



# ORLEN Paczka API Documentation

## 1 TABLE OF CONTENTS

1	Table of contents.....	1
2	Change History .....	6
3	Introduction.....	9
3.1	<b>WSDL ADDRESSES API ORLEN PACZKA .....</b>	<b>9</b>
3.2	<b>ORLEN PACZKA ADVISING SYSTEMS .....</b>	<b>9</b>
3.3	<b>FORM OF AUTHORIZATION .....</b>	<b>9</b>
3.4	<b>DOWNLOADING AND VISUALIZING DATA ON PUDO/APM ORLEN PACZKA LOCATIONS.....</b>	<b>9</b>
3.5	<b>TRANSFER OF ADVISORY DATA AND GENERATION OF LABELS .....</b>	<b>10</b>
3.5.1	Additional data for reporting purposes .....	10
3.5.2	Declaration of identifiers of ORLEN Paczka pickup points.....	11
3.5.3	ORLEN Paczka provides three scenarios for advisory data transfer and label generation: .....	11
3.6	<b>EXPIRATION DATE OF LABELS .....</b>	<b>13</b>
3.7	<b>GENERATION OF PROTOCOLS FOR HANDING OVER SHIPMENTS TO THE COURIER.....</b>	<b>13</b>
3.8	<b>ORDERING A COURIER AND CHECKING THE POSSIBILITY OF A DRIVE TO THE SPECIFIED ADDRESS .....</b>	<b>13</b>
3.9	<b>DOWNLOAD TRACKING STATUSES .....</b>	<b>14</b>
3.9.1	TRACKING SHIPMENT IN TRACK&TRACE ORLEN PACZKA SERVICE.....	14
3.10	<b>CONSUMER RETURNS.....</b>	<b>14</b>
3.11	<b>PRIMARY PACKAGE NUMBER .....</b>	<b>15</b>
3.12	<b>COD PARCEL .....</b>	<b>15</b>
3.13	<b>PACKAGE DIMENSIONS .....</b>	<b>15</b>
3.14	<b>SHIPMENT INSURANCE SERVICE .....</b>	<b>16</b>
4	Methods.....	16
4.1	<b>CreateCustomer .....</b>	<b>17</b>
4.1.1	Request.....	17
4.1.2	Response .....	18
4.2	<b>GenerateBusinessPack .....</b>	<b>18</b>
4.2.1	Request.....	18

4.2.2	Response .....	20
<b>4.3</b>	<b>GenerateBusinessPackTwo .....</b>	<b>20</b>
4.3.1	Request.....	20
4.3.2	Response .....	22
<b>4.4</b>	<b>GenerateLabelBusinessPack .....</b>	<b>23</b>
4.4.1	Request.....	23
4.4.2	Response .....	25
<b>4.5</b>	<b>GenerateLabelBusinessPackList.....</b>	<b>26</b>
4.5.1	Request.....	26
4.5.2	Response .....	28
<b>4.6</b>	<b>GenerateLabelBusinessPackListTwo.....</b>	<b>29</b>
4.6.1	Request.....	29
4.6.2	Response .....	31
<b>4.7</b>	<b>GenerateLabelBusinessPackTwo .....</b>	<b>32</b>
4.7.1	Request.....	32
4.7.2	Response .....	34
<b>4.8</b>	<b>GenerateNumberWaybills .....</b>	<b>35</b>
4.8.1	Request.....	35
4.8.2	Response .....	35
<b>4.9</b>	<b>GenerateProtocol .....</b>	<b>36</b>
4.9.1	Request.....	36
4.9.2	Response .....	36
<b>4.10</b>	<b>GenerateShippingCode.....</b>	<b>37</b>
4.10.1	Request.....	37
4.10.2	Response .....	39
<b>4.11</b>	<b>GetGeneratedParcels .....</b>	<b>40</b>
4.11.1	Request.....	40
4.11.2	Response .....	40
<b>4.12</b>	<b>GiveCurrentPack .....</b>	<b>41</b>
4.12.1	Request.....	41
4.12.2	Response .....	41
<b>4.13</b>	<b>GiveMasterPack.....</b>	<b>41</b>
4.13.1	Request.....	42
4.13.2	Response .....	42

<b>4.14</b>	<b>GiveMeAllPSDLocation</b> .....	42
4.14.1	Request.....	42
4.14.2	Response .....	42
<b>4.15</b>	<b>GiveMeAllLocationWithAllData</b> .....	43
4.15.1	Request.....	43
4.15.2	Response .....	43
<b>4.16</b>	<b>GiveMeAllLocationWithAllDataWithZipCode</b> .....	46
4.16.1	Request.....	46
4.16.2	Response .....	46
<b>4.17</b>	<b>GiveMeAllAPMLocation</b> .....	48
4.17.1	Request.....	48
4.17.2	Response .....	49
<b>4.18</b>	<b>GiveMeAllRUCHLocation</b> .....	50
4.18.1	Request.....	50
4.18.2	Response .....	50
<b>4.19</b>	<b>GiveMeAllRUCHZipcode</b> .....	51
4.19.1	Request.....	51
4.19.2	Response .....	51
<b>4.20</b>	<b>GiveMeAllRUCHWithFilled</b> .....	52
4.20.1	Request.....	52
4.20.2	Response .....	52
<b>4.21</b>	<b>GiveMeAllRUCHWithFilledD1</b> .....	55
4.21.1	Request.....	55
4.21.2	Response .....	55
<b>4.22</b>	<b>GiveMePSDZipcode</b> .....	57
4.22.1	Request.....	57
4.22.2	Response .....	57
<b>4.23</b>	<b>GiveMePackStatus</b> .....	58
4.23.1	Request.....	59
4.23.2	Response .....	59
<b>4.24</b>	<b>GiveMePackStatusList</b> .....	59
4.24.1	Request.....	60
4.24.2	Response .....	60
<b>4.25</b>	<b>GiveMePackStatusFullHistory</b> .....	60

4.25.1	Request.....	61
4.25.2	Response .....	61
<b>4.26</b>	<b>GiveMePackStatusFullHistoryList .....</b>	<b>62</b>
4.26.1	Request.....	62
4.26.2	Response .....	63
<b>4.27</b>	<b>GivePartnerStatus.....</b>	<b>64</b>
4.27.1	Request.....	64
4.27.2	Response .....	64
<b>4.28</b>	<b>GivePaymentType.....</b>	<b>65</b>
4.28.1	Request.....	65
4.28.2	Response .....	66
<b>4.29</b>	<b>GiveShippingCodeStatus .....</b>	<b>66</b>
4.29.1	Request.....	66
4.29.2	Response .....	66
<b>4.30</b>	<b>LabelPrintDuplicate .....</b>	<b>67</b>
4.30.1	Request.....	67
4.30.2	Response .....	67
<b>4.31</b>	<b>LabelPrintDuplicateTwo .....</b>	<b>68</b>
4.31.1	Request.....	68
4.31.2	Response .....	68
<b>4.32</b>	<b>LabelPrintDuplicateList .....</b>	<b>69</b>
4.32.1	Request.....	69
4.32.2	Response .....	69
<b>4.33</b>	<b>LabelPrintDuplicateListTwo .....</b>	<b>70</b>
4.33.1	Request.....	70
4.33.2	Response .....	70
<b>4.34</b>	<b>PutCustomerPack .....</b>	<b>71</b>
4.34.1	Request.....	71
4.34.2	Response .....	73
<b>4.35</b>	<b>PutCustomerPackCanceled .....</b>	<b>74</b>
4.35.1	Request.....	74
4.35.2	Response .....	74
<b>4.36</b>	<b>Ping.....</b>	<b>74</b>
4.36.1	Request.....	74

4.36.2	Response .....	74
<b>4.37</b>	<b>CallPickup</b> .....	<b>75</b>
4.37.1	Request.....	75
4.37.2	Response .....	75
<b>4.38</b>	<b>CallPickupNew</b> .....	<b>76</b>
4.38.1	Request.....	76
4.38.2	Response .....	77
<b>4.39</b>	<b>GetAvailablePickups</b> .....	<b>77</b>
4.39.1	Request.....	78
4.39.2	Response .....	78
<b>4.40</b>	<b>GiveMeShippingCodeStatusFullHistory</b> .....	<b>79</b>
4.40.1	Request.....	79
4.40.2	Response .....	79
<b>4.41</b>	<b>PutCustomerShippingCodeCanceled</b> .....	<b>80</b>
4.41.1	Request.....	80
4.41.2	Response .....	80
<b>4.42</b>	<b>GenerateCustomerReturn</b> .....	<b>81</b>
4.42.1	Request.....	81
4.42.2	Response .....	82
<b>4.43</b>	<b>GenerateCustomerReturnShippingCode</b> .....	<b>82</b>
4.43.1	Request.....	83
4.43.2	Response .....	83
<b>4.44</b>	<b>GenerateStandardCustomerReturn</b> .....	<b>84</b>
4.44.1	Request.....	84
4.44.2	Response .....	85
<b>4.45</b>	<b>GenerateStandardCustomerReturnShippingCode</b> .....	<b>86</b>
4.45.1	Request.....	86
4.45.2	Response .....	87
<b>4.46</b>	<b>ORLENPaczkaWeekendOpenPUDOPoints</b> .....	<b>88</b>
4.46.1	Request.....	88
4.46.2	Response .....	88
<b>4.47</b>	<b>GenerateBrokerReturn2Home</b> .....	<b>90</b>
4.47.1	Request.....	90
4.47.2	Response .....	91

<b>4.48</b>	<b>GenerateBrokerShippingCodeReturn2Home</b> .....	92
4.48.1	Request.....	92
4.48.2	Response .....	93
5	Error messages .....	94
6	Warning messages .....	96
7	Parcel statuses.....	97

## 2 CHANGE HISTORY

Date	Version	Changes
06.2026	v_1_26_003	Updated information about the ability to generate a label in ZPL 300 DPI format using the GenerateLabelBusinessPackListTwo method
05.2026	v_1_26_002	Adding a new methods GenerateBrokerReturn2Home, GenerateBrokerShippingCodeReturn2Home
03.2026	v_1_26_001	Adding a new method ORLENPaczkaWeekendOpenPUDOPoints
11.2025	v_1_25_008	Minor corrections.
10.2025	v_1_25_007	Revision of sections 3.8 and 3.14 (Insurance Service).
09.2025	v_1_25_006	Update descriptions of consumer returns methods.
03.2025	v_1_25_005	Update descriptions of methods GiveMePackStatus, GiveMePackStatusList, GiveMePackStatusFullHistory, GiveMePackStatusFullHistoryList.
03.2025	v_1_25_004	Added statuses used during temporary storage (241, 681, 691, 696, 708).
02.2025	v_1_25_003	Added a description of the <BuildingNumber> tag in the methods that return ORLEN Paczka pickup point data. This tag contains the building number and the unit number (if applicable) of the pickup point.
01.2025	v_1_25_002	Adding specification Track&Trace link ORLEN Paczka.
01.2025	v_1_25_001	Adding dimension specifications in ORLEN Paczka service (S, M, L).

11.2024	v_1_24_002	Adding information on deactivation of the COD (Cash On Delivery) service.
10.2024	v_1_24_001	Update the 'Introduction' section.
		Updating descriptions of methods and their parameters.
		Adding a new method GiveMeAllLocationWithAllDataWithZipcode that returns the list of points with an expanded description and zipcode.
		Adding a new method GenerateCustomerReturn.
		Adding a new method GenerateCustomerReturnShippingCode.
		Adding a new method GenerateStandardCustomerReturn.
		Adding a new method GenerateStandardCustomerReturnShippingCode.
31.08.2023	v_1_23_003	Adding a new method GiveMeAllLocationWithAllData that returns the list of points with an expanded description.
01.07.2023	v_1_23_002	Adding a new method GetAvailablePickups that returns the list of possible pickup dates.
12.04.2023	v_1_23_001	Adding new method GiveMeShippingCodeStatusFullHistory and PutCustomerShippingCodeCanceled.
14.07.2022	v_1_22_002	Adding a new pick-up point type <PointType> PPK that accepts sizes S, M i L. You cannot send a parcel from a PPK pick-up point. Information about handling only pickups is returned in a location description parameter <Location>.
25.05.2022	v_1_22_001	Adding new method GiveMeAllRUCHWithFilledD1 that returns a list of pick-up points that meet the criteria of delivery within 24 hours (D+1).
		Adding new method Ping that checks the status of connection with the API webservice.
		Update of the „Introduction” section with a description of the procedure for generating return labels.
13.10.2021	v_1_16_007	Adding new method GiveMeAllAPMLocation that returns a list of all Automated Parcel Machines available in ORLEN Paczka service.

		Adding new types of pick-up points <PointType>: APM (Automated Parcel Machines) and PKN (ORLEN Petrol Stations) to the two methods GiveMeAllRUCHLocation and GiveMeAllRUCHWithFilled.
		Adding new package sizes into the parameter <BoxSize> that are suitable for a new pick-up point types APM and PKN.

## 3 INTRODUCTION

### 3.1 WSDL ADDRESSES API ORLEN PACZKA

Production environment:

<https://api.ornlenpaczka.pl/WebServicePwRProd/WebServicePwR.asmx?wsdl>

Test environment:

<https://api-test.ornlenpaczka.pl/WebServicePwR/WebServicePwR.asmx?WSDL>

ORLEN Paczka API is a SOAP-based WebService supporting SOAP version 1.1 and SOAP version 1.2. We recommend using SOAP version 1.2.

### 3.2 ORLEN PACZKA ADVISING SYSTEMS

Shipments sent in ORLEN PaczkaAPI will not be visible in WebTrucker system (<https://manager.ornlenpaczka.pl/Default.aspx>) and vice versa.

Shipments sent through ORLEN Paczka API and through WebTrucker system have different numberings and are broadcast through two different IT systems.

Shipments generated by ORLEN Paczka API have a number starting with 21. Shipments generated via WebTrucker start their number with 26.

The history of shipments sent through the API should be stored directly on integration side using the ORLEN Parcels API. WebTrucker only stores the history of shipments sent through it.

### 3.3 FORM OF AUTHORIZATION

Methods that require authorization do so by specifying in the sent request in the <PartnerID> field the client's login and in the <PartnerKey> field the client's password.

Access data to the production and test environment (login and password) are provided by the sales representative.

Access data for the test environment is different from access data for the production environment.

### 3.4 DOWNLOADING AND VISUALIZING DATA ON PUDO/APM ORLEN PACZKA LOCATIONS

To provide a choice of pick-up points in ORLEN Paczka network, we provide three options:

#### A. ORLEN Paczka map widget

A link to the documentation is available at:

<https://www.ornlenpaczka.pl/integracja-z-ornlen-paczka> - „Widget mapy ORLEN Paczka” option.

An individual access token is generated for the ORLEN Paczka map widget.

To receive the token necessary for the widget implementation, please contact your sales representative with the following details:

- domain name of the store where the map will be embedded
- customer number (PWR...)
- if you intend to test the widget in a local environment (localhost), please inform your sales

representative about this in your request. The token required for testing in a local environment is generated separately, is temporary, and expires 3 months after being generated

## **B. Older ORLEN Parcel widget based on OpenStreetMap map**

Link to the documentation is available at: <https://www.ornlenpaczka.pl/integracja-z-ornlen-paczka> - „Widget mapy OSM” option.

The documentation includes instructions for installing the OSM widget, along with examples of possible implementations in PHP.

The packet with sample implementations does not contain a plugin but refers to all the links of the production plugin in the instructions.

## **C. Downloading data on PUDO points via ORLEN Paczka API.**

We recommend using the GiveMeAllLocationWithAllDataWithZipcode method to retrieve a list of all types with expanded point descriptions and zip codes

Updating of PUDO points databases takes place once a day in the morning. In order to maintain up-to-date data, we recommend downloading the list of points once a day after 6 o'clock.

Other methods for retrieving the list of points are placed in this documentation in sections 4.14 - 4.22.

Types of points available in the ORLEN Paczka service:

APM - Parcel Machines

PKN - ORLEN petrol stations

PSD - Retail Point of Sale (RUCH kiosks) – points supporting COD service (as of 01.01.2025, PSD points have been closed)

PPP - Out-of-press point

PPK - PointPack network points – points that only handle parcel pick-ups

### **Change in how address data is returned for methods retrieving PUDO/APM point data.**

As of January 1, 2025, the street name where the PUDO/APM point is located will be returned in the <StreetName> tag, while the street number and building/unit number will be returned in the <BuildingNumber> tag.

## **3.5 TRANSFER OF ADVISORY DATA AND GENERATION OF LABELS**

### **3.5.1 Additional data for reporting purposes**

“TransferDescription” parameter is used to enter any data such as shipper's reference data (for example: order number, buyer ID, package contents). Data transferred in the “TransferDescription” field is not printed on the label.

„SenderOrders” parameter is used to enter any data such as shipper's reference data (for example: order number, buyer ID, package contents). Data transferred in the “SenderOrders” field is printed on the label.

Parameters “ExternalSenderPackageNumber”, “ExternalPackageNumber” available in GenerateStandardCustomerReturn, GenerateStandardCustomerReturnShippingCode methods are used to enter any data such as shipper's reference data (for example: order number, buyer ID, package contents). Data passed in the fields “ExternalSenderPackageNumber”, “ExternalPackageNumber” are not printed on the label.

At the customer's request, after consultation with an ORLEN Paczka sales representative, a shipment report will be generated containing the data entered in these parameters.

### 3.5.2 Declaration of identifiers of ORLEN Paczka pickup points

In the advising methods, pass in the <DestinationCode> field the value returned in the <DestinationCode> field of the GiveMeAllLocationWithAllDataWithZipcode method.

At the time of transmission of advisory data (section 3.5.3), the full or universal identifier of the reception point must be declared.

identifier of pickup point is returned in the DestinationCode parameter by the methods described in Section 3.3 - point C.

Example: <DestinationCode>WS-116263-C3-01</DestinationCode>

Universal identifier is created by replacing the prefix with the value XX and replacing the suffix with 00-00.

Example: XX-116263-00-00.

Passing a universal identifier will correctly generate the shipment number and label in spite of returning a warning message „Saved but changed DestinationCode” (section 6).

### 3.5.3 ORLEN Paczka provides three scenarios for advisory data transfer and label generation:

**A: We recommend using the GenerateLabelBusinessPackListTwo method for reserving the package number and generating the label.** This method allows advising a single parcel or a list of ORLEN Paczka shipments, returns label in PDF, PDF10, EPL, ZPL or ZPL 300 DPI format in response.

Other methods for transmitting advisory data and generating labels are placed in this documentation in sections 4.4 - 4.7.

**B: Booking a waybill number using one method, and then generating a shipping label using an additional method.**

Step 1.

Calling the GenerateBusinessPack method used to transfer advisory data to ORLEN Paczka systems. This method does not return label in the response. In response, the system returns the waybill number for which the label should be generated.

Step 2.

To generate a label for a shipment that was messaged in step one, it is recommended to use the LabelPrintDuplicateListTwo method (section 4.33).

Other methods for retrieving duplicate shipping labels are placed in this documentation in sections 4.30 - 4.33.

### **C: Transfer of advisory data and generation of a label for the reserved pool of shipment numbers.**

The use of this scenario is possible only with the approval of the sales representative of ORLEN Paczka.

The scenario consists of three consecutive steps:

Step 1. Reserving a pool of parcel numbers.

Use the GenerateNumberWaybills method (section 4.8) to reserve a pool of numbers. Such reserved shipment numbers have no expiration date. At this stage, the parcel is not notified in ORLEN Paczka systems, which means that it does not receive tracking status and is not visible on the parcel tracking page ([www.ornlenpaczka.pl/sledz-paczke](http://www.ornlenpaczka.pl/sledz-paczke)).

Step 2. Transmission of advisory data.

to transfer the advisory data to the shipment number reserved in step one, use the PutCustomerPack method (section 4.34).

**Warning:** If the parcel is not forwarded to the ORLEN Paczka network within 14 calendar days from the time of avigation, it will be automatically canceled.

Step 3. Downloading a shipping label.

The process of retrieving the shipment label for the messaged package in step two via the recommended LabelPrintDuplicateListTwo method (section 4.33) or methods described in sections 4.30 - 4.33.

The <PackPrice> tag included in the response of the advising methods is directly related to the declared package size. Its value, does not represent the final amount for the courier service. The final price is determined individually with the sales supervisor when signing the contract. Below are the standard values of the <PackPrice> tag for different sizes, valid for all ORLEN Paczka customers:

REQUEST: <BoxSize>S</BoxSize>

RESPONSE: <PackPrice>999</PackPrice>

REQUEST: <BoxSize>M</BoxSize>

RESPONSE: <PackPrice>1099</PackPrice>

REQUEST: <BoxSize>L</BoxSize>

RESPONSE: <PackPrice>1299</PackPrice>

The <PackPaid> tag included in the advising methods response specifies the type of customer account:

<PackPaid>>true</PackPaid> - means that the business contract has been signed and customer has POSTPAID status in ORLEN Paczka systems.

<PackPaid>>false</PackPaid> - means that the business contract has not been signed and has PREPAID status in ORLEN Paczka systems.

### 3.6 EXPIRATION DATE OF LABELS

Labels generated through the ORLEN Paczka API are valid for 14 days from the date of generation of the shipping label.

If the parcel is not received in the ORLEN Paczka network within 14 days from the date of advice, it is automatically canceled and receives the status "201 - canceled advice".

### 3.7 GENERATION OF PROTOCOLS FOR HANDING OVER SHIPMENTS TO THE COURIER

Use the GenerateProtocol method (section 4.9) to generate the handover protocol.

Protocol file is returned in PDF format for the indicated ORLEN Paczka numbers. Method returns incorrect packages (all with a different status than notified) and correct packages with the protocol number

### 3.8 ORDERING A COURIER AND CHECKING THE POSSIBILITY OF A DRIVE TO THE SPECIFIED ADDRESS

To call a courier to pick up packages, follow the steps below.

Step 1. Check the availability of courier time slots for the shipper's zip code.

To verify available courier time slots, use the GetAvailablePickups method (section 4.39).

Method returns a list of the nearest available driveway dates and times for the indicated zip code.

In response of method, the parameter "MinimumInterval" is returned, which specifies the minimum time to call the courier according to the end of the hourly interval. "MinimumInterval" value is returned as the number of minutes.

Example:

hourly range below indicates courier availability from 08:00 to 16:00

```
<AvailablePickupDay>
  <Date>2024-08-29</Date>
  <MinReadyDate>2024-08-29T08:00:00+02:00</MinReadyDate>
  <MaxPickupDate>2024-08-29T16:00:00+02:00</MaxPickupDate>
  <MinimumInterval>120</MinimumInterval>
</AvailablePickupDay>
```

Value of the returned parameter <MinimumInterval>120</MinimumInterval> means that the courier order can be placed no later than 2:00 p.m. (according to the formula 16:00 - 2 hours = 14:00).

Step 2. Calling for the arrival of a courier to pick up shipments.

To order a courier's pick-up, we recommend using the CallPickupNew method (section 4.38). Method allows to enter a specific address along with the hours of a possible driveway taken from the response from GetAvailablePickups method.

Calling a courier is also possible with the help of the CallPickup method (section 4.37), which automatically retrieves the arrival address from the data assigned to the customer's contract in the ORLEN Paczka system defined when signing the contract.

When the GetAvailablePickups method returned the available time slot of 8:00 - 16:00 in step one, the following sample slots can be declared in the CallPickupNew method call:

Time slot 8:00 – 10:00. The order must be submitted no later than 8:00.

```
<ReadyDate>2024-08-29T08:00:00+02:00</ReadyDate>  
<PickupDate>2024-08-29T10:00:00+02:00</PickupDate>
```

Time slot 12:00 – 15:00. The order must be submitted no later than 12:00.

```
<ReadyDate>2024-08-29T12:00:00+02:00</ReadyDate>  
<PickupDate>2024-08-29T15:00:00+02:00</PickupDate>
```

Time slot 14:00 – 16:00. The order must be submitted no later than 14:00.

```
<ReadyDate>2024-08-29T14:00:00+02:00</ReadyDate>  
<PickupDate>2024-08-29T16:00:00+02:00</PickupDate>
```

### **IMPORTANT!**

The example time slots provided in the API call are not final, but merely suggested pickup times.

Examples of invalid interval declarations, not meeting the minimum interval requirement

```
<MinimumInterval>120</MinimumInterval>
```

```
<ReadyDate>2024-08-29T08:00:00+02:00</ReadyDate>  
<PickupDate>2024-08-29T09:00:00+02:00</PickupDate>
```

```
<ReadyDate>2024-08-29T15:00:00+02:00</ReadyDate>  
<PickupDate>2024-08-29T16:00:00+02:00</PickupDate>
```

## **3.9 DOWNLOAD TRACKING STATUSES**

Tracking statuses of shipments can be retrieved through methods described in sections 4.23 – 4.26.

### **3.9.1 TRACKING SHIPMENT IN TRACK&TRACE ORLEN PACZKA SERVICE**

To track a shipment, use the following link, replacing @numerpaczki with the actual shipment number received in the notification response:

<https://www.ornpaczka.pl/sledz-paczke/?numer=@parcelnumber>

Only shipments generated on ORLEN Paczka production environment can be tracked, both API and WebTrucker (<https://manager.ornpaczka.pl/Default.aspx>).

Shipments generated on test environment will not be visible on production T&T.

## **3.10 CONSUMER RETURNS**

To use the consumer return service, please choose one of the following scenarios:

### **A. CONSUMER RETURN ASSOCIATED WITH THE ORIGINAL PACKAGE - SENDING ON THE LABEL**

“GenerateCustomerReturn” method is used to generate returns in the consumer return service. The return is associated with the original shipment. The method is described in section 4.42.

#### **B. CONSUMER RETURN ASSOCIATED WITH THE ORIGINAL PACKAGE - SENDING WITHOUT A LABEL**

“GenerateCustomerReturnShippingCode” method is used to generate returns in the consumer return service without a label. The return is associated with the original shipment. The method is described in section 4.43.

#### **C. CONSUMER RETURN NOT ASSOCIATED WITH THE ORIGINAL PACKAGE - SENDING ON THE LABEL**

“GenerateStandardCustomerReturn” method is used to generate returns in the consumer return service. The return is not associated with the original shipment. The method is described in section 4.44.

According to the business arrangement with sales representative, the shipment generated by this method is directed to the address of the company's headquarters or to the address indicated for returns.

#### **D. CONSUMER RETURN NOT ASSOCIATED WITH THE ORIGINAL PACKAGE - SENDING WITHOUT A LABEL**

“GenerateStandardCustomerReturnShippingCode” method is used to generate returns in the consumer return service without a label. The return is not associated with the original shipment. The method is described in section 4.45.

According to the business arrangement with sales representative, the shipment generated by this method is directed to the address of the company's headquarters or to the address indicated for returns.

### **3.11 PRIMARY PACKAGE NUMBER**

To get the original parcel number if you have a return parcel number, use the GiveMasterPack method. (Section 4.13).

### **3.12 COD PARCEL**

In accordance with a recent business decision, the cash on delivery (COD) payment service has been terminated as of January 2, 2025. Please kindly remove the parameters <CashOnDelivery> and <AmountCashOnDelivery> from your API requests.

### **3.13 PACKAGE DIMENSIONS**

When sending notification data to the ORLEN Paczka API, declare one of the following values in the <BoxSize> field: S, M or L. This field is used to define one of the three available package sizes. If no declaration is made, size M will be assigned by default.

Dimensional specifications of ORLEN Paczka service:

S: 8 x 38 x 60 cm, max 20 kg

M: 19 x 38 x 60 cm, max 20 kg  
L: 41 x 38 x 60 cm, max 20 kg

### 3.14 SHIPMENT INSURANCE SERVICE

Effective December 4, 2024, the parcel insurance service was discontinued in ORLEN Paczka due to a business decision.

Consequently, any attempt to generate a label with the active insurance option will result in the following error: "no permission to Insurance"

To ensure correct label generation and maintain compliance with the current ORLEN Paczka API specification:

- You should remove the <Insurance> and <PackValue> parameters from your system's calls to the ORLEN Paczka API.
- Alternatively, please set the value of the insurance declaration <Insurance> to F (False) in your requests.

In accordance with the current ORLEN Paczka Terms and Conditions,

the insurance service has been replaced by the default liability for shipments. This liability covers the value up to 5000 PLN (zł) and is automatically included – there is no need to add supplementary insurance when generating an order.

The Terms and Conditions are available at:

<https://www.ornlenpaczka.pl/wp-content/uploads/2025/07/Regulamin-swadczenia-przedsiębiorcom-uslug-pocztowych-i-przewozowych-od-25-sierpnia-2025-r..pdf>

## 4 METHODS

Call examples for all of the following methods are available in the listing at:

production environment: <https://api.ornlenpaczka.pl/WebServicePwRProd/WebServicePwR.aspx>

test environment: <https://api-test.ornlenpaczka.pl/WebServicePwR/WebServicePwR.aspx>

The methods named **GenerateLabelBusinessPackAllegro**, **GenerateLabelBusinessPackListAllegro**, **GenerateOrlenPaczkaLabel**, **GenerateOrlenPaczkaShippingCode**, **GenerateOrlenPaczkaShippingCodeReturn2Home** are tools for Allegro's internal use only. **Please do not use them as part of your built solutions.**

### Allegro SMART program

A customer sending parcels from a self-contract using ORLEN Paczka API through any plug-in integrating with ORLEN Parcel services will be billed in accordance with ORLEN Parcel price.

Customer's own agreement with ORLEN Paczka sales representative operates under different invoicing rules than Allegro brokerage agreement.

Shipments to qualify for the Allegro SMART program and be billed according to the price list of this

program, shipping labels must be generated in the “Wysyłam z Allegro” portal or in Allegro integration using the Allegro brokerage agreement.

## 4.1 CREATECUSTOMER

Access to the method is possible only after obtaining permission from the ORLEN Paczka sales representative.

The method is used to create a new ORLEN Parcel customer.

### 4.1.1 Request

**Request parameters:**

<PartnerID>	<b>Customer’s id given by ORLEN Paczka</b>	<b>(REQUIRED) 10 char</b>
<PartnerKey>	<b>Customer’s key given by ORLEN Paczka</b>	<b>(REQUIRED) 10 char</b>
<FirstName>	Recipient’s name	30 char
<LastName>	Recipient’s surname	30 char
<CompanyName>	Recipient’s company name	70 char
<Nip>	<b>Customer’s nip</b>	<b>(REQUIRED) 15 char</b>
<City>	Recipient’s town	30 char
<StreetName>	Recipient’s street name	30 char
<BuildingNumber>	Recipient’s building number	10 char
<PostCode>	Recipient’s postcode	6 char
<EMail>	Recipient’s e-mail address	60 char
<PhoneNumber>	<b>Recipient’s phone number</b>	<b>(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48</b>

When creating a new customer, enter a NIP number consisting of only 10 digits (no spaces or special characters).

**Example with minimum fields required:**

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<FirstName>Jan</FirstName>
<LastName>Nowak</LastName>
<Nip>123-456-78-90</Nip>
<City>Warszawa</City>
<CompanyName>Firma SA</CompanyName>
<StreetName></StreetName>
<BuildingNumber></BuildingNumber>
<FlatNumber></FlatNumber>
<PostCode></PostCode>
<PhoneNumber></PhoneNumber>
<EMail></EMail>
<PhoneNumber></PhoneNumber>
<EMail></EMail>
```

## 4.1.2 Response

Returned values:

<Err>	Error number
<ErrDes>	Error description
<NewCustomerPartnerID>	Newly created customer's ID
<NewCustomerPartnerKey>	Newly created customer's key

Example:

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<NewCustomerPartnerID>1234567890</NewCustomerPartnerID>
<NewCustomerPartnerKey>abcdefghijk</NewCustomerPartnerKey>
```

## 4.2 GENERATEBUSINESSPACK

This method is used for advising the basic ORLEN Paczka parcel.

### 4.2.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	<b>(REQUIRED)</b> 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	<b>(REQUIRED)</b> 10 char
<DestinationCode>	POPeR code	<b>(REQUIRED)</b> 15 char
<AlternativeDestinationCode>	Alternative POPeR code	15 char
<ReturnDestinationCode>	POPeR code for return	15 char
<BoxSize>	Parcel size	<b>(REQUIRED)</b> 1 char
<PackValue>	Parcel value (in grosze)	6 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	
<AmountCashOnDelivery>	Cash on delivery amount (in grosze) (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean
<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<PhoneNumber>	Recipient's phone number	<b>(REQUIRED)</b> 9 char – number without prefix +48 12 char – number with prefix +48
<SenderEMail>	Sender's e-mail address	<b>(REQUIRED)</b> 60 char
<SenderFirstName>	Sender's name	<b>(REQUIRED)</b> 30 char

<SenderLastName>	Sender's surname	(REQUIRED) 30 char
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	Sender's street name	(REQUIRED)30 char
<SenderBuildingNumber>	Sender's building number	(REQUIRED)10 char
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	Sender's city	(REQUIRED) 30 char
<SenderPostCode>	Sender's postcode	(REQUIRED) 6 char
<SenderPhoneNumber>	Sender's phone number	(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48
<SenderOrders>	Sender's reference data	30 char
<ReturnEMail >	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<PrintAdress>	<b>Selection of address for printout in case of return (1 – dispatch address, 2 – return address)</b>	<b>(REQUIRED) 1 char</b>
<ReturnPack>	Return parcel (t – if return)	1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<ReturnAvailable>	Ability to send a return parcel (t -if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijkl</PartnerKey>
<DestinationCode>XX-142450-00-00</DestinationCode>
<FirstName>Jan</FirstName>
```

```

<LastName>Nowak</LastName>
<PhoneNumber>123456789</PhoneNumber>
<SenderEMail>test@tt.pl</SenderEMail>
<SenderCompanyName>Firma SA</SenderCompanyName>
<SenderStreetName>Prosta</SenderStreetName>
<SenderBuildingNumber>1234</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>99-999</SenderPostCode>
<SenderPhoneNumber>123456789</SenderPhoneNumber>
<PrintAdress>1</PrintAdress>

```

## 4.2.2 Response

### Returned values:

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel ean-13 number
<DestinationCode>	POPeR code (confirmation or change)
<DestinationId>	PSD code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)

### Example:

```

<Err>000</Err>
<ErrDes>saved</ErrDes>
<PackCode_RUCH>2900000000230</PackCode_RUCH>
<DestinationCode>XX-142450-00-00</DestinationCode>
<DestinationId>142450</DestinationId>
<PackPrice>745</PackPrice>
<PackPaid>>false</PackPaid>

```

## 4.3 GENERATEBUSINESSPACKTWO

This method is used for advising the basic ORLEN Paczka parcel.

### 4.3.1 Request

#### Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PhoneNumber>	Recipient's phone number	(REQUIRED)

		<b>9 char – number without prefix +48</b> <b>12 char – number with prefix +48</b>
<DestinationCode>	<b>POPeR code</b>	<b>(REQUIRED) 15 char</b>
<AlternativeDestinationCode>	Alternative POPeR code	15 char
<BoxSize>	<b>Parcel size</b>	<b>(REQUIRED) 1 char</b>
<PackValue>	Parcel value (in grosze)	6 char
<CashOnDelivery>	<del>Is the cash on delivery service available?</del> (section 3.12)	
<AmountCashOnDelivery>	<del>Cash on delivery amount (in grosze)</del> (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean
<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<SenderEMail>	<b>Sender's e-mail address</b>	<b>(REQUIRED) 60 char</b>
<SenderFirstName>	<b>Sender's name</b>	<b>(REQUIRED) 30 char</b>
<SenderLastName>	<b>Sender's surname</b>	<b>(REQUIRED) 30 char</b>
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	<b>Sender's street name</b>	<b>(REQUIRED) 30 char</b>
<SenderBuildingNumber>	<b>Sender's building number</b>	<b>(REQUIRED) 10 char</b>
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	<b>Sender's city</b>	<b>(REQUIRED) 30 char</b>
<SenderPostCode>	<b>Sender's postcode</b>	<b>(REQUIRED) 6 char</b>
<SenderPhoneNumber>	<b>Sender's phone number</b>	<b>(REQUIRED)</b> <b>9 char – number without prefix +48</b> <b>12 char – number with prefix +48</b>
<SenderOrders>	Sender's reference data	30 char
<ReturnDestinationCode>	Return POPeR code	15 char
<ReturnEMail >	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<ReturnPack>	Return parcel (t – if return)	1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<PrintAdress>	<b>Selection of address for printout in case of return (1 – dispatch address, 2 – return address)</b>	<b>(REQUIRED) 1 char</b>
<ReturnAvailable>	Ability to send a return parcel (t -if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char

<ConfirmTermsOfUse>	Confirm terms of use	string
<ConfirmMarketing>	Confirm receiving advertising materials	string
<CofirmMailing>	Confirm e-mail address	string

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<FirstName>ImięOdbiorcy</FirstName>
<LastName>NazwiskoOdbiorcy</LastName>
<PhoneNumber>111222333</PhoneNumber>
<DestinationCode>XX-142450-00-00</DestinationCode>
<PrintAdress>1</PrintAdress>
<SenderEMail>mail@mail.pl</SenderEMail>
<SenderFirstName>ImięNadawcy</SenderFirstName>
<SenderLastName>NazwiskoNadawcy</SenderLastName>
<SenderStreetName>Przykładowa</SenderStreetName>
<SenderBuildingNumber>15</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>00-000</SenderPostCode>
<SenderPhoneNumber>123123123</SenderPhoneNumber>
```

## 4.3.2 Response

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel EAN-13 number
<DestinationCode>	POPeR code (confirmation or change)
<DestinationId>	PSD code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)

### Example:

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<PackCode_RUCH>2100123123123</PackCode_RUCH>
<DestinationCode>XX-142450-00-00</DestinationCode>
<DestinationId>142450</DestinationId>
<PackPrice>849</PackPrice>
<PackPaid>>true</PackPaid>
<NameCL>WARSZAWA</NameCL>
<NrCL>0130</NrCL>
<OriginDestinationCode>XX-142450-00-00</OriginDestinationCode>
<AutoChangeDestinationConfirm>0</AutoChangeDestinationConfirm>
```

## 4.4 GENERATELABELBUSINESSPACK

This method is used for advising the ORLEN Paczka parcel, in response it returns a PDF label.

### 4.4.1 Request

Requests' parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<DestinationCode>	POPeR code	(REQUIRED) 15 char
<AlternativeDestinationCode>	Alternative POPeR code	15 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<PackValue>	Parcel value (in grosze)	6 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	
<AmountCashOnDelivery>	Cash on delivery amount (in grosze) (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean
<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<PhoneNumber>	Recipient's phone number	(REQUIRED) 9 char – number without prefix +48

		<b>12 char – number with prefix +48</b>
<SenderEMail>	Sender's e-mail address	<b>(REQUIRED) 60 char</b>
<SenderFirstName>	Sender's name	<b>(REQUIRED) 30 char</b>
<SenderLastName>	Sender's surname	<b>(REQUIRED) 30 char</b>
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	Sender's street name	<b>(REQUIRED) 30 char</b>
<SenderBuildingNumber>	Sender's building number	<b>(REQUIRED) 10 char</b>
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	Sender's city	<b>(REQUIRED) 30 char</b>
<SenderPostCode>	Sender's postcode	<b>(REQUIRED) 6 char</b>
<SenderPhoneNumber>	Sender's phone number	<b>(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48</b>
<SenderOrders>	Sender's reference data	30 char
<ReturnDestinationCode>	Return POPeR code	15 char
<ReturnEMail >	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<ReturnPack>	Return parcel (t – if return)	1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<PrintAdress>	<b>Selection of address for printout in case of return (1 – dispatch address, 2 – return address)</b>	<b>(REQUIRED) 1 char</b>
<ReturnAvailable>	Ability to send a return parcel (t - if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char
<PrintType>	<b>Choice of label for printout (1 - full address, 2 – anonymous address, 3- remain only the names of the recipient and sender)</b>	<b>(REQUIRED) 1 char</b>

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<DestinationCode>XX-142450-00-00</DestinationCode>
<FirstName>Jan</FirstName>
<LastName>Nowak</LastName>
<PhoneNumber>123456789</PhoneNumber>
<SenderEMail>test@tt.pl</SenderEMail>
<SenderCompanyName>Firma SA</SenderCompanyName>
<SenderStreetName>Prosta</SenderStreetName>
<SenderBuildingNumber>1234</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>99-999</SenderPostCode>
<SenderPhoneNumber>123456789</SenderPhoneNumber>
<PrintAddress>1</PrintAddress>
<PrintType>1</PrintType>
```

## 4.4.2 Response

PDF label is returned in the <LabelData> tag.

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel EAN-13 number
<DestinationCode>	POPeR code (confirmation or change)
<DestinationId>	PSD code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)
<LabelData>	PDF label (Base 64)

#### Example:

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<PackCode_RUCH>2100123123123</PackCode_RUCH>
<DestinationCode>XX-142450-00-00</DestinationCode>
<DestinationId>142450</DestinationId>
<PackPrice>745</PackPrice>
<PackPaid>>false</PackPaid>
<LabelData>label decoded in BASE64</LabelData>
```

#### C# code sample with web service call:

```
PwR.WebServicePwRTest service = new PwR.WebServicePwRTest();
service.UseDefaultCredentials = true;
byte[] data;
```

```

service.GenerateLabelBusinessPack("<user>", "<password>", "<destination code>", null, null, null, null,
null, null, null, "<first name>", "last name",
"<company>", "<address>", "<number>", null, "<city>", "<postal code>", "<phone>", "<email>",
null, null, "<sender company>", "<adress>", "number", null, "<city>", "<postal code>",
"<sender phone>", null, null, null, null, null, null, null, null, null, null, null, null, null, "1", null,
null, "1", out data);
    if (data != null)
    {
        File.WriteAllBytes("c:\\Paczka_test\\label.pdf", data);}

```

## 4.5 GENERATELABELBUSINESSPACKLIST

This method is used to advise a single parcel or a list of ORLEN Paczka parcels, in response it returns a PDF, PDF10, EPL label. The maximum number of labels returned at one time is 50.

### 4.5.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<Format>	Label format (pdf, pdf10 or epl)	(REQUIRED) 3 char
BusinessPackList	Structure of „businesspack” for the parcels' list	(REQUIRED) (quantity: 1 ... 50)

BusinessPack structure		
<DestinationCode>	POPeR code	(REQUIRED) 15 char
<AlternativeDestinationCode>	Alternative po POPeR per code	15 char
<ReturnDestinationCode>	POPeR code for return	15 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<PackValue>	Parcel value (in grosze)	6 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	
<AmountCashOnDelivery>	Cash on delivery amount (in grosze) (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean
<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<PhoneNumber>	Recipient's phone number	(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48
<SenderEMail>	Sender's e-mail address	(REQUIRED) 60 char

<SenderFirstName>	Sender's name	(REQUIRED) 30 char
<SenderLastName>	Sender's surname	(REQUIRED) 30 char
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	Sender's street name	(REQUIRED) 30 char
<SenderBuildingNumber>	Sender's building number	(REQUIRED) 10 char
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	Sender's city	(REQUIRED) 30 char
<SenderPostCode>	Sender's postcode	(REQUIRED) 6 char
<SenderPhoneNumber>	Sender's phone number	(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48
<SenderOrders>	Sender's reference data	30 char
<ReturnEMail >	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<ReturnPack>	Return parcel (t – if return)	1 char
<PrintAddress>	Selection of address for printout in case of return (1 – dispatch address, 2 – return address)	(REQUIRED) 1 char
<PrintType>	Choice of label for printout (1 - full address, 2 – anonymous address, 3- remain only the names of the recipient and sender)	(REQUIRED) 1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<ReturnAvailable>	Ability to send a return parcel (t - if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

#### Example with minimum fields required:

<PartnerID>1234567890</PartnerID>

<PartnerKey>abcdefghijkl</PartnerKey>

```

<Format>PDF</Format>
<BusinessPackList>

  <BusinessPack>
    <DestinationCode>XX-142450-00-00 </DestinationCode>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <PhoneNumber>123456789</PhoneNumber>
    <SenderEMail>test@tt.pl</SenderEMail>
    <SenderCompanyName>Firma SA</SenderCompanyName>
    <SenderStreetName>Prosta</SenderStreetName>
    <SenderBuildingNumber>1234</SenderBuildingNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>99-999</SenderPostCode>
    <SenderPhoneNumber>123456789</SenderPhoneNumber>
    <SenderOrders>dane referencyjne</ SenderOrders>
    <PrintAdress>1</PrintAdress>
    <PrintType>1</PrintType>
  </BusinessPack>

</BusinessPackList>

```

## 4.5.2 Response

The label is returned in the <LabelData> tag.

### Returned values:

<GenerateLabelBusinessPackList> - parcels' list (1 ... n)

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel EAN-13 number
<DestinationCode>	POPeR code (confirmation or change)
<DestinationId>	PSD code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)
<SenderOrders>	Sender's reference data

<GenerateLabelBusinessPackList>

<LabelData>	PDF, PDF10, EPL label (Base 64)
-------------	---------------------------------

### Example:

```
<GenerateLabelBusinessPackList>
```

```

<Err>000</Err>
<ErrDes>saved</ErrDes>
<PackCode_RUCH>2100123123123</PackCode_RUCH>
<DestinationCode>XX-142450-00-00</DestinationCode>
<DestinationId>142450</DestinationId>
<PackPrice>745</PackPrice>
<PackPaid>>false</PackPaid>
<SenderOrders>reference data</SenderOrders>

</GenerateLabelBusinessPackList>

<LabelData>label encoded in BASE64</LabelData>

```

## 4.6 GENERATELABELBUSINESSPACKLISTTWO

This method is used to advise a single parcel or a list of ORLEN Paczka parcels, in response it returns a PDF, PDF10, EPL, ZPL, ZPL 300 DPI label. The maximum number of labels returned at one time is 50.

### 4.6.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<Format>	Label format (pdf, pdf10, epl or zpl, zpl_300dpi)	(REQUIRED) 10 char
BusinessPackList	Structure of „businesspack” for the list of parcels	(REQUIRED) (quantity: 1 ... 50)

BusinessPack structure		
<DestinationCode>	POPeR code	(REQUIRED) 15 char
<AlternativeDestinationCode>	Alternative POPeR code	15 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<PackValue>	Parcel value (in grosze)	6 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	
<AmountCashOnDelivery>	Cash on delivery amount (in grosze) (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean
<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char

<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<PhoneNumber>	Recipient's phone number	<b>(REQUIRED)</b> 9 char – number without prefix +48 12 char – number with prefix +48
<SenderEMail>	Sender's e-mail address	<b>(REQUIRED)</b> 60 char
<SenderFirstName>	Sender's name	<b>(REQUIRED)</b> 30 char
<SenderLastName>	Sender's surname	<b>(REQUIRED)</b> 30 char
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	Sender's street name	<b>(REQUIRED)</b> 30 char
<SenderBuildingNumber>	Sender's building number	<b>(REQUIRED)</b> 10 char
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	Sender's city	<b>(REQUIRED)</b> 30 char
<SenderPostCode>	Sender's postcode	<b>(REQUIRED)</b> 6 char
<SenderPhoneNumber>	Sender's phone number	<b>(REQUIRED)</b> 9 char – number without prefix +48 12 char – number with prefix +48
<SenderOrders>	Sender's reference data	30 char
<ReturnDestinationCode>	Return POPEr code	15 char
<ReturnEMail >	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<ReturnPack>	Return parcel (t – if return)	1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<PrintAdress>	<b>Selection of address for printout in case of return (1 – dispatch address, 2 – return address)</b>	<b>(REQUIRED)</b> 1 char
<ReturnAvailable>	Ability to send a return parcel (t - if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char
<PrintType>	<b>Choice of label for printout (1 - full address, 2 – anonymous address, 3- remain only the names of the recipient and sender)</b>	<b>(REQUIRED)</b> 1 char

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

**Example with minimum fields required:**

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<Format>PDF</Format>
<BusinessPackList>

  <BusinessPack>
    <DestinationCode>XX-142450-00-00</DestinationCode>
    <BoxSize></BoxSize>
    <PackValue>899</PackValue>
    <EMail>test@mail.com</EMail>
    <FirstName>Zenon</FirstName>
    <LastName>Zenonowicz</LastName>
    <CompanyName></CompanyName>
    <StreetName>Testowinska</StreetName>
    <BuildingNumber>7</BuildingNumber>
    <City>Warszawa</City>
    <PostCode>00-000</PostCode>
    <PhoneNumber>111555899</PhoneNumber>
    <SenderEMail>nadawca@test.com</SenderEMail>
    <SenderFirstName>test</SenderFirstName>
    <SenderLastName>test</SenderLastName>
    <SenderStreetName>Stalowa</SenderStreetName>
    <SenderBuildingNumber>89</SenderBuildingNumber>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>999666333</SenderPhoneNumber>
    <PrintAdress>1</PrintAdress>
    <PrintType>1</PrintType>
  </BusinessPack>
</BusinessPackList>
```

### 4.6.2 Response

The label is returned in the <LabelData> tag.

**Returned values:**

<GenerateLabelBusinessPackList> - list of parcels (1 ... n)

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel EAN-13 number

<DestinationCode>	POPeR code (confirmation or change)
<DestinationID>	PSD code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)
<ReturnDestinationId>	ID pick-up point for return
<ReturnDestinationCode>	POPeR code for return
<NameCL>	Region name for the pick-up point
<NrCL>	Region ID for the pick-up point
<OriginDestinationCode>	POPeR code for point of dispatch
<AutoChangeDestinationConfirm>	0 or 1

<GenerateLabelBusinessPackListTwo>

<LabelData>	Etykieta PDF, PDF10, EPL, ZPL, ZPL 300 DPI (Base 64)
-------------	--

### Example:

```

<Err>000</Err>
<ErrDes>saved</ErrDes>
<PackCode_RUCH>2100123123123</PackCode_RUCH>
<DestinationCode>XX-142450-00-00</DestinationCode>
<DestinationId>142450</DestinationId>
<PackPrice>849</PackPrice>
<PackPaid>>true</PackPaid>
<NameCL>WARSZAWA</NameCL>
<NrCL>0130</NrCL>
<OriginDestinationCode>XX-142450-00-00</OriginDestinationCode>
<AutoChangeDestinationConfirm>0</AutoChangeDestinationConfirm>

```

## 4.7 GENERATELABELBUSINESSPACKTWO

This method is used for advising the basic ORLEN Paczka parcel, in response it returns a PDF label.

### 4.7.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<DestinationCode>	POPeR code	(REQUIRED) 15 char
<AlternativeDestinationCode>	Alternative POPeR code	15 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<PackValue>	Parcel value (in grosze)	6 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	
<AmountCashOnDelivery>	Cash on delivery amount (in grosze) (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean

<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<PhoneNumber>	Recipient's phone number	<b>(REQUIRED)</b> 9 char – number without prefix +48 12 char – number with prefix +48
<SenderEMail>	Sender's e-mail address	<b>(REQUIRED)</b> 60 char
<SenderFirstName>	Sender's name	<b>(REQUIRED)</b> 30 char
<SenderLastName>	Sender's surname	<b>(REQUIRED)</b> 30 char
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	Sender's street name	<b>(REQUIRED)</b> 30 char
<SenderBuildingNumber>	Sender's building number	<b>(REQUIRED)</b> 10 char
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	Sender's city	<b>(REQUIRED)</b> 30 char
<SenderPostCode>	Sender's postcode	<b>(REQUIRED)</b> 6 char
<SenderPhoneNumber>	Sender's phone number	<b>(REQUIRED)</b> 9 char – number without prefix +48 12 char – number with prefix +48
<SenderOrders>	Sender's reference data	30 char
<ReturnDestinationCode>	POPeR code for return	15 char
<ReturnEMail >	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<ReturnPack>	Return parcel (t – if return)	1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<PrintAdress>	<b>Selection of address for printout in case of return (1 – dispatch address, 2 – return address)</b>	<b>(REQUIRED)</b> 1 char
<ReturnAvailable>	Ability to send a return parcel (t - if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char
<PrintType>	<b>Choice of label for printout (1 - full address, 2 – anonymous address, 3- remain only the names of the recipient and sender)</b>	<b>(REQUIRED)</b> 1 char

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijkl</PartnerKey>
<DestinationCode>XX-142450-00-00</DestinationCode>
<FirstName>Jan</FirstName>
<LastName>Nowak</LastName>
<PhoneNumber>123456789</PhoneNumber>
<SenderEMail>test@tt.pl</SenderEMail>
<SenderCompanyName>Firma SA</SenderCompanyName>
<SenderStreetName>Prosta</SenderStreetName>
<SenderBuildingNumber>1234</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>99-999</SenderPostCode>
<SenderPhoneNumber>123456789</SenderPhoneNumber>
<PrintAdress>1</PrintAdress>
<PrintType>1</PrintType>
```

## 4.7.2 Response

The PDF label is returned in the <LabelData> tag.

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel ean-13 number
<DestinationCode>	POPeR code (confirmation or change)
<DestinationId>	PSD code
<PackPrice>	Parcel price (in groszy)
<PackPaid>	Parcel payment (true/false)
<LabelData>	Pdf label (base 64)

#### Example:

```

<Err>000</Err>
<ErrDes>saved</ErrDes>
<PackCode_RUCH>2100123123123</PackCode_RUCH>
<DestinationCode>XX-142450-00-00</DestinationCode>
<DestinationId> 142450 </DestinationId>
<PackPrice>745</PackPrice>
<PackPaid>>false</PackPaid>
<LabelData>label encoded in BASE64</LabelData>

```

## 4.8 GENERATENUMBERWAYBILLS

This method is used to reserve a pool of ORLEN Paczka numbers.

Transmission of the size of the pool of reserved letter numbers is declared in the NumberOfWaybills parameter.

The maximum pool size in a single call is 10000.

Reserved shipment numbers have no expiration date.

For shipment numbers reserved by this method, the advisory data should be transmitted using the PutCustomerPack method (section 4.34).

### 4.8.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<NumberOfWaybills>	Number of waybills to be generated	(REQUIRED) 5 char

**Example:**

```

<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<NumberOfWaybills>2</NumberOfWaybills>

```

### 4.8.2 Response

**Returned values:**

<PackCode_RUCH>	Parcel ean-13 number
-----------------	----------------------

**Example:**

```

<PackCode_RUCH>210000000230</PackCode_RUCH>
<PackCode_RUCH>210000000231</PackCode_RUCH>

```

## 4.9 GENERATEPROTOCOL

This method is used to generate protocol (in PDF format) for given parcel numbers that were advised within ORLEN Paczka service. Additionally this method returns incorrect parcels (all that have other status than advised) as well as correct parcels with protocol number.

The maximum number of packages declared on a single protocol is 571.

Declaration of the parcel to be included in the protocol is done by entering the number in the <unsignedLong> tag of the <parcels></parcels> object (maximum number of <unsignedLong> tags is 571).

Time returned in <DATA\_MOD> tag is in the UTC+1 time zone (local Polish time).

### 4.9.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<parcels>	Parcel ean-13 numbers	(REQUIRED) List<unsignedLong>

**Example with minimum fields required:**

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>

<parcels>

    <unsignedLong>2100123123123</unsignedLong>
    <unsignedLong>2100123123124</unsignedLong>
    <unsignedLong>2100123123125</unsignedLong>

</parcels>
```

### 4.9.2 Response

**Returned values:**

PDF protocol is returned in the <LabelData> tag.

<Err>	Error number (when err 210 – parcel with other status than advised- 200, will not be included in protocol)
<ErrDes>	Error description
<ProtocolCode>	Protocol number (if err = 0)
<PackCode_RUCH>	Parcel EAN-13 number
<DATA_MOD>	Date of status modification
<status>	Status number

<status_opis>	Status description
<LabelData>	Protocol in PDF (BASE64)

### Example:

Shipment added to the transfer protocol:

```
<Err>0</Err>
<ErrDes>OK</ErrDes>
<ProtocolCode>1000185903089</ProtocolCode>
<PackCodeRUCH>2101142967553</PackCodeRUCH>
<DATA_MOD>2024-10-22T13:12:55+02:00</DATA_MOD>
<status>200</status>
<status_opis>Brak Paczki</status_opis>
```

Shipment not added to the transfer protocol:

```
<Err>210</Err>
<ErrDes>NIE DODANE DO PROTOKOŁU</ErrDes>
<PackCodeRUCH>2101142967553</PackCodeRUCH>
<DATA_MOD>2024-10-22T13:15:12+02:00</DATA_MOD>
<status>201</status>
<status_opis>Brak Paczki</status_opis>
```

Shipment not added to the transfer protocol:

```
<Err>210</Err>
<ErrDes>NIE DODANE DO PROTOKOŁU</ErrDes>
<PackCodeRUCH>2101142968208</PackCodeRUCH>
<DATA_MOD>2024-10-22T13:16:20+02:00</DATA_MOD>
<status>400</status>
<status_opis>Brak Paczki</status_opis>
```

## 4.10 GENERATE SHIPPING CODE

This method is used for advising the ORLEN Paczka parcel by generating shipping code (WhiteLabel).

### 4.10.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PhoneNumber>	Recipient's phone number	(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48
<DestinationCode>	POPeR code	(REQUIRED) 15 char
<AlternativeDestinationCode>	Alternative POPeR code	15 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<PackValue>	Parcel value (in grosze)	6 char

<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	
<AmountCashOnDelivery>	Cash on delivery amount (in grosze) (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean
<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<SenderEMail>	<b>Sender's e-mail address</b>	<b>(REQUIRED) 60 char</b>
<SenderFirstName>	<b>Sender's name</b>	<b>(REQUIRED) 30 char</b>
<SenderLastName>	<b>Sender's surname</b>	<b>(REQUIRED) 30 char</b>
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	<b>Sender's street name</b>	<b>(REQUIRED) 30 char</b>
<SenderBuildingNumber>	<b>Sender's building number</b>	<b>(REQUIRED) 10 char</b>
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	<b>Sender's city</b>	<b>(REQUIRED) 30 char</b>
<SenderPostCode>	<b>Sender's postcode</b>	<b>(REQUIRED) 6 char</b>
<SenderPhoneNumber>	<b>Sender's phone number</b>	<b>(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48</b>
<SenderOrders>	Sender's reference data	30 char
<ReturnDestinationCode>	POPeR code for return	15 char
<ReturnEMail >	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<ReturnPack>	Return parcel (t – if return)	1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<PrintAdress>	<b>Selection of address for printout in case of return (1 – dispatch address, 2 – return address)</b>	<b>(REQUIRED) 1 char</b>
<ReturnAvailable>	Ability to send a return parcel (t - if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijkl</PartnerKey>
<DestinationCode>XX-142450-00-00</DestinationCode>
<FirstName>Jan</FirstName>
<LastName>Nowak</LastName>
<PhoneNumber>123456789</PhoneNumber>
<SenderEMail>test@tt.pl</SenderEMail>
<SenderCompanyName>Firma SA</SenderCompanyName>
<SenderStreetName>Prosta</SenderStreetName>
<SenderBuildingNumber>1234</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>99-999</SenderPostCode>
<SenderPhoneNumber>123456789</SenderPhoneNumber>
<PrintAdress>1</PrintAdress>
```

## 4.10.2 Response

Shipping code is returned in the <ShippingCode> tag.

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<ShippingCode>	Parcel shipping code
<DestinationCode>	POPeR code (confirmation or change)
<DestinationID>	PSD code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)

### Example:

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<ShippingCode>1234567</ShippingCode>
<DestinationCode>XX-142450-00-00</DestinationCode>
<DestinationId>142450</DestinationId>
<PackPrice>745</PackPrice>
<PackPaid>>false</PackPaid>
```

## 4.11 GETGENERATEDPARCELS

This method is used to retrieve a pool of numbers for ORLEN Paczka parcels with the notification date for given date range. As a result the method will return parcel numbers notified in the last 30 or fewer days.

Time returned in <Data> tag is in the UTC+1 time zone (local Polish time).

### 4.11.1 Request

#### Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<dateFrom>	Starting date	(YYYY-MM-DD) dateTime
<dateTo>	End date	(YYYY-MM-DD) dateTime

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<dateFrom>2024-07-01</dateFrom>
<dateTo>2024-07-15</dateTo>
```

### 4.11.2 Response

#### Returned values:

<GetGeneratedParcels> - list of parcels (1 ... n)

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel EAN-13 number
<Data>	Parcel notification date

**Example:**

```
<Err>000</Err>  
<ErrDes>saved</ErrDes>  
<PackCode_RUCH>2100123123123</PackCode_RUCH>  
<Data>2024-07-01T00:13:33.757+02:00</Data>
```

## 4.12 GIVECURRENTPACK

This method is used to retrieve current parcel number in case when return label has been generated in addition to original parcel label.

### 4.12.1 Request

**Request parameters:**

<packCode>	Package number	(REQUIRED) 13 char
------------	----------------	--------------------

**Example with minimum fields required:**

```
<packCode>2100123123123</packCode>
```

### 4.12.2 Response

**Returned values:**

```
<GiveCurrentPackResult>string</GiveCurrentPackResult>
```

GiveCurrentPackResult	Current parcel number
-----------------------	-----------------------

**Example:**

```
<GiveCurrentPackResult>2600123123123</GiveCurrentPackResult>
```

## 4.13 GIVEMASTERPACK

This method is used to retrieve original parcel number while having only a return parcel number.

### 4.13.1 Request

Request parameters:

<packCode>	Package number	(REQUIRED) 13 char
------------	----------------	--------------------

Example with minimum fields required:

```
<packCode>2600123123123</packCode>
```

### 4.13.2 Response

Returned values:

```
<GiveMasterPackResult>string</GiveMasterPackResult>
```

GiveMasterPackResult	Original parcel number
----------------------	------------------------

Example:

```
<GiveMasterPackResult>2100123123123</GiveMasterPackResult>
```

## 4.14 GIVEMEALLPSDLOCATION

This method is inactive as of January 1, 2025 due to the closure of PSD points and the discontinuation of the COD service.

This method returns a list of PSD (Kiosks RUCH) points with an extended location description.

### 4.14.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char

Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>  
<PartnerKey>abcdefghijk</PartnerKey>
```

### 4.14.2 Response

Returned values:

<DestinationCode>	POPeR code	15 char
<StreetName>	Street on which the parcel collection point (pop) is located	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	City where the pop is located	40 char
<District>	District where the pop is located	30 char
<Latitude>	Latitude of the pop	10 char
<Longitude>	Longitude of the pop	10 char
<Province>	Province	20 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	(false) boolean
<OpeningHours>	Information about pop opening hours	50 char
<Location>	Description of the location of the parcel collection point	255 char
<PSD>	PSD number	6 char
<Available>	Is the pop available (y/n)	1 char

#### Example:

```

<DestinationCode>RZ-395162-KK-35</DestinationCode>
<StreetName>WARSZAWSKA 75A</StreetName>
<BuildingNumber>82</BuildingNumber>
<City>Rzeszów</City>
<District>Rzeszów</District>
<Longitude>21.99427</Longitude>
<Latitude>50.05775</Latitude>
<Province>Podkarpackie</Province>
<CashOnDelivery>>false</CashOnDelivery>
<OpeningHours>Pn-Pt:00:00-24:00, So:00:00-24:00, Nd:00:00-24:00</OpeningHours>
<Location>Punkt odbioru paczek znajduje się w budynku Stacji Paliw ORLEN</Location>
<PSD>395162</PSD>
<Available>T</Available>
<PointType>PKN</PointType>

```

## 4.15 GIVEMEALLLOCATIONWITHALLDATA

The method returns extended data for all points (APM, PKN, PSD, PPP, PPK) available in the ORLEN Paczka service.

### 4.15.1 Request

Request parameters:

NONE		
------	--	--

### 4.15.2 Response

Returned values:

<DestinationCode>	Destination code	15 char
<StreetName>	Customer street name	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	Customer city	40 char
<District>	District	30 char
<Longitude>	Longitude	10 char
<Latitude>	Latitude	10 char
<Province>	Province	20 char
<CashOnDelivery>	Availability of cash on delivery service at the point of collection (section 3.12)	(false) boolean
<OpeningHours>	Opening hours of the pick-up point	50 char
<Location>	Description of the location of the collection point	255 char
<PSD>	Unique 6-digit id 44umer of the collection point	6 char
<Available>	Availability of a point in the orlen parcel service	(T/N) 1 char
<Obszar>	Collection point area	20 char
<Mikrorejon>	Micro-district of collection point	20 char
<Skrotnrpok>	Abbreviation of the id 44umer of the collection point	20 char
<Sortownia>	Sorting plant to which the collection point is assigned	20 char
<PointType>	Type of point (own – psd, partner – psp, ppp, psf, ppk, parcel machine – apm, petrol station orlen – pkn)	3 char
<Presort>	Presort of collection point	20 char
<Czas>	Parameter for courier needs, regarding point availability time for courier deliveries	20 char
<NearestPoint>	Structure “nearestpoint” – returns the address of the three nearest points relative to the original point	3 points

### Example:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GiveMeAllLocationWithAllDataResponse xmlns="https://91.242.220.103/WebServicePwR">
      <GiveMeAllLocationWithAllDataResult>
        <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
          <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
            <xs:complexType>
              <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="LocationWithAllData">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="DestinationCode" type="xs:string" minOccurs="0" />
                      <xs:element name="StreetName" type="xs:string" minOccurs="0" />
                      <xs:element name="BuildingNumber" type="xs:string" minOccurs="0" />
                      <xs:element name="City" type="xs:string" minOccurs="0" />
                      <xs:element name="District" type="xs:string" minOccurs="0" />
                    
```

```

        <xs:element name="Longitude" type="xs:string" minOccurs="0" />
        <xs:element name="Latitude" type="xs:string" minOccurs="0" />
        <xs:element name="Province" type="xs:string" minOccurs="0" />
        <xs:element name="CashOnDelivery" type="xs:boolean" minOccurs="0" />
        <xs:element name="OpeningHours" type="xs:string" minOccurs="0" />
        <xs:element name="Location" type="xs:string" minOccurs="0" />
        <xs:element name="PSD" type="xs:string" minOccurs="0" />
        <xs:element name="Available" type="xs:string" minOccurs="0" />
        <xs:element name="Obszar" type="xs:string" minOccurs="0" />
        <xs:element name="Mikrorejon" type="xs:string" minOccurs="0" />
        <xs:element name="Skrotnrpok" type="xs:string" minOccurs="0" />
        <xs:element name="Sortownia" type="xs:string" minOccurs="0" />
        <xs:element name="Presort" type="xs:string" minOccurs="0" />
        <xs:element name="Czas" type="xs:string" minOccurs="0" />
        <xs:element name="PointType" type="xs:string" minOccurs="0" />
        <xs:element name="NearestPoints" msdata:DataType="System.Collections.Generic.List`1[[Pwr.Common.Models.NearestPoint, Pwr.Common.Models, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null]], mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" type="xs:anyType" minOccurs="0" />
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
    <NewDataSet xmlns="">
        <LocationWithAllData diffgr:id="LocationWithAllData1" msdata:rowOrder="0">
            <DestinationCode>WS-100001-27-26</DestinationCode>
            <StreetName>ANNOPOL</StreetName>
            <BuildingNumber>17</BuildingNumber>
            <City>Warszawa</City>
            <District>Warszawa</District>
            <Longitude>21.013830</Longitude>
            <Latitude>52.311519</Latitude>
            <Province>Mazowieckie</Province>
            <CashOnDelivery>>false</CashOnDelivery>
            <OpeningHours>Pn-Pt:00:00-24:00, So:00:00-24:00, Nd:00:00-24:00</OpeningHours>
            <Location>Punkt testowy</Location>
            <PSD>100001</PSD>
            <Available>T</Available>
            <Obszar>MIN</Obszar>
            <Mikrorejon>WSK6</Mikrorejon>
            <Skrotnrpok>K6-55</Skrotnrpok>
            <Sortownia />
            <Czas>XX</Czas>
            <PointType>PSD</PointType>
            <NearestPoints>
                <NearestPoint>

```

```

    <Name>Warszawa Chodecka 13 </Name>
    <Distance>2179</Distance>
  </NearestPoint>
  <NearestPoint>
    <Name>Warszawa Chodecka 13 </Name>
    <Distance>2179</Distance>
  </NearestPoint>
  <NearestPoint>
    <Name>Warszawa Chodecka 13 </Name>
    <Distance>2179</Distance>
  </NearestPoint>
</NearestPoints>
</LocationWithAllData>

```

## 4.16 GIVEMEALLOCATIONWITHALLDATAWITHZIPCODE

The method returns extended data with zipcode for all points (APM, PKN, PSD, PPP, PPK) available in the ORLEN Paczka service.

### 4.16.1 Request

Request parameters:

NONE		
------	--	--

### 4.16.2 Response

Returned values:

<DestinationCode>	Destination code	15 char
<StreetName>	Customer street name	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	Customer city	40 char
<Zipcode>	Zipcode	6 char
<District>	District	30 char
<Longitude>	Longitude	10 char
<Latitude>	Latitude	10 char
<Province>	Province	20 char
<CashOnDelivery>	Availability of cash on delivery service at the point of collection (section 3.12)	(false) boolean
<OpeningHours>	Opening hours of the pick-up point	50 char
<Location>	Description of the location of the collection point	255 char
<PSD>	Unique 6-digit id 46umer of the collection point	6 char
<Available>	Availability of a point in the orlen parcel service	(T/N) 1 char
<Obszar>	Collection point area	20 char
<Mikrorejon>	Micro-district of collection point	20 char
<Skrotrnpok>	Abbreviation of the id 46umer of the collection point	20 char

<Sortownia>	Sorting plant to which the collection point is assigned	20 char
<PointType>	Type of point (own – psd, partner – psp, ppp, psf, ppk, parcel machine – apm, petrol station orlen – pkn)	3 char
<Presort>	Presort of collection point	20 char
<Czas>	Parameter for courier needs, regarding point availability time for courier deliveries	20 char
<NearestPoint>	Structure “nearestpoint” – returns the address of the three nearest points relative to the original point	3 points

### Example:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GiveMeAllLocationWithAllDataWithZipCodeResponse xmlns="https://91.242.220.103/WebServicePwR">
      <GiveMeAllLocationWithAllDataWithZipCodeResult>
        <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
          <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
            <xs:complexType>
              <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="LocationWithAllData2">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="DestinationCode" type="xs:string" minOccurs="0" />
                      <xs:element name="StreetName" type="xs:string" minOccurs="0" />
                      <xs:element name="BuildingNumber" type="xs:string" minOccurs="0" />
                      <xs:element name="City" type="xs:string" minOccurs="0" />
                      <xs:element name="Zipcode" type="xs:string" minOccurs="0" />
                      <xs:element name="District" type="xs:string" minOccurs="0" />
                      <xs:element name="Longitude" type="xs:string" minOccurs="0" />
                      <xs:element name="Latitude" type="xs:string" minOccurs="0" />
                      <xs:element name="Province" type="xs:string" minOccurs="0" />
                      <xs:element name="CashOnDelivery" type="xs:boolean" minOccurs="0" />
                      <xs:element name="OpeningHours" type="xs:string" minOccurs="0" />
                      <xs:element name="Location" type="xs:string" minOccurs="0" />
                      <xs:element name="PSD" type="xs:string" minOccurs="0" />
                      <xs:element name="Available" type="xs:string" minOccurs="0" />
                      <xs:element name="Obszar" type="xs:string" minOccurs="0" />
                      <xs:element name="Mikrorejon" type="xs:string" minOccurs="0" />
                      <xs:element name="Skrotnrpok" type="xs:string" minOccurs="0" />
                      <xs:element name="Sortownia" type="xs:string" minOccurs="0" />
                      <xs:element name="Presort" type="xs:string" minOccurs="0" />
                      <xs:element name="Czas" type="xs:string" minOccurs="0" />
                      <xs:element name="PointType" type="xs:string" minOccurs="0" />
                      <xs:element name="NearestPoints"
msdata:DataType="System.Collections.Generic.List`1[[Pwr.Common.Models.NearestPoint,
```



<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char

#### 4.17.2 Response

##### Returned values:

<DestinationCode>	POPeR code	15 char
<StreetName>	Street on which the parcel collection point (pop) is located	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	City where the pop is located	40 char
<District>	District where the pop is located	30 char
<Zipcode>	Pop zipcode	6 char
<Longitude>	Longitude of the pop	10 char
<Latitude>	Latitude of the pop	10 char
<Province>	Province	20 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	(false) boolean
<OpeningHours>	Information about pop opening hours	50 char
<Location>	Description of the location of the parcel collection point	255 char
<PSD>	PSD number	6 char
<Available>	Is the pop available (y/n)	1 char
<PointType>	Point type (parcel machine – apm, orlen petrol station – pkn)	3 char

##### Example:

```

<AllApmLocation diffgr:id="AllApmLocation3" msdata:rowOrder="2">
  <DestinationCode>LB-524373-20-13</DestinationCode>
  <StreetName>LUBARTOWSKA</StreetName>
  <BuildingNumber>12</BuildingNumber>
  <City>Kamionka</City>
  <District>Kamionka</District>
  <Zipcode>21-132</Zipcode>
  <Longitude>22.4643</Longitude>
  <Latitude>51.4695</Latitude>
  <Province>Lubelskie</Province>
  <CashOnDelivery>>false</CashOnDelivery>
  <OpeningHours>Pn-Pt:08:00-17:00, So:08:00-15:00, Nd: nieczynne</OpeningHours>
  <Location>Punkt odbioru paczek znajduje się w budynku Stacji Paliw ORLEN</Location>
  <PSD>524373</PSD>
  <Available>T</Available>
</AllApmLocation>

```

## 4.18 GIVEMEALLRUCHLOCATION

This method returns a list of all ORLEN Paczka points (APM, PKN, PSD, PPP, PPK) with an extended location description.

### 4.18.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char

### 4.18.2 Response

Returned values:

<DestinationCode>	POPeR code	15 char
<StreetName>	Street on which the parcel collection point (pop) is located	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	City where the pop is located	40 char
<District>	District where the pop is located	30 char
<Latitude>	Latitude of the pop	10 char
<Longitude>	Longitude of the pop	10 char
<Province>	Province	20 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	(false) boolean
<OpeningHours>	Information about pop opening hours	50 char
<Location>	Description of the location of the parcel collection point	255 char
<PSD>	PSD number	6 char
<Available>	Is the pop available (y/n)	1 char
<PointType>	Type of point (own – psd, partner – psp, ppp, psf, ppk, parcel machine – apm, petrol station orlen – pkn)	3 char

Example:

```
<DestinationCode>BD-105239-46-43</DestinationCode>
<StreetName>POZNAŃSKA</StreetName>
<BuildingNumber>5</BuildingNumber>
<City>Inowrocław</City>
<District>Inowrocław</District>
```

<Longitude>18.2615</Longitude>  
 <Latitude>52.7936</Latitude>  
 <Province>Kujawsko-Pomorskie</Province>  
 <CashOnDelivery>>false</CashOnDelivery>  
 <OpeningHours>Pn-Pt:06:00-18:00, So:07:00-14:00, Nd: nieczynne</OpeningHours>  
 <Location>Automat paczkowy obok sklepu</Location>  
 <PSD>105239</PSD>  
 <Available>T</Available>  
 <PointType>PSD</PointType>

## 4.19 GIVEMEALLRUCHZIPCODE

This method returns a list of ORLEN Paczka points (APM, PKN, PSD, PPP, PPK) with an extended location description and additional zipcode information.

### 4.19.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char

### 4.19.2 Response

Returned values:

<DestinationCode>	POPeR code	15 char
<StreetName>	Street on which the parcel collection point (pop) is located	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	City where the pop is located	40 char
<District>	District where the pop is located	30 char
<Latitude>	Latitude of the pop	10 char
<Longitude>	Longitude of the pop	10 char
<Province>	Province	20 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	(false) boolean
<OpeningHours>	Information about pop opening hours	50 char
<Location>	Description of the location of the parcel collection point	255 char
<PSD>	PSD number	6 char
<Available>	Is the pop available (y/n)	1 char
<Zipcode>	Pop zipcode	6 char

### Example:

```
<DestinationCode>BD-105239-46-43</DestinationCode>
<StreetName>POZNAŃSKA</StreetName>
<BuildingNumber>5</BuildingNumber>
<City>Inowrocław</City>
<District>Inowrocław</District>
<Longitude>18.2615</Longitude>
<Latitude>52.7936</Latitude>
<Province>Kujawsko-Pomorskie</Province>
<CashOnDelivery>>false</CashOnDelivery>
<OpeningHours>Pn-Pt:06:00-18:00, So:07:00-14:00, Nd: nieczynne</OpeningHours>
<Location>Automat paczkowy obok sklepu</Location>
<PSD>105239</PSD>
<Available>T</Available>
<PointType>PSD</PointType>
<Zipcode>12-345</Zipcode>
```

## 4.20 GIVEMEALLRUCHWITHFILLED

Functionality of indicating overflow of pickup point has been disabled.

This method returns a list of all ORLEN Paczka points (APM, PKN, PSD, PPP, PPK) with an extended location description. ORLEN Paczka points that are temporary filled will not be excluded or hidden. In case of mentioned overflow, the method will return up to 5 alternative points within 1.5 km from filled pick-up point

### 4.20.1 Request

#### Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
```

### 4.20.2 Response

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<Data>	List of pick-up points within orlen paczka service

Data structure	
<PointPwR>	Pick-up point data within orlen paczka service

PointPwR structure		
<DestinationCode>	POPeR code	15 char
<StreetName>	Street on which the parcel collection point (pop) is located	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	City where the pop is located	40 char
<District>	District where the pop is located	30 char
<Latitude>	Latitude of the pop	10 char
<Longitude>	Longitude of the pop	10 char
<Province>	Province	20 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	false (boolean)
<OpeningHours>	Information about pop opening hours	50 char
<Location>	Description of the location of the parcel collection point	255 char
<PSD>	PSD number	6 char
<PointType>	Type of point (own – psd, partner – psp, ppp, psf, ppk, parcel machine – apm, petrol station orlen – pkn)	3 char
<Filled>	Is pop overfilled?	true/false (boolean)
<Suggestions xsi:nil="true" />	A list of up to 5 suggested alternative pick-up points (available in case of overflow)	(1...5)
<Available>	Is the pop available (y/n)	1 char
<ZipCode>	Pop zipcode	6 char

Suggestions structure	
<PointPwRSuggestion>	Data on suggested alternative pick-up point within orlen paczka service

PointPwRSuggestion structure		
<DestinationCode>	POPeR code	15 char
<PSD>	PSD number	6 char
<Distance>	The distance from overfilled pick-up point, expressed in kilometers	6 char

#### Example:

```

<Err>0</Err>
<ErrDes>OK</ErrDes>
<Data>
<PointPwR>
<DestinationCode>BD-106088-31-40</DestinationCode>
  <StreetName>POWSTAŃCÓW WIELKOPOLSKICH</StreetName>

```

```
<BuildingNumber>53</BuildingNumber>
<City>Bydgoszcz</City>
<District>Bydgoszcz</District>
<Latitude>53.1303</Latitude>
<Longitude>18.0389</Longitude>
<Province>Kujawsko-Pomorskie</Province>
<CashOnDelivery>>false </CashOnDelivery>
<OpeningHours>Pn-Pt:06:00-18:00, So:07:00-14:00, Nd: nieczynne</OpeningHours>
<Location>Automat paczkowy obok sklepu</Location>
<PSD>106088</PSD>
<PointType>PSD</PointType>
<Filled>>true</Filled>
<Suggestions>
  <PointPwRSuggestion>
    <DestinationCode>BD-185280-31-26</DestinationCode>
    <PSD>185280</PSD>
    <Distance>0.4815</Distance>
  </PointPwRSuggestion>
  <PointPwRSuggestion>
    <DestinationCode>BD-268086-31-01</DestinationCode>
    <PSD>268086</PSD>
    <Distance>0.5564</Distance>
  </PointPwRSuggestion>
  <PointPwRSuggestion>
    <DestinationCode>BD-785576-31-21</DestinationCode>
    <PSD>785576</PSD>
    <Distance>0.6301</Distance>
  </PointPwRSuggestion>
  <PointPwRSuggestion>
    <DestinationCode>BD-857086-31-41</DestinationCode>
    <PSD>857086</PSD>
    <Distance>0.7714</Distance>
  </PointPwRSuggestion>
  <PointPwRSuggestion>
    <DestinationCode>BD-640946-31-30</DestinationCode>
    <PSD>640946</PSD>
    <Distance>1.0583</Distance>
  </PointPwRSuggestion>
</Suggestions>
<Available>>true</Available>
<ZipCode>03-236</ZipCode>
</PointPwR>
<PointPwR>
...
</PointPwR>
...
</Data>
```

## 4.21 GIVEMEALLRUCHWITHFILLEDD1

Functionality of indicating overflow of pickup point has been disabled.

This method returns a list of all ORLEN Paczka points (APM, PKN, PSD, PPP, PPK) that meet the requirements for delivery within 24 hours (D+1), with an extended location description. ORLEN Paczka points that are temporary filled will not be excluded or hidden. In case of mentioned overflow, the method will return up to 5 alternative points within 1.5 km from filled pick-up point.

### 4.21.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char

Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>  
<PartnerKey>abcdefghijk</PartnerKey>
```

### 4.21.2 Response

Returned values:

<Err>	Error number
<ErrDes>	Error description
<Data>	List of pick-up points within ORLEN Paczka service

Data structure	
<PointPwR>	Point data

PointPwR struktura		
<DestinationCode>	POPeR code	15 char
<StreetName>	Street on which the parcel collection point (pop) is located	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	City where the pop is located	40 char
<District>	District where the pop is located	30 char
<Latitude>	Latitude of the pop	10 char
<Longitude>	Longitude of the pop	10 char
<Province>	Province	20 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	false (boolean)
<OpeningHours>	Information about pop opening hours	50 char

<Location>	Description of the location of the parcel collection point	255 char
<PSD>	PSD number	6 char
<PointType>	Type of point (own – psd, partner – psp, ppp, psf, ppk, parcel machine – apm, petrol station orlen – pkn)	3 char
<Filled>	Is pop overfilled?	True/false (boolean)
<Suggestions xsi:nil="true" />	A list of up to 5 suggested alternative pick-up points (available in case of overflow)	(1...5)
<Available>	Is the pop available (y/n)	1 char
<ZipCode>	Pop zipcode	6 char

Suggestions structure	
<PointPwRSuggestion>	Data on suggested alternative pick-up point within orlen paczka service

PointPwRSuggestion structure		
<DestinationCode>	POPeR code	15 char
<PSD>	PSD number	6 char
<Distance>	The distance from overfilled pick-up point expressed in kilometers	6 char

### Example:

```

<Err>0</Err>
<ErrDes>OK</ErrDes>
<Data>
  <PointPwR>
    <DestinationCode>BD-106088-31-40</DestinationCode>
    <StreetName>POWSTAŃCÓW WIELKOPOLSKICH</StreetName>
    <BuildingNumber>53</BuildingNumber>
    <City>Bydgoszcz</City>
    <District>Bydgoszcz</District>
    <Latitude>53.1303</Latitude>
    <Longitude>18.0389</Longitude>
    <Province>Kujawsko-Pomorskie</Province>
    <CashOnDelivery>>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:06:00-18:00, So:07:00-14:00, Nd: nieczynne</OpeningHours>
    <Location>Automat paczkowy obok sklepu</Location>
    <PSD>106088</PSD>
    <PointType>PSD</PointType>
    <Filled>>true</Filled>
    <Suggestions>
      <PointPwRSuggestion>
        <DestinationCode>BD-185280-31-26</DestinationCode>
        <PSD>185280</PSD>
        <Distance>0.4815</Distance>
      </PointPwRSuggestion>
    </Suggestions>
  </PointPwR>
</Data>

```

```

    <PointPwRSuggestion>
      <DestinationCode>BD-268086-31-01</DestinationCode>
      <PSD>268086</PSD>
      <Distance>0.5564</Distance>
    </PointPwRSuggestion>
    <PointPwRSuggestion>
      <DestinationCode>BD-785576-31-21</DestinationCode>
      <PSD>785576</PSD>
      <Distance>0.6301</Distance>
    </PointPwRSuggestion>
    <PointPwRSuggestion>
      <DestinationCode>BD-857086-31-41</DestinationCode>
      <PSD>857086</PSD>
      <Distance>0.7714</Distance>
    </PointPwRSuggestion>
    <PointPwRSuggestion>
      <DestinationCode>BD-640946-31-30</DestinationCode>
      <PSD>640946</PSD>
      <Distance>1.0583</Distance>
    </PointPwRSuggestion>
  </Suggestions>
  <Available>true</Available>
  <ZipCode>03-236</ZipCode>
</PointPwR>
<PointPwR>
...
</PointPwR>
...
</Data>

```

## 4.22 GIVEMEPSDZIPCODE

This method is inactive as of January 1, 2025 due to the closure of PSD points and the discontinuation of the COD service.

This method returns a list of ORLEN Paczka Kiosk (PSD) based on given province with an extended location description.

### 4.22.1 Request

**Request parameters:**

NONE		
------	--	--

### 4.22.2 Response

**Returned values:**

<DestinationCode>	POPeR code	15 char
-------------------	------------	---------

<StreetName>	Street on which the parcel collection point (pop) is located	50 char
<BuildingNumber>	Building number, apartment number	15 char
<City>	City where the pop is located	40 char
<District>	District where the pop is located	30 char
<Latitude>	Latitude of the pop	10 char
<Longitude>	Longitude of the pop	10 char
<Province>	Province	20 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	(false) boolean
<OpeningHours>	Information about pop opening hours	50 char
<Location>	Description of the location of the parcel collection point	255 char
<PSD>	PSD number	6 char
<Available>	Is the pop available? (y/n)	1 char
<Zipcode>	Zipcode where the pop is located	6 char

#### Example:

```

<DestinationCode>KL-895926-J1-62</DestinationCode>
<StreetName>MARSZAŁKOWSKA</StreetName>
<BuildingNumber>82</BuildingNumber>
<City>Warszawa</City>
<District>Śródmieście</District>
<Longitude>21.0137</Longitude>
<Latitude>52.2274</Latitude>
<Province>Mazowieckie</Province>
<CashOnDelivery>>false</CashOnDelivery>
<OpeningHours>Pn-Pt:06:00-19:00, So:06:00-15:00, Nd: nieczynne</OpeningHours>
<Location>Automat paczkowy obok sklepu</Location>
<PSD>895926</PSD>
<Available>T</Available>
<Zipcode> 00-517</Zipcode>

```

### 4.23 GIVEMEPACKSTATUS

This method provides the current status of the parcel in the ORLEN Paczka service.

The GiveMePackStatus method returns tracking statuses no older than 90 days. Statuses added to the shipment more than 90 days before the API query date will not be returned.

This method returns the latest status of the package specified in the API request. It does not return statuses for packages related to this package.

Time returned in <Data> tag is in the UTC+1 time zone (local Polish time). Please ignore the "Z" character at the end of time format.

### 4.23.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char
<PackCode>	Pack code number of the notified parcel	13 char
<PhoneNumber>	Recipient's phone number	9 char – number without prefix +48 12 char – number with prefix +48

### 4.23.2 Response

Returned values:

<PackCode>	Pack code number of the notified parcel	13 char
<Trans>	Parcel status	4 char
Trans_Des>	Status description	40 char
<Data>	Date since when the parcel is in the status	YYYY-MM-DDThh:mm:ssTZD
<Destination>	POPeR code	15 char

Example:

```
<PackCode>2100123123123</PackCode>
<Trans>680</Trans>
<Trans_Des>W Transporcie do Kiosku</Trans_Des>
<Data>2024-12-14T04:35:10.9237746Z</Data>
<Destination>KL-895926-J1-62</Destination>
```

## 4.24 GIVEMEPACKSTATUSLIST

This method returns the current status of the parcels list in the ORLEN Paczka service.

The GiveMePackStatusList method returns tracking statuses no older than 90 days. Statuses added to the shipment more than 90 days before the API query date will not be returned.

The maximum number of packages declared in one call is 1000.

The declaration of the package for which statuses are to be returned is done in the <string> tag of the <PackCodes></PackCodes> object (the maximum number of <string> tags is 1000).

This method returns the current statuses of the packages specified in the API request. It does not return statuses for packages related to these packages.

Time returned in <Data> tag is in the UTC+1 time zone (local Polish time). Please ignore the "Z" character at the end of time format.

## 4.24.1 Request

### Request parameters:

<PackCodes>	The list of parcels	(REQUIRED) (quantity: 1 ... 1000)
<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char

<PackCodes> structure		
<string>	Pack code number	(REQUIRED)13 char

### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<PackCodes>
<string>2100123123123</string>
</PackCodes>
```

## 4.24.2 Response

### Returned values:

<PackCode>	Pack code number of the notified parcel	13 char
<Trans>	Parcel status	4 char
Trans_Des>	Status description	40 char
<Data>	Date since when the parcel is in the status	YYYY-MM-DDThh:mm:ssTZD
<Destination>	POPeR code	15 char

### Example:

```
<PackCode>2100123123123</PackCode>
<Trans>200</Trans>
<Trans_Des>Zaawizowana do PwR</Trans_Des>
<Data>2024-10-22T13:18:49.9237746Z</Data>
<Destination>WS-324889-U6-02</Destination>
```

## 4.25 GIVEMEPACKSTATUSFULLHISTORY

This method returns full history of parcel.

The GiveMePackStatusFullHistory method returns tracking statuses no older than 90 days. Statuses added to the shipment more than 90 days before the API query date will not be returned.

This method returns the full status history of the package specified in the API request. It also returns statuses for related operational return shipments, but does not return statuses for related consumer return shipments.

Time returned in <Data> tag is in the UTC+1 time zone (local Polish time). Please ignore the "Z" character at the end of time format.

#### 4.25.1 Request

**Request parameters:**

<PackCode>	Pack code number of the notified parcel	(REQUIRED)13 char
<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char

**Example with minimum fields required:**

<PackCode>2100123123123</PackCode>

<PartnerID>1234567890</PartnerID>

<PartnerKey>abcdefghijkl</PartnerKey>

#### 4.25.2 Response

**Returned values:**

PackCode>	Pack code number of the notified parcel	13 char
<Trans>	Parcel status	4 char
<Trans_Des>	Status description	40 char
<Data>	Date since when the parcel is in the status	YYYY-MM-DDThh:mm:ssTZD
<Destination>	POPeR code	15 char
<Cl_From>	Number of central sorting department where notification of the parcel took place	4 char
<Cl_From_Des>	Central sorting department where notification of the parcel took place	9 char
<Cl>	Central sorting department where the parcel must be delivered	9 char
<Cl_Des>	Number of central sorting department where the parcel must be delivered	4 char
<Attribute>	Parcel attribute (MOBILE – mobile point, ALTERNATIVE – alternative point, CHANGED – changed point, RETURN – parcel return to the shipping pick-up point, 2_RETURN – parcel return to the Central sorting department, ZWROT – return pack to the sender's warehouse)	12 char
<StreetName>	Street where the parcel collection point is located	50 char
<City>	City where the pop is located	40 char
<OpeningHours>	Information about pop opening hours	50 char

<Location>	Description of the location of the parcel collection point	255 char
------------	--	----------

#### Example:

```

<PackCode>2100123123123</PackCode>
<Trans>200</Trans>
<Trans_Des>Zaawizowana do PwR</Trans_Des>
<Data>2024-10-22T13:18:49.8306261Z</Data>
<Destination>WS-324889-U6-02</Destination>
<CI_From>0000</CI_From>
<CI_From_Des>SC</CI_From_Des>
<CI>0130</CI>
<CI_Des>WARSZAWA</CI_Des>
<Attribute />
<StreetName>NOWY ŚWIAT 6/12</StreetName>
<City>Warszawa</City>
<OpeningHours>Pn-Pt:06:15-16:00, So: nieczynne, Nd: nieczynne</OpeningHours>
<Location>Punkt odbioru paczek znajduje się w budynku Stacji Paliw ORLEN</Location>

```

## 4.26 GIVEMEPACKSTATUSFULLHISTORYLIST

This method returns the full status history of the parcels in the ORLEN Paczka service for a list of parcel numbers.

The GiveMePackStatusFullHistoryList method returns tracking statuses no older than 90 days. Statuses added to the shipment more than 90 days before the API query date will not be returned. The maximum number of packages declared in one call is 1000.

The declaration of the package for which statuses are to be returned is done in the <string> tag of the <PackCodes></PackCodes> object (the maximum number of <string> tags is 1000).

This method returns the full status history of the package specified in the API request. It does not return statuses for packages related to this package.

Time returned in <Data> tag is in the UTC+1 time zone (local Polish time). Please ignore the "Z" character at the end of time format.

### 4.26.1 Request

#### Request parameters:

<PackCodes>	List of parcel ean-13 number	(REQUIRED) (quantity: 1...1000)
<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char

#### Example with minimum fields required:

```

<PackCodes>
<string>2100123123123</string>

```

```

</PackCodes>
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijkl</PartnerKey>

```

## 4.26.2 Response

### Returned values:

<PackCode>	Pack code number of the notified parcel	13 char
<Trans>	Parcel status number	4 char
<Trans_Des>	Status description	40 char
<Data>	Date since when the parcel is in the status	YYYY-MM-DDThh:mm:ssTZD
<Destination>	POPeR code	15 char
<CI_From>	Number of central sorting department where notification of the parcel took place	4 char
<CI_From_Des>	Central sorting department where notification of the parcel took place	9 char
<CI>	Central sorting department where the parcel must be delivered	9 char
<CI_Des>	Number Of Central Sorting Department Where The Parcel Must Be Delivered	4 char
<Attribute>	Parcel attribute (MOBILE – mobile point, ALTERNATIVE – alternative point, CHANGED – changed point, RETURN – parcel return to the shipping pick-up point, 2_RETURN – parcel return to the Central sorting department, ZWROT – return pack to the sender’s warehouse)	12 char
<StreetName>	Street where the parcel collection point is located	50 char
<City>	City where the pop is located	40 char
<OpeningHours>	Information about pop opening hours	50 char
<Location>	Description of the location of the parcel collection point	255 char

### Example:

```

<PackCode>2100123123123</PackCode>
<Trans>200</Trans>
<Trans_Des> Zaawizowana do PwR </Trans_Des>

```

```

<Data>2018-01-15T12:00:00.9237746Z</Data>
<Destination>PZ-730614-04-38</Destination>
<Cl_From>0000</Cl_From>
<Cl_From_Des>SC</Cl_From_Des>
<Cl>6306</Cl>
<Cl_Des>POZNAŃ</Cl_Des>
<Attribute />
<StreetName>Jasielska 11</StreetName>
<City>Poznań</City>
<OpeningHours>Pn-Pt:06:00-16:00, So: nieczynne, Nd: nieczynne</OpeningHours>
<Location>Automat paczkowy obok sklepu</Location>

```

## 4.27 GIVEPARTNERSTATUS

This method returns information about the ORLEN Paczka customer.

### 4.27.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char

### 4.27.2 Response

Returned values:

<Err>	Error number
<ErrDes>	Error description
<PaymentType>	Payment type (POSTPAID or PREPAID)
<CashOnDelivery>	Is the cash on delivery service available? (false) (section 3.12)
<Insurance>	Is the insurance service available? (true/false)
<DeliveryRUCH>	Is the delivery service available? (true/false)
<PackPrice>	Parcel price (in grosze)
<CODProvision>	Cash on delivery commission (percentage-wise)
<CODMinPrice>	Minimum of cash on delivery commission (in grosze)
<FirstName>	Name
<LastName>	Surname
<CompanyName>	Company name
<StreetName>	Street name
<BuildingNumber>	Building number
<PostCode>	Postal code
<City>	City
<Nip>	NIP number
<PhoneNumber>	Phone number

<EMail>	E-mail address
<InsPrice100>	Commission for insurance up to 100 pln (in grosze)
<InsPrice300>	Commission for insurance up to 300 pln (in grosze)
<InsPrice500>	Commission for insurance up to 500 pln (in grosze)
<InsPrice1000>	Commission for insurance up to 1000 pln (in grosze)
<InsPrice1500>	Commission for insurance up to 1500 pln (in grosze)
<InsPrice2000>	Commission for insurance up to 2000 pln (in grosze)
<InsPrice2500>	Commission for insurance up to 2500 pln (in grosze)
<InsPrice5000>	Commission for insurance up to 5000 pln (in grosze)

### Example:

```

<Err>000</Err>
<ErrDes>saved</ErrDes>
<PaymentType>POSTPAID</PaymentType>
<CashOnDelivery>>false</CashOnDelivery>
<Insurance >>false</Insurance>
<DeliveryRUCH >>true</DeliveryRUCH>
<PackPrice>850</PackPrice>
<CODProvision>0.95</CODProvision>
<CODMinPrice>150</CODMinPrice>
<FirstName>Jan</FirstName>
<LastName>Nowak</LastName>
<CompanyName>Firma SA</CompanyName>
<StreetName>Prosta</StreetName>
<BuildingNumber>1234</BuildingNumber>
<PostCode>99-999</PostCode>
<City>Warszawa</City>
<Nip>123-456-78-90</Nip>
<PhoneNumber>123456789</PhoneNumber>
<EMail>test@tt.pl</EMail>
<InsPrice100>0</InsPrice100>
<InsPrice200>0</InsPrice200>
<InsPrice300>0</InsPrice300>
<InsPrice500>100</InsPrice500>
<InsPrice1000>100</InsPrice1000>
<InsPrice1500>100</InsPrice1500>
<InsPrice2000>200</InsPrice2000>
<InsPrice2500>200</InsPrice2500>
<InsPrice5000>400</InsPrice5000>

```

## 4.28 GIVEPAYMENTTYPE

The method is used to verify the payment type of the ORLEN Paczka customer.

### 4.28.1 Request

#### Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char

## 4.28.2 Response

### Returned values:

<Err>	Error number
<ErrDes>	Error description
<PaymentType>	Payment type (POSTPAID or PREPAID)

### Example:

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<PaymentType>POSTPAID</PaymentType>
```

## 4.29 GIVESHIPPINGCODESTATUS

This method is used to verify whether the given shipping code is already taken. Shipping code is generated via the GenerateShippingCode method (section 4.10).

### 4.29.1 Request

#### Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char
<Codes>	List of shipping codes to verify	(REQUIRED) (quantity: 1...n)

<Codes> structure		
<string>		string

#### Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<Codes>
<string>0431523</string>
<string>0256356</string>
</Codes>
```

### 4.29.2 Response

#### Returned values:

<ShippingCode>	Shipping code number
<PackCode>	Parcel number related to shipping code (if shipping code is taken)

<CodeStatus>	Shipping code status
--------------	----------------------

**Example:**

```
<ShippingCode>431512</ShippingCode>
<CodeStatus>wrong PartnerID</CodeStatus>
</GiveShippingCodeStatus>
<ShippingCode>256334</ShippingCode>
<PackCode>2100123123123</PackCode>
<CodeStatus>code used</CodeStatus>
</GiveShippingCodeStatus>
```

### 4.30 LABELPRINTDUPLICATE

This method returns label duplicate for advised parcel. Label is returned in PDF format.

#### 4.30.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char
<PackCode>	EAN-13 number of the notified parcel	(REQUIRED) 13 char

**Example with minimum fields required:**

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<PackCode>2100123123123</PackCode>
```

#### 4.30.2 Response

**Returned values:**

<Err>	Error number
	Error description 211 - "lack of routing data for this label" 212 - "no permission to this label number" 213 - "parcel with this number already sent via Ruch"
<ErrDes>	
Label	Pdf document with BASE64 encoded (if err = 0)

**Example:**

```

<Err>213</Err>
<ErrDes> Parcel with this number already sent via Ruch</ErrDes>
<Label>null</Label>

```

### 4.31 LABELPRINTDUPLICATE TWO

This method return a label duplicate for advised parcel. Label is returned in PDF, PDF10, EPL, ZPL format.

#### 4.31.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<Format>	Label format (pdf, pdf10, epl or zpl)	(REQUIRED) 5 char
<PackCode>	EAN-13 number of the notified parcel	(REQUIRED) (13 char)

Example with minimum fields required:

```

<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<PackCode>2100123123123</PackCode>
<Format>PDF</Format>

```

#### 4.31.2 Response

Returned values:

<Err>	Error number
<ErrDes>	Error description 211 - "lack of routing data for this label" 212 - "no permission to this label number" 213 - "parcel with this number already sent via Ruch"
Label	Label in PDF, PDF10, EPL, ZPL encoded BASE64 format (if err = 0)

Example:

```

<Err>213</Err>
<ErrDes> Parcel with this number already sent via Ruch</ErrDes>
<Label>null</Label>

```

## 4.32 LABELPRINTDUPLICATELIST

This method returns a label duplicate list for advised parcels.

Label is returned in PDF, PDF10, EPL format.

Maximum number of labels returned at one time is 50.

Declaration of the package for which a duplicate is to be generated is done in the <string> tag of the <PackCodeList></PackCodeList> object (the maximum number of <string> tags is 50).

### 4.32.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<Format>	Label format (PDF, PDF10 or EPL)	(REQUIRED) 3 char
<PackCodeList>	List of notified parcels	(REQUIRED) (quantity: 1 ... 50)

<PackCodeList> structure		
<PackCode>	Parcel EAN-13 number	(REQUIRED)13 char

Example with minimum fields required:

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijkl</PartnerKey>
<Format>PDF</Format>
<PackCodeList>
<string>2100123123123</string>
</PackCodeList>
```

### 4.32.2 Response

Label is returned in the <Label> tag.

Returned values:

< PackCodeList > - list of parcels (1 ... n)

<Err>	Error number
<ErrDes>	Error description 211 - "lack of routing data for this label" 212 - "no permission to this label number" 213 - "parcel with this number already sent via Ruch"

Label	Label in PDF, PDF10, EPL encoded BASE64 format (if err = 0)
-------	---

**Example:**

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<LabelData>base64</LabelData>
```

### 4.33 LABELPRINTDUPLICATELISTTWO

This method return a label duplicate list for advised parcels.

Label is returned in PDF, PDF10, EPL, ZPL format.

Maximum number of labels returned at one time is 50.

Declaration of the package for which a duplicate is to be generated is done in the <string> tag of the <PackCodeList></PackCodeList> object (the maximum number of <string> tags is 50).

#### 4.33.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<Format>	Label format (PDF, PDF10, EPL or ZPL)	(REQUIRED) 5 char
<PackCodeList>	List of notified parcels	(REQUIRED) (quantity: 1 ... 50)

<PackCodeList> structure		
<string>	Parcel EAN-13 number	(REQUIRED) 13 char

**Example with minimum fields required:**

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<Format>PDF</Format>
  <PackCodeList>
    <string>2100123123123</string>
  </PackCodeList>
```

#### 4.33.2 Response

Label is returned in the <Label> tag.

**Returned values:**

<PackCodeList> - list of parcels (1 ... n)

<Err>	Error number
<ErrDes>	Error description 211 - "lack of routing data for this label" 212 - "no permission to this label number" 213 - "parcel with this number already sent via Ruch"
Label	Label in PDF, PDF10, EPL, ZPL encoded BASE64 format (if err = 0)

**Example:**

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<LabelData>base64</LabelData>
```

**4.34 PUTCUSTOMERPACK**

Method for advising an ORLEN Paczka from a retrieved pool of numbers generated via the GenerateNumberWaybills method (Section 4.8).

Advising data can be transferred only once for a given reserved package number. When attempting to transfer advisory data for a package that already has it, an error "209 - incorrect PackCode" will be returned.

**4.34.1 Request**

**Request parameters:**

<PackCode>	Parcel EAN-13 number	(REQUIRED)13 char
<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED)10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED)10 char
<DestinationCode>	POPeR code	(REQUIRED)15 char
<AlternativeDestinationCode>	Alternative POPeR code	15 char
<ReturnDestinationCode>	POPeR code for return	15 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<PackValue>	Parcel value (in grosze)	6 char
<CashOnDelivery>	Is the cash on delivery service available? (section 3.12)	
<AmountCashOnDelivery>	Cash on delivery amount (in grosze) (section 3.12)	
<Insurance>	Is the parcel insured?	(true/false) boolean
<EMail>	Recipient's e-mail address	60 char
<FirstName>	Recipient's name	30 char
<LastName>	Recipient's surname	30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char

<BuildingNumber>	Recipient's building number	10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	30 char
<PostCode>	Recipient's postcode	6 char
<PhoneNumber>	Recipient's phone number	<b>(REQUIRED)</b> 9 char – number without prefix +48 12 char – number with prefix +48
<SenderEMail>	Sender's e-mail address	<b>(REQUIRED)</b> 60 char
<SenderFirstName>	Sender's name	<b>(REQUIRED)</b> 30 char
<SenderLastName>	Sender's surname	<b>(REQUIRED)</b> 30 char
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	Sender's street name	<b>(REQUIRED)</b> 30 char
<SenderBuildingNumber>	Sender's building number	<b>(REQUIRED)</b> 10 char
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	Sender's city	<b>(REQUIRED)</b> 30 char
<SenderPostCode>	Sender's postcode	<b>(REQUIRED)</b> 6 char
<SenderPhoneNumber>	Sender's phone number	<b>(REQUIRED)</b> 9 char – number without prefix +48 12 char – number with prefix +48
<SenderOrders>	Sender's order number	30 char
<ReturnEMail>	Return e-mail address	60 char
<ReturnFirstName>	Person's name for return	30 char
<ReturnLastName>	Person's surname for return	30 char
<ReturnCompanyName>	Company name for return	70 char
<ReturnStreetName>	Street name for return	30 char
<ReturnBuildingNumber>	Building number for return	10 char
<ReturnFlatNumber>	Flat number for return	10 char
<ReturnCity>	City for return	30 char
<ReturnPostCode>	Postcode for return	6 char
<ReturnPhoneNumber>	Phone number for return	9 char – number without prefix +48 12 char – number with prefix +48
<PrintAdress>	<b>Selection of address for printout in case of return (1 – dispatch address, 2 – return address)</b>	<b>(REQUIRED)</b> 1 char
<ReturnPack>	Return parcel (t – if return)	1 char
<TransferDescription>	Transfer title for cash on delivery parcels	70 char
<ReturnAvailable>	Ability to send a return parcel (t - if yes)	1 char
<ReturnQuantity>	Number of return parcels	1 char

TransferDescription should not contain special characters.

The fields <FirstName> and <LastName> are obligatory unless a value is given in the <CompanyName> field.

The fields <SenderFirstName> and <SenderLastName> are obligatory unless a value is given in the <SenderCompanyName> field.

The field <ReturnDestinationCode> should be filled only if a parcel is sent from a pick-up point and different pick-up point is chosen for return than the pick-up point from which the parcel was sent.

While choosing the pick-up point of APM or PKN type, field <BoxSize> is obligatory and takes values: S, M or L.

**Example with minimum fields required:**

```

<PackCode >2100123123123</PackCode>
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<DestinationCode>XX-142450-00-00</DestinationCode>
<FirstName>Jan</FirstName>
<LastName>Nowak</LastName>
<PhoneNumber>123456789</PhoneNumber>
<SenderEMail>test@tt.pl</SenderEMail>
<SenderCompanyName>Firma SA</SenderCompanyName>
<SenderStreetName>Prosta</SenderStreetName>
<SenderBuildingNumber>1234</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>99-999</SenderPostCode>
<SenderPhoneNumber>123456789</SenderPhoneNumber>
<PrintAdress>1</PrintAdress>

```

**4.34.2 Response****Returned values:**

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel ean-13 number
<DestinationCode>	POPeR code (confirmation or change)
<DestinationId>	PSD code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)

**Example:**

```

<Err>000</Err>
<ErrDes>saved</ErrDes>
<PackCode_RUCH>2100123123123</PackCode_RUCH>
<DestinationCode>XX-142450-00-00</ DestinationCode>
<DestinationId>142450</ DestinationId>
<PackPrice>745</PackPrice>
<PackPaid>>false</ PackPaid>

```

## 4.35 PUTCUSTOMERPACKCANCELED

This method is used to cancel parcels previously advised in the ORLEN Paczka service.

### 4.35.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PackCode>	EAN-13 number of the notified parcel	(REQUIRED) 13 char

### 4.35.2 Response

Returned values:

<Err>	Error number
<ErrDes>	Error description
<PackCode>	Cancelled pack numbers

Example:

```
<Err>000</Err>  
<ErrDes>saved</ErrDes>  
<PackCode>2100123123123</PackCode>
```

## 4.36 PING

This method is used to check the connection status with the API webservice.

### 4.36.1 Request

Request parameters:

NONE		
------	--	--

### 4.36.2 Response

Returned values:

<PingResult>	Connection status (true/false)
--------------	--------------------------------

## 4.37 CALLPICKUP

This method is used to order a courier to drive up to pick up packages. Arrival address is automatically taken from the data assigned to the customer's contract in the ORLEN Paczka system. In the call, declare the parameters <readyDate>, <pickupDate> specifying the hourly interval of the courier's arrival. Available courier arrival intervals are returned in the GetAvailablePickups method response (section 4.39).

When this method is called, the courier's pick-up order number is returned in the response. Courier pickup cancellation is possible by contacting your sales representative or the ORLEN Paczka helpline and providing the courier order number.

<ReadyDate>, <PickupDate> tags expect a time in UTC+1 time zone (local Polish time).

### 4.37.1 Request

**Request parameters:**

<partnerID>	Customer's ID given by ORLEN Paczka	(REQUIRED) 10 char
<partnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<packList>	List of parcel numbers for collection	(REQUIRED) (ilość 1...n)
<readyDate>	Date and time of the beginning of the readiness of the parcel to be picked up	(REQUIRED) YYYY-MM-DDThh:mm:ss
<pickupDate>	Date and time of the end of the parcel's readiness to be picked up	(REQUIRED) YYYY-MM-DDThh:mm:ss

**Example:**

```
<partnerId>1234567890</partnerId>
<partnerKey>abcdefghijkl</partnerKey>
<packList>
  <string>2100123123123</string>
  <string>2100123123124</string>
</packList>
<readyDate>2025-11-26T11:00:00</readyDate>
<pickupDate>2025-11-26T13:00:00</pickupDate>
```

### 4.37.2 Response

**Returned values:**

<Err>	Error number
<ErrDes>	Description of error
<Data>	Order code number

**Example:**

```

<Err>000</Err>
<ErrDes>Success</ErrDes>
<Data>13835004</Data>

```

## 4.38 CALLPICKUPNEW

This method is used to order a courier to drive up to pick up parcels at the specified address declared when calling the method.

In the call, declare the parameters <readyDate>, <pickupDate> specifying the hourly interval of the courier's arrival. Available courier arrival intervals are returned in the GetAvailablePickups method response (section 4.39).

When this method is called, the courier's pick-up order number is returned in the response.

Courier pickup cancellation is possible by contacting your sales representative or the ORLEN Paczka helpline and providing the courier order number.

<ReadyDate>, <PickupDate> tags expect a time in UTC+1 time zone (local Polish time).

### 4.38.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PackList>	List of parcel numbers for collection	(REQUIRED) (quantity: 1...n) List<string>
<ReadyDate>	Date and time of the beginning of the readiness of the parcel to be picked up	(REQUIRED) YYYY-MM-DDThh:mm:ss
<PickupDate>	Date and time of the end of the parcel's readiness to be picked up	(REQUIRED) YYYY-MM-DDThh:mm:ss
<PostCode>	Post code	(REQUIRED) 6 char
<City>	City	(REQUIRED) 30 char
<Street>	Street	(REQUIRED) 30 char
<BuildingNo>	Building number	10 char
<Email>	E-mail adress	(REQUIRED) 60 char
<PartnerName>	Partner name	(REQUIRED) 30 char
<PersonName>	Customer name	30 char
<PersonSurname>	Customer surname	30 char
<Telephone>	Telephone number	9 char

- if PickupDate has a default value ("0001-01-01 00:00:00"), an error is returned with code 1052 (=MaxPickupDate is required);
- if PickupDate is set to Sunday, an error 1054 is returned (=MaxPickupDate cannot be set to Sunday);

- if ReadyDate has a default value ("0001-01-01 00:00:00"), an error is returned with code 1053 (=ReadyDate value has not been set);
- if ReadyDate is specified as earlier than the current time, the current ReadyDate is set;
- if ReadyDate is greater than or equal to (later than or equal to) PickupDate, error 1055 is returned (=MaxPickupDate HOUR cannot be smaller than ReadyDate HOUR).
- if the indicated zip code does not exist or exists, but the pickup time (MaxPickupDate parameter) for the result of the GetAvailablePickups method call is 01:00:00, an error 400: Bad Request will be returned preventing a pickup order for this zip code. For such postal codes, driveways are not implemented in the ORLEN Parcel service.

**Example:**

```

<PartnerId>1234567890</PartnerId>
<PartnerKey>abcdefghijk</PartnerKey>
<PackList>
  <string>2100000000001</string>
  <string>2100000000002</string>
</PackList>
<ReadyDate>2024-10-28T11:00:00</ReadyDate>
<PickupDate>2024-10-28T13:00:00</PickupDate>
<PostCode>03-236</PostCode>
<City>Warszawa</City>
<Street>Annopol</Street>
<BuildingNo>17A</BuildingNo>
<Email>test@klient.pl</Email>
<PartnerName>Firma Testowa</PartnerName>
<PersonName>Jan</PersonName>
<PersonSurname>Testowy</PersonSurname>
<Telephone>123456789</Telephone>

```

### 4.38.2 Response

**Returned values:**

<Err>	Error number
<ErrDes>	Description of error
<Data>	Order code number

**Example:**

```

<CallPickupNewResult>
  <Err>0</Err>
  <ErrDes>Success</ErrDes>
  <Data>12345678</Data>
</CallPickupNewResult>

```

### 4.39 GETAVAILABLEPICKUPS

The method is used to retrieve a list of the nearest available driveway dates and times for the specified location.

If a non-existent postal code is indicated an error 1048 - „PickupLocation post code is invalid” is returned.

If the indicated postal code for which the courier pick-up is unavailable an error 401 - “Postal Code not available, check input” will be returned.

Time returned in <MinReadyDate>, <MaxPickupDate> tags is in the UTC+1 time zone (local Polish time).

#### 4.39.1 Request

**Request parameters:**

<PartnerID>	Customer’s id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer’s key given by ORLEN Paczka	(REQUIRED) 10 char
<PostCode>	Post code	(REQUIRED) 6 char
<City>	City	30 char
<Street>	Street	30 char
<BuildingNo>	Building number	10 char
<Email>	E-mail adress	60 char
<PartnerName>	Partner name	30 char
<PersonName>	Customer name	30 char
<PersonSurname>	Customer surname	30 char
<Telephone>	Telephone number	9 char

**Example:**

```
<PartnerId>1234567890</PartnerId>
<PartnerKey>abcdefghijk</PartnerKey>
<PostCode>03-236</PostCode>
```

#### 4.39.2 Response

**Returned values:**

<Data>	Available driveways	AvailablePickupDay[]
--------	---------------------	----------------------

<b>&lt;AvailablePickupDay&gt; structure</b>	
<Date>	Driveway day
<MinReadyDate>	Date and time of the beginning of readiness to receive the pickup
<MaxPickupDate>	Date and time of the end of readiness to receive the package
<MinimumInterval>	The minimum interval between the declared minreadydate and maxpickupdate
<Err>	Error number
<ErrDes>	Description of error

If a non-existent post code is indicated, a "PickupLocation post code is invalid" error will be returned.

### Example:

```
<GetAvailablePickupsResult>
<Err>0</Err>
<ErrDes>Success</ErrDes>
<Data>
<AvailablePickupDay>
<Date>2024-10-23</Date>
<MinReadyDate>2024-10-23T08:00:00+02:00</MinReadyDate>
<MaxPickupDate>2024-10-23T16:00:00+02:00</MaxPickupDate>
<MinimumInterval>120</MinimumInterval>
</AvailablePickupDay>
<AvailablePickupDay>
<Date>2024-10-24</Date>
<MinReadyDate>2024-10-24T08:00:00+02:00</MinReadyDate>
<MaxPickupDate>2024-10-24T16:00:00+02:00</MaxPickupDate>
<MinimumInterval>120</MinimumInterval>
</AvailablePickupDay>
<AvailablePickupDay>
<Date>2024-10-25</Date>
<MinReadyDate>2024-10-25T08:00:00+02:00</MinReadyDate>
<MaxPickupDate>2024-10-25T16:00:00+02:00</MaxPickupDate>
<MinimumInterval>120</MinimumInterval>
</AvailablePickupDay>
</Data>
</GetAvailablePickupsResult>
```

## 4.40 GIVEMESHIPPINGCODESTATUSFULLHISTORY

A method that allows to retrieve the full history of a package using the shipping code.

Time returned in <Data> tag is in the UTC+1 time zone (local Polish time). Please ignore the "Z" character at the end of time format.

### 4.40.1 Request

<ShippingCode>	Shipping code	(REQUIRED) 7 char
<partnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<partnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char

### 4.40.2 Response

#### Returned values:

<PackCode>	Parcel EAN-13 number
<Trans>	Parcel status
<Trans_Des>	Status description
<Data>	Date from when the parcel is in status

<Destination>	POPeR code
<CI_From>	The Number Of The Logistics Center In The Area Of Which The Shipment Was Shipped
<CI_From_Des>	CL, in the area of which the shipment was shipped
<CI>	CL, to the area of which the shipment was pledged
<CI_Des>	CL number, to the area of which the shipment was pledged
<Attribute>	Attribute of the shipment (MOBILNY – mobile point, ALTERNATYWNY – alternative point, ZMIENIONY – point amended, POWROT – return of the parcel to the point of dispatch, 2_POWROT – return of the package to the sending warehouse, ZWROT – return parcel to sender's warehouse)
<StreetName>	The street on which the parcel collection point is located
<City>	The city where parcel collection point is located
<OpeningHours>	Information about the opening hours of parcel collection point
<Location>	Description of the location of the parcel collection point.

### Example:

```

<PackCode>3400017462421</PackCode>
<Trans>200</Trans>
<Trans_Des>Zaawizowana do PWR</Trans_Des>
<Data>2024-01-08T09:03:00.5701769Z</Data>
<Destination>WR-561043-11-11</Destination>
<CI_From>0000</CI_From>
<CI_From_Des />
<CI>9305</CI>
<CI_Des />
<Attribute />
<StreetName>KOLBERGA 1</StreetName>
<City>Jelenia Góra</City>
<OpeningHours>Pn-Pt:07:00-17:00, So:08:00-14:00, Nd: nieczynne</OpeningHours>
<Location>Automat paczkowy obok sklepu</Location>
</PackStatus>

```

## 4.41 PUTCUSTOMERSHIPINGCODECANCELED

A method to cancel the shipping code.

### 4.41.1 Request

Request parameters:

<ShippingCode>	Shipping code	(REQUIRED) 7 char
<partnerID>	Customer's ID given by ORLEN Paczka	(REQUIRED) 10 char
<partnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char

### 4.41.2 Response

Returned values:

<Err>	error number
<ErrDes>	description of error

**Example:**

```
<Err>000</Err>
<ErrDes>Saved</ErrDes>
```

#### 4.42 GENERATECUSTOMERRETURN

Method to generate a return label associated with the original package.

To activate the consumer return service, contact the ORLEN Paczka sales representative.

To generate a return shipment, in the advisement data of the original shipment (declared in the request for the methods GenerateBusinessPack, GenerateBusinessPackTwo, GenerateLabelBusinessPack, GenerateLabelBusinessPackList, GenerateLabelBusinessPackListTwo, GenerateLabelBusinessPackTwo) should provide the <ReturnQuantity>1</ReturnQuantity> parameter. Do not use the following fields:

- <ReturnDestinationCode>
- <ReturnEMail>
- <ReturnFirstName>
- <ReturnLastName>
- <ReturnCompanyName>
- <ReturnStreetName>
- <ReturnBuildingNumber>
- <ReturnFlatNumber>
- <ReturnCity>
- <ReturnPostCode>
- <ReturnPhoneNumber>

Then:

- receiver's data will be replaced by the sender's data.

**Consumer return associated with the original shipment inherits its rights. This means that if the original shipment had insurance, the return shipment will also be covered.**

New return shipment number will be returned in response to the request. For the return shipment number, generate a label using methods that return duplicate labels (section 4.30 - 4.33).

##### 4.42.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PackCode>	Primary parcel ean-13 number	(REQUIRED) 13 char
<SenderPhoneNumber>	Sender's phone number	(REQUIRED)

		9 char – number without prefix +48 12 char – number with prefix +48
--	--	--

**Example with minimum fields required:**

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<PackCode>2100123123123</PackCode>
<SenderPhoneNumber>123123123</SenderPhoneNumber>
```

#### 4.42.2 Response

**Returned values:**

<Err>	Error number
<ErrDes>	Error description
<packCode_base>	Primary shipment number
<packCode_return>	Return shipment number
<OBSZAR>	Area that will handle the shipment
<MIKROREJON>	Mikrorejon that will handle the shipment
<SORTOWNIA>	Sorting plant that will handle the shipment
<KURIER_ZWROTY>	Courier who will handle the shipment

**Example:**

```
<Err>000</Err>
<ErrDes>saved</ErrDes>
<packCode_base>2100123123123</packCode_base>
<packCode_return>2100321321321</packCode_return>
<OBSZAR />
<MIKROREJON>WSF9</MIKROREJON>
<SORTOWNIA />
<KURIER_ZWROTY>WSF9</KURIER_ZWROTY>
```

#### 4.43 GENERATECUSTOMERRETURN SHIPPINGCODE

Method to generate a shipping code associated with the original package.

To activate the consumer return service, contact the ORLEN Paczka sales representative.

To generate a return shipment, in the advisement data of the original shipment (declared in the request for the methods GenerateBusinessPack, GenerateBusinessPackTwo, GenerateLabelBusinessPack, GenerateLabelBusinessPackList, GenerateLabelBusinessPackListTwo, GenerateLabelBusinessPackTwo) should provide the <ReturnQuantity>1</ReturnQuantity> parameter. Do not use the following fields:

- <ReturnDestinationCode>

- <ReturnEMail>
- <ReturnFirstName>
- <ReturnLastName>
- <ReturnCompanyName>
- <ReturnStreetName>
- <ReturnBuildingNumber>
- <ReturnFlatNumber>
- <ReturnCity>
- <ReturnPostCode>
- <ReturnPhoneNumber>

Then:

- receiver's data will be replaced by the sender's data.

**Consumer return associated with the original shipment inherits its rights. This means that if the original shipment had insurance, the return shipment will also be covered.**

Return shipping code will be returned in response to the request.

#### 4.43.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PackCode>	Primary parcel ean-13 number	(REQUIRED) 13 char
<SenderPhoneNumber>	Sender's phone number	(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48

**Example with minimum fields required:**

```
<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<PackCode>2100123123123</PackCode>
<SenderPhoneNumber>123123123</SenderPhoneNumber>
```

#### 4.43.2 Response

**Returned values:**

<Err>	Error number
-------	--------------

<ErrDes>	Error description
<packCode_base>	Primary shipment number
<shippingCode_return>	Return shipping code number
<OBSZAR>	Area that will handle the shipment
<MIKROREJON>	Mikrorejon that will handle the shipment
<SORTOWNIA>	Sorting plant that will handle the shipment
<KURIER_ZWROTY>	Courier who will handle the shipment

#### Example:

```

<Err>000</Err>
<ErrDes>saved</ErrDes>
<packCode_base>2100123123123</packCode_base>
<shippingCode_return>1234567</shippingCode_return>
<OBSZAR />
<MIKROREJON>WSC3</MIKROREJON>
<SORTOWNIA />
<KURIER_ZWROTY>WSC3</KURIER_ZWROTY>

```

## 4.44 GENERATESTANDARDCUSTOMERRETURN

Method to generate a return label in a consumer return service.

To activate the consumer return service, contact the ORLEN Paczka sales representative.

According to the business arrangement with sales representative, the shipment generated by this method is directed to the company's headquarters address or to the address indicated for returns.

### 4.44.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PrintLabel>	Label printing	(T – True, F – false) 1 char
<Format>	Label format (PDF, EPL, ZPL, PNG)	(REQUIRED) 3 char
<PrintType>	Choice of label for printout (1 - full address, 2 – anonymous address, 3- remain only the names of the recipient and sender)	(REQUIRED) 1 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<Insurance>	Is the parcel insured?	(T/F) 1 char
<SenderFirstName>	Sender's name	(REQUIRED) 30 char
<SenderLastName>	Sender's surname	(REQUIRED) 30 char
<SenderCompanyName>	Sender's company name	70 char

<SenderStreetName>	Sender's street name	(REQUIRED) 30 char
<SenderBuildingNumber>	Sender's building number	(REQUIRED) 10 char
<SenderFlatNumber>	Sender's flat number	10 char
<SenderCity>	Sender's city	(REQUIRED) 30 char
<SenderPostCode>	Sender's postcode	(REQUIRED) 6 char
<SenderMailAdress>	Sender's e-mail address	(REQUIRED) 30 char
<SenderPhoneNumber>	Sender's phone number	(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48
<SenderOrders>	Sender's reference data	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in section 3.5.1)	30 char
<ExternalPackageNumber>	External package number (additional information in section 3.5.1)	30 char
<PackValue>	Parcel value (in grosze)	6 char

#### Example with minimum fields required:

```

<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<PrintLabel>T</PrintLabel>
<Format>PDF</Format>
<PrintType>1</PrintType>
<BoxSize>S</BoxSize>
<SenderFirstName>ImięNadawcy</SenderFirstName>
<SenderLastName>NazwiskoNadawcy</SenderLastName>
<SenderStreetName>Przykładowa</SenderStreetName>
<SenderBuildingNumber>15</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>00-000</SenderPostCode>
<SenderMailAdress>mail@mail.pl</SenderMailAdress>
<SenderPhoneNumber>111222333</SenderPhoneNumber>

```

#### 4.4.2 Response

PDF label is returned in the <LabelData> tag.

#### Returned values:

<PackCode_RUCH>	Parcel EAN-13 number
<DEST_ODDZIAL>	Logistics center that will carry out the delivery of the shipment
<KURIER_ZWROTY>	Courier who will carry out the delivery of the shipment

<OP_ADRES>	Address to which the parcel will be returned
<OP_KODPOCZTOWY>	Postal code to which the parcel will be returned
<OP_MIASTO>	City to which the shipment will be returned
<OP_NAZWA>	Name to which the shipment will be returned
<OP_KRAJ>	Country to which the shipment will be returned

**Example:**

```

<PackCode_RUCH>2100000000230</PackCode_RUCH>
<DEST_ODDZIAL>0130</DEST_ODDZIAL>
<KURIER_ZWROTY>03</KURIER_ZWROTY>
<OP_ADRES>ULICA 6/6</OP_ADRES>
<OP_KODPOCZTOWY>12-345</OP_KODPOCZTOWY>
<OP_MIASTO>MIASTO</OP_MIASTO>
<OP_NAZWA>NAZWA</OP_NAZWA>
<OP_KRAJ>PL</OP_KRAJ>

```

#### 4.45 GENERATESTANDARDCUSTOMERRETURNSHIPPINGCODE

Method to generate a shipping code in a consumer return service.

To activate the consumer return service, contact the ORLEN Paczka sales representative.

According to the business arrangement with sales representative, the shipment generated by this method is directed to the company's headquarters address or to the address indicated for returns.

##### 4.45.1 Request

**Request parameters:**

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<PrintType>	Choice of label for printout (1 - full address, 2 – anonymous address, 3- remain only the names of the recipient and sender)	(REQUIRED) 1 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<Insurance>	Is the parcel insured?	(T/F) 1 char
<SenderFirstName>	Sender's name	(REQUIRED) 30 char
<SenderLastName>	Sender's surname	(REQUIRED) 30 char
<SenderCompanyName>	Sender's company name	70 char
<SenderStreetName>	Sender's street name	(REQUIRED) 30 char
<SenderBuildingNumber>	Sender's building number	(REQUIRED) 10 char
<SenderFlatNumber>	Sender's flat number	10 char

<SenderCity>	Sender's city	(REQUIRED) 30 char
<SenderPostCode>	Sender's postcode	(REQUIRED) 6 char
<SenderMailAdress>	Sender's e-mail address	(REQUIRED) 30 char
<SenderPhoneNumber>	Sender's phone number	(REQUIRED) 9 char – number without prefix +48 12 char – number with prefix +48
<SenderOrders>	Sender's reference data	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in section 3.5.1)	30 char
<ExternalPackageNumber>	External package number (additional information in section 3.5.1)	30 char
<PackValue>	Parcel value (in grosze)	6 char

#### Example with minimum fields required:

```

<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghijk</PartnerKey>
<PrintType>1</PrintType>
<BoxSize>S</BoxSize>
<SenderFirstName>ImięNadawcy</SenderFirstName>
<SenderLastName>NazwiskoNadawcy</SenderLastName>
<SenderStreetName>Przykładowa</SenderStreetName>
<SenderBuildingNumber>15</SenderBuildingNumber>
<SenderCity>Warszawa</SenderCity>
<SenderPostCode>00-000</SenderPostCode>
<SenderMailAdress>mail@mail.pl</SenderMailAdress>
<SenderPhoneNumber>111222333</SenderPhoneNumber>

```

#### 4.45.2 Response

Shipping code in text format is returned in the <ShippingCode> tag.

#### Returned values:

<ShippingCode>	Shipping code
<DEST_ODDZIAL>	Logistics center that will carry out the delivery of the shipment
<KURIER_ZWROTY>	Courier who will carry out the delivery of the shipment
<OP_ADRES>	Address to which the parcel will be returned
<OP_KODPOCZTOWY>	Postal code to which the parcel will be returned
<OP_MIASTO>	City to which the shipment will be returned
<OP_NAZWA>	Name to which the shipment will be returned
<OP_KRAJ>	Country to which the shipment will be returned

### Example:

```
<ShippingCode>1234567</ShippingCode>
<DEST_ODDZIAL>0130</DEST_ODDZIAL>
<KURIER_ZWROTY>03</KURIER_ZWROTY>
<OP_ADRES>ULICA 6/6</OP_ADRES>
<OP_KODPOCZTOWY>12-345</OP_KODPOCZTOWY>
<OP_MIASTO>MIASTO</OP_MIASTO>
<OP_NAZWA>NAZWA</OP_NAZWA>
<OP_KRAJ>PL</OP_KRAJ>
```

## 4.46 ORLENPACZKAWEEKENDOPENPUDOPPOINTS

Method returns data for pickup points that are open on Saturdays and/or Sundays within ORLEN Paczka service.

Method does not return APM (automated parcel machine) locations, as they are open 24/7.

If the opening hours returned in the tags <OpenHoursSaStart>, <OpenHoursSaEnd> are equal to 00:00:00, this means that the point is closed on Saturday. In this case, the information returned in the tag 'IsSaturdayPickupAvailable' = false.

If the opening hours returned in the tags <OpenHoursSuStart>, <OpenHoursSuEnd> are equal to 00:00:00, this means that the point is closed on Sunday. In this case, the information returned in the tag 'IsSundayPickupAvailable' = false.

Pickup point may be available on Saturday or on Sunday or on both Saturday and Sunday.

### 4.46.1 Request

#### Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char

### 4.46.2 Response

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<Data>	List of points within ORLEN Paczka Service

<Data> structure	
<WeekendOpenPUDOPointsDto>	Point details within ORLEN Paczka service

<b>&lt;WeekendOpenPUDOPointsDto&gt; structure</b>		
<PSD>	Unique 6-digit ID number of pickup point	6 char
<PointType>	Point type (Partner – PPP, PPK, ORLEN petrol station – PKN)	3 char
<Address>	Street, Building number, Apartment number	50 char
<PostCode>	Postal code of pickup point	6 char
<City>	City of pickup point	30 char
<Longitude>	Longitude of pickup point	16 char
<Latitude>	Latitude of pickup point	16 char
<Province>	Province of pickup point	25 char
<PhysicalName>	Proper name of pickup point	70 char
<PhysicalLocation>	Location description of pickup point	255 char
<OpenHoursSaStart>	Information about opening hours of given point on a given day of the week – Saturday	8 char HH:mm:ss
<OpenHoursSaEnd>	Information about closing hours of given point on a given day of the week – Saturday	8 char HH:mm:ss
<OpenHoursSuStart>	Information about opening hours of given point on a given day of the week – Sunday	8 char HH:mm:ss
<OpenHoursSuEnd>	Information about closing hours of given point on a given day of the week – Sunday	8 char HH:mm:ss
<IsSaturdayPickupAvailable>	Is pickup available on Saturday	5 char true/false
<IsSundayPickupAvailable>	Is pickup available on Sunday	5 char true/false
<PointLastModifiedTimestamp>	Time of last update of pickup point data	30 char YYYY-MM-DDTHH:mm:ss.ms

### Example:

```

<WeekendOpenPUDOPointsDto>
  <PSD>925587</PSD>
  <PointType>PPP</PointType>
  <Address>Sienkiewicza 320</Address>
  <PostCode>42-583</PostCode>
  <City>Bobrowniki</City>
  <Longitude>18.9796190000000</Longitude>
  <Latitude>50.3800180000000</Latitude>
  <Province>śląskie</Province>
  <PhysicalName>Delikatesy Kubik</PhysicalName>
  <PhysicalLocation>obok przystanku Bobrownik i 1 Maja 1</PhysicalLocation>
  <OpenHoursSaStart>06:00:00</OpenHoursSaStart>
  <OpenHoursSaEnd>22:00:00</OpenHoursSaEnd>
  <OpenHoursSuStart>06:00:00</OpenHoursSuStart>
  <OpenHoursSuEnd>22:00:00</OpenHoursSuEnd>
  <IsSaturdayPickupAvailable>true</IsSaturdayPickupAvailable>
  <IsSundayPickupAvailable>true</IsSundayPickupAvailable>
  <PointLastModifiedTimestamp>2026-03-09T02:10:18.0133333</PointLastModifiedTimestamp>
</WeekendOpenPUDOPointsDto>

```

## 4.47 GENERATEBROKERRETURN2HOME

Method is used to generate a return label for the selected address. The response returns the text result of the operation and the label as Base64 data in the LabelData field.

To activate the return to address service, contact the ORLEN Paczka sales representative.

### 4.47.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<OriginalPackageNumber>	Original parcel number associated with the return	13 char
<PhoneNumber>	Recipient's phone number	(REQUIRED) 9 lub 12 char
<EMail>	Recipient's e-mail address	(REQUIRED) 30 char
<FirstName>	Recipient's name	(REQUIRED) 30 char
<LastName>	Recipient's surname	(REQUIRED) 30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	(REQUIRED) 10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	(REQUIRED) 30 char
<PostCode>	Recipient's postcode	(REQUIRED) 6 char
<SenderEMail>	Return sender's email address	(REQUIRED) 30 char
<SenderFirstName>	Return sender's name	(REQUIRED) 30 char
<SenderLastName>	Return sender's surname	(REQUIRED) 30 char
<SenderCompanyName>	Return sender's company name	70 char
<SenderStreetName>	Return sender's street name	30 char
<SenderBuildingNumber>	Return sender's building number	(REQUIRED) 10 char
<SenderFlatNumber>	Return sender's flat number	10 char
<SenderCity>	Return sender's city	(REQUIRED) 30 char
<SenderPostCode>	Return sender's postcode	(REQUIRED) 6 char
<SenderPhoneNumber>	Return sender's phone number	(REQUIRED) 9 lub 12 char
<SenderOrders>	Sender's reference data	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in section 3.5.1)	30 char
<ExternalPackageNumber>	External package number	30 char

	(additional information in section 3.5.1)	
<Format>	Label format (PDF, PDF10, EPL, ZPL)	(REQUIRED) string

#### Example with minimum fields required:

```

<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghij</PartnerKey>
<BoxSize>S</BoxSize>
<PhoneNumber>111222333</PhoneNumber>
<EMail>firmazwoty@firma.pl</EMail>
<FirstName>imie odbiorcy</FirstName>
<LastName>nazwisko odbiorcy</LastName>
<StreetName>ulica odbiorcy</StreetName>
<BuildingNumber>15</BuildingNumber>
<City>miasto odbiorcy</City>
<PostCode>00-000</PostCode>
<SenderEMail>klient@mail.pl</SenderEMail>
<SenderFirstName>Imię nadawcy</SenderFirstName>
<SenderLastName>Nazwisko nadawcy</SenderLastName>
<SenderStreetName>ulica nadawcy</SenderStreetName>
<SenderBuildingNumber>10</SenderBuildingNumber>
<SenderCity>miasto nadawcy</SenderCity>
<SenderPostCode>90-001</SenderPostCode>
<SenderPhoneNumber>111222333</SenderPhoneNumber>
<Format>PDF</Format>

```

#### 4.47.2 Response

The Base64-encoded label is returned in the <LabelData> tag.

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel EAN-13 number
<DestinationCode>	POPeR code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)
<LabelData>	PDF, PDF10, EPL, ZPL label (Base64)

#### Example:

```

<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
  <NewDataSet xmlns="">
    <GenerateBrokerReturn2Home diffgr:id="GenerateBrokerReturn2Home1"
msdata:rowOrder="0">
      <Err>000</Err>
      <ErrDes>saved</ErrDes>
      <PackCode_RUCH>3400940315351</PackCode_RUCH>
      <DestinationCode>0130</DestinationCode>
      <PackPrice>999</PackPrice>
    </GenerateBrokerReturn2Home>
  </NewDataSet>
</diffgr:diffgram>

```

```

        <PackPaid>false</PackPaid>
    </GenerateBrokerReturn2Home>
</NewDataSet>
</diffgr:diffgram>
</GenerateBrokerReturn2HomeResult>
    <LabelData>label encoded in BASE64</LabelData>

```

## 4.48 GENERATEBROKERSHIPPINGCODERETURN2HOME

Method is used to generate a return shipment code for the selected address. The response returns the shipment code.

To activate the return to address service, contact the ORLEN Paczka sales representative.

### 4.48.1 Request

Request parameters:

<PartnerID>	Customer's id given by ORLEN Paczka	(REQUIRED) 10 char
<PartnerKey>	Customer's key given by ORLEN Paczka	(REQUIRED) 10 char
<BoxSize>	Parcel size	(REQUIRED) 1 char
<OriginalPackageNumber>	Original parcel number associated with the return	13 char
<PhoneNumber>	Recipient's phone number	(REQUIRED) 9 lub 12 char
<EMail>	Recipient's e-mail address	(REQUIRED) 30 char
<FirstName>	Recipient's name	(REQUIRED) 30 char
<LastName>	Recipient's surname	(REQUIRED) 30 char
<CompanyName>	Recipient's company name	70 char
<StreetName>	Recipient's street name	30 char
<BuildingNumber>	Recipient's building number	(REQUIRED) 10 char
<FlatNumber>	Recipient's flat number	10 char
<City>	Recipient's city	(REQUIRED) 30 char
<PostCode>	Recipient's postcode	(REQUIRED) 6 char
<SenderEMail>	Return sender's email address	(REQUIRED) 30 char
<SenderFirstName>	Return sender's name	(REQUIRED) 30 char
<SenderLastName>	Return sender's surname	(REQUIRED) 30 char
<SenderCompanyName>	Return sender's company name	70 char
<SenderStreetName>	Return sender's street name	30 char
<SenderBuildingNumber>	Return sender's building number	(REQUIRED) 10 char
<SenderFlatNumber>	Return sender's flat number	10 char
<SenderCity>	Return sender's city	(REQUIRED) 30 char
<SenderPostCode>	Return sender's postcode	(REQUIRED) 6 char

<SenderPhoneNumber>	Return sender's phone number	(REQUIRED) 9 lub 12 char
<SenderOrders>	Sender's reference data	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in section 3.5.1)	30 char
<ExternalPackageNumber>	External package number (additional information in section 3.5.1)	30 char

#### Example with minimum fields required:

```

<PartnerID>1234567890</PartnerID>
<PartnerKey>abcdefghij</PartnerKey>
<BoxSize>S</BoxSize>
<PhoneNumber>111222333</PhoneNumber>
<EMail>firmazwoty@firma.pl</EMail>
<FirstName>imie odbiorcy</FirstName>
<LastName>nazwisko odbiorcy</LastName>
<StreetName>ulica odbiorcy</StreetName>
<BuildingNumber>15</BuildingNumber>
<City>miasto odbiorcy</City>
<PostCode>00-000</PostCode>
<SenderEMail>klient@mail.pl</SenderEMail>
<SenderFirstName>imie nadawcy</SenderFirstName>
<SenderLastName>nazwisko nadawcy</SenderLastName>
<SenderStreetName>ulica nadawcy</SenderStreetName>
<SenderBuildingNumber>10</SenderBuildingNumber>
<SenderCity>miasto nadawcy</SenderCity>
<SenderPostCode>90-001</SenderPostCode>
<SenderPhoneNumber>111222333</SenderPhoneNumber>

```

#### 4.48.2 Response

Shipping code / operation result is returned in the <Shipping Code> tag.

#### Returned values:

<Err>	Error number
<ErrDes>	Error description
<ShippingCode>	Shipping code
<DestinationCode>	POPeR code
<PackPrice>	Parcel price (in grosze)
<PackPaid>	Parcel payment (true/false)
<ShippingCodeExpirationDate>	Validity date of shipping code

#### Example:

```

<GenerateBrokerShippingCodeReturn2Home diffgr:id="GenerateBrokerShippingCodeReturn2Home1"
msdata:rowOrder="0">
  <Err>000</Err>
  <ErrDes>saved</ErrDes>

```

```

    <ShippingCode>1234567890</ShippingCode>
    <DestinationCode>0130</DestinationCode>
    <PackPrice>999</PackPrice>
    <PackPaid>False</PackPaid>
    <ShippingCodeExpirationDate>2026-03-23T00:00:00+01:00</ShippingCodeExpirationDate>
</GenerateBrokerShippingCodeReturn2Home>

```

## 5 ERROR MESSAGES

<b>100</b>	missing PartnerID
<b>101</b>	missing PartnerKey
<b>103</b>	missing PhoneNumber
<b>104</b>	missing DestinationCode
<b>105</b>	missing FirstName or LastName or CompanyName
<b>106</b>	missing PackCode
<b>107</b>	missing PrintAdress
<b>108</b>	missing Email
<b>111</b>	missing SenderEMail
<b>112</b>	missing SenderPhoneNumber
<b>113</b>	missing SenderCity
<b>114</b>	missing SenderStreetName
<b>115</b>	missing SenderBuildingNumber
<b>116</b>	missing SenderPostCode
<b>117</b>	missing SenderFirstName or SenderLastName or SenderCompanyName
<b>120</b>	missing AmountCashOnDelivery
<b>121</b>	missing TransferDescription
<b>123</b>	missing ReturnCity
<b>124</b>	missing ReturnStreetName
<b>125</b>	missing ReturnBuildingNumber
<b>126</b>	missing ReturnPostCode
<b>127</b>	missing ReturnCompanyName
<b>128</b>	missing ReturnAvailable
<b>129</b>	missing ReturnQuantity
<b>130</b>	illegal characters in AmountCashOnDelivery
<b>131</b>	illegal characters in TransferDescription
<b>132</b>	incorrect format
<b>133</b>	incorrect PhoneNumber
<b>134</b>	incorrect PackValue
<b>135</b>	PackValue exceeds the maximum insurance value
<b>136</b>	incorrect TransferDescription
<b>137</b>	incorrect ReturnQuantity
<b>138</b>	incorrect PostCode

139	missing PrintType
140	cannot be zero in AmountCashOnDelivery
141	incorrect BoxSize
142	incorrect SenderPhoneNumber
143	incorrect label format (PDF / EPL / ZPL / etc.)
144	incorrect apckage number
150	exceeded maximum amount of packages
161	missing City
162	missing Nip
163	missing PackCode or PhoneNumber
164	missing NumberOfWaybills
191	unsupported zipcode
201	already cancelled
202	cannot be cancelled
205	unknown PackCode
206	unknown DestinationCode
207	unknown ReturnDestinationCode
208	customer already exists
209	incorrect PackCode
211	lack of routing data for this label
212	no permission to this label number
213	parcel with this number already sent via Ruch
214	package does not belong to the partner
222	pack not found in RUCH system
223	pack does not belong to parnter
240	zbyt wiele zwrotów paczki
243	niepoprawny PackCode
250	missing Psd
251	unknown Psd
252	unknown ReturnPsd
253	unknown AlternativePsd
254	unknown AlternativeDestinationCode
300	error writing
308	no permission
309	no permission to create customer
310	no permission to COD
311	no permission to Insurance
312	no permission to Return
314	no permission to WhiteLabel
318	no service COD in DestinationCode
330	label generation error
336	label generation error
350	invalid PromotionCode

<b>399</b>	pack not found
<b>400</b>	too many packages when trying to get package numbers
<b>401</b>	incorrect PartnerID and/or PartnerKey
<b>801</b>	no parcels transferred to the shipment protocol
<b>802</b>	too many parcels handed over for shipment protocol
<b>901</b>	unspecified error
<b>999</b>	API is not ready to confirm user permissions. Try again in a while.
<b>1038</b>	PartnerName parameter is empty or NULL
<b>1039</b>	Street parameter is empty or NULL
<b>1040</b>	City parameter is empty or NULL
<b>1041</b>	PostCode parameter is empty or NULL
<b>1043</b>	Email parameter is empty or NULL
<b>1052</b>	PickupDate value has not been set
<b>1053</b>	ReadyDate value has not been set
<b>1054</b>	PickupDate value was set on Sunday
<b>1055</b>	MaxPickupDate HOUR cannot be smaller than ReadyDate HOUR
<b>1067</b>	incorrect interval's hours between MaxPickupDate and ReadyDate for this day
<b>1077</b>	MaxPickupDate is invalid. Max available value for selected date: YYYY-MM-DD HH:MM:SS
<b>1084</b>	pickup not available for selected day

## 6 WARNING MESSAGES

The presence of messages 006, 007 or 008 indicates an outdated list of ORLEN Paczka points. Downloading the current list of points will eliminate the appearance of messages. However, these are not errors that prevent the label from being generated. Despite their appearance, the label will be generated correctly and the current value defining the given point will be substituted in the DestinationCode field.

<b>000</b>	saved
<b>006</b>	saved but changed DestinationCode
<b>007</b>	saved but changed ReturnDestinationCode
<b>008</b>	saved but changed DestinationCode or ReturnDestinationCode

## 7 PARCEL STATUSES

Trans	Trans_Des	Attribute
100	in regional sorting department	NULL / 2_POWROT / POWROT / ZWROT
110	in transport to central sorting department from dispatch	NULL / 2_POWROT / POWROT
193	redirect to APM	NULL
195	redirect to pickup point	NULL
200	advised to PwR	NULL / 2_POWROT / POWROT / ZWROT
201	Advised cancelled	NULL / 2_POWROT / POWROT / ZWROT
210	sent in drop point	NULL / ZWROT
230	between dispatch point and regional sorting center	NULL / ZWROT
240	between dispatch point and regional sorting center	NULL / 2_POWROT / POWROT / ZWROT
241	in transport after storage	NULL / POWROT
300	in central sorting department	NULL / 2_POWROT / POWROT / ZWROT
400	in central sorting department	NULL / 2_POWROT / POWROT / ZWROT
450	departure from central sorting centre	NULL / 2_POWROT / POWROT
653	in dispatch	NULL / 2_POWROT / POWROT / ZWROT
660	in transit by courier to the collection point	NULL / 2_POWROT / POWROT / ZWROT
665	transferred for placement in PUDO	NULL / 2_POWROT / POWROT
677	unsuccessful delivery	NULL / 2_POWROT
679	not handed over to the courier	NULL / 2_POWROT / POWROT
680	in transport to collection point	NULL / 2_POWROT / POWROT / ZWROT
681	for storage	NULL / POWROT
690	in the pick-up point	NULL / 2_POWROT / POWROT / ZWROT
691	storage at the pickup point	NULL / POWROT
695	in the pickup point, SMS sent	NULL / 2_POWROT / POWROT
696	storage at the pickup point – SMS sent	NULL / POWROT
700	incorrect regional sorting department	NULL / 2_POWROT / POWROT
708	storage at the pickup point – not collected	NULL / POWROT

<b>709</b>	return - not collected in time	NULL / POWROT
<b>729</b>	return to regional sorting centre – incorrect drop point	NULL / 2_POWROT / POWROT
<b>739</b>	not handed over to the pickup point	NULL / 2_POWROT / POWROT
<b>749</b>	claim	NULL
<b>790</b>	return of parcel to dispatch	NULL / 2_POWROT / POWROT
<b>800</b>	return to sorting department	NULL
<b>888</b>	the package has been archived	NULL / 2_POWROT / POWROT / ZWROT
<b>900</b>	return to sender	NULL / 2_POWROT / POWROT / ZWROT
<b>999</b>	destroyed or lost	NULL / 2_POWROT / POWROT / ZWROT
<b>1000</b>	collected by customer	NULL / 2_POWROT / POWROT
<b>1100</b>	collected by customer with signature	POWROT
<b>1200</b>	collected return parcel	ZWROT
<b>1220</b>	return to the sender	2_POWROT