         )
         */
      }
      ?>
```

ParcelSendForeign

ParselSendForeign method uploads parcels’ data to Boxberry system. It's necessary to send encoded JSON array by POST. Description of the array can be found below. Method will return JSON encoded array with identification number.

Datatable 17. Method ParselSendForeign

SOAP

Example for PHP 5.0 or later version

```
<?php

      $query=[];
      $query['token']='YourAPItoken';
      $query['control']='on';    // Control of box's conformance to parcel, is automatically switch ON,
press off to switch OFF,
      $query['pallet']=array(    // Array of pallets,
         array( // Array of one pallet, pallets might be few,
            'barcode'=>'',              // Barcode of pallet. 0 or empty if absence,
            'box'=>'10000000000,10000000000', // Barcodes of boxes in pallets, cross commas(,).
         ),
      );

      $url = "http://api.boxberry.de/__soap/1c_lc.php?wsdl";

      try{
         ini_set("soap.wsdl_cache_enabled", "0");
         $client = new SoapClient($url);
         $data = json_decode(json_encode($client->ParcelSendForeign($query)),true);
      } catch(SoapFault $ex){
         // if an error occurred and the response has not been received.
         $data = [];
         print $ex->getMessage();
      }

      if(!empty($data['result'])){

         /*
         // everything is fine, the response is received, now in the array
         $data: print_r($data);

         Array
            (
               [result] => Array
                  (
                     [id] => U-XXXXXXX
                  )
            )
         */

      }
```

	?>

ParcelUpdateForeign

ParcelUpdateForeign method updates previously created parcel’s data in Boxberry system. It is necessary to transfer data to array by POST method. Method will return JSON encoded array with Identification number (tracking number), link to the labels and the list of boxes barcodes in the parcel separated with commas(,).

Method contains same structure as ParcelsCreateForeign method plus extra parameter ‘track_id’ in the main data array

List of changes

Datatable 18. List of changes

Date	Description of change
February 2016	ParcelCreateForeign method added. ListCountry method added. ListCities method updated. ListPoints method updated. PointsDescription method updated.
June 2016	Description’s visual renewal.
March 2017	ParcelDel method added. ParcelSend method added. ParcelCreateForeign method updated.
September 2017	ParcelCreateForeign method updated. Description’s visual renewal.
June 2018	GetStatusForeign method added ListStatuses method replaced with GetStatusesForeign

Contents