API Documentation

ORLEN Paczka

1 Table of Contents

1 **Table of Contents** 1
2 **Business Description** 4
2.1 **ORLEN Paczka API WSDL Addresses** 4
2.2 **Authorization** 4
2.3 **ORLEN Paczka Pre-advice Systems** 4
2.4 **Retrieving and Displaying ORLEN Paczka Pickup Point Location Data** 5
2.4.1 ORLEN Paczka Map Widget 5
2.4.2 Retrieving Pickup Point Data Through the ORLEN Paczka API 5
2.5 **Submitting Pre-advice Data and Generating Labels** 5
2.5.1 Pre-advising Parcels in ORLEN Paczka 6
2.5.2 Shipping Without a Label 7
2.5.3 Declaring ORLEN Paczka Pickup Point Identifiers 7
2.5.4 Parcel Sizes 8
2.5.5 Additional Data for Reporting 8
2.6 **Label Validity Period** 9
2.7 **Generating Courier Handover Protocols** 9
2.8 **Ordering a Courier Pickup and Checking Pickup Availability for a Specified Address** 9
2.9 **Retrieving Parcel Tracking Statuses** 10
2.9.1 Track&Trace 10
2.9.2 Universal Status Pusher (webhook) 11
2.10 **Consumer Returns** 11
2.11 **Original Parcel Number** 12
2.12 **Cash on Delivery (COD) Parcel** 12
2.13 **Shipment Insurance** 12
2.14 **Allegro SMART! Program** 13
3 **Quick Start with the ORLEN Paczka API** 14
4 **Methods** 15
4.1 **Administration** 15
4.1.1 Ping 15
4.1.2 CreateCustomer 16

4.1.3 CreateBrokerBranch 18
4.1.4 GivePartnerStatus 20
4.1.5 GivePartnerStatusWithAllData 22
4.1.6 GivePaymentType 24
4.1.7 GiveShippingCodeStatus 25
4.2 Pickup Point Search 26
4.2.1 GiveMeAllLocationWithAllDataWithZipCode 26
4.2.2 ORLENPaczkaWeekendOpenPUDOPoints 28
4.2.3 GiveMeAllRUCHWithFilled 30
4.2.4 GiveMeAllRUCHWithFilledD1 32
4.3 Pre-advice and Label Generation 34
4.3.1 GenerateLabelBusinessPackListTwo 34
4.3.2 GenerateBusinessPackTwo 40
4.3.3 GenerateBusinessPack 44
4.3.4 GenerateShippingCode 48
4.3.5 LabelPrintDuplicateListTwo 52
4.3.6 GenerateNumberWaybills 54
4.3.7 PutCustomerPack 55
4.3.8 PutCustomerPackCanceled 59
4.3.9 PutCustomerShippingCodeCanceled 60
4.4 Goods Handover Protocol 61
4.4.1 GenerateProtocol 61
4.5 Courier Vehicle Availability Check 63
4.5.1 GetAvailablePickups 63
4.6 Ordering a Courier Pickup 65
4.6.1 CallPickupNew 65
4.7 Parcel Status Tracking 67
4.7.1 GiveMePackStatusList 67
4.7.2 GiveMePackStatusFullHistoryList 69
4.7.3 GiveMeShippingCodeStatusFullHistory 72
4.7.4 GiveMePackStatus 74
4.7.5 GiveMePackStatusFullHistory 76
4.7.6 GetGeneratedParcels 78
4.7.7 GiveCurrentPack 80
4.7.8 GiveMasterPack 81
4.8 Consumer Returns 82
4.8.1 GenerateCustomerReturn 82
4.8.2 GenerateCustomerReturnShippingCode 84
4.8.3 GenerateStandardCustomerReturn 86
4.8.4 GenerateStandardCustomerReturnShippingCode 89

4.8.5 GenerateBrokerReturn2Home ... 92
4.8.6 GenerateBrokerShippingCodeReturn2Home ... 95
4.9 Archive ... 98
4.9.1 GenerateLabelBusinessPack ... 99
4.9.2 GenerateLabelBusinessPackTwo ... 103
4.9.3 GenerateLabelBusinessPackList ... 107
4.9.4 LabelPrintDuplicate ... 111
4.9.5 LabelPrintDuplicateTwo ... 112
4.9.6 LabelPrintDuplicateList ... 113
4.9.7 CallPickup ... 115
4.9.8 GiveMeAllLocationWithAllData ... 117
4.9.9 GiveMeAllApmLocation ... 119
4.9.10 GiveMeAllRUCHLocation ... 121
4.9.11 GiveMeAllRUCHZipcode ... 123
5 Error Messages ... 125
6 Warning Messages Indicating Successful Pre-advice Generation ... 131
7 Parcel Statuses ... 132
7.1 Status Response Fields ... 132
7.2 Status Attributes ... 132
7.3 Parcel Status Table ... 132
7.4 Standard Parcel Lifecycle – Status Visualization ... 137
8 Change History ... 139

2 Business Description

2.1 ORLEN Paczka API WSDL Addresses

Environment	WSDL
Production	https://api.orlenpaczka.pl/WebServicePwRProd/WebServicePwR.asmx?WSDL
Test	https://api-test.orlenpaczka.pl/WebServicePwR/WebServicePwR.asmx?WSDL

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

The current WSDL for a given environment is the primary source of information about operation names, parameter names and order, and letter case. XML element names are case-sensitive and must be sent exactly as specified in the WSDL.

2.2 Authorization

Methods that require authorization are authorized by providing the customer login in the `<PartnerID>` field and the customer password in the `<PartnerKey>` field in the request.

Production environment credentials are sent automatically to the specified email address when the agreement with ORLEN Paczka is activated. If the credentials are forgotten, contact your ORLEN Paczka account manager, who will request that they be sent again.

Test environment credentials are different from production credentials. To obtain them, contact your ORLEN Paczka account manager.

The `<PartnerKey>` in the ORLEN Paczka API and the password for the WebTrucker application are the same.

Changing the password in WebTrucker also changes the `<PartnerKey>` used for authentication in the ORLEN Paczka API.

2.3 ORLEN Paczka Pre-advice Systems

Shipments created through the ORLEN Paczka API are not visible in WebTrucker, and vice versa. This is because the two services use separate IT systems and different shipment numbering schemes.

Parcels generated using ORLEN Paczka API methods have numbers starting with “21”. Parcels with a 10-digit shipping code have numbers starting with “23”, while shipments created in WebTrucker have numbers starting with “26”.

The history of shipments created through the API must be stored by the integration using the ORLEN Paczka API. WebTrucker stores only the history of shipments created through WebTrucker.

2.4 Retrieving and Displaying ORLEN Paczka Pickup Point Location Data

Pickup point selection in the ORLEN Paczka network can be implemented in two ways.

2.4.1 ORLEN Paczka Map Widget

Use the map widget provided by ORLEN Paczka.

Documentation is available at https://www.orlenpaczka.pl/integracja-z-orlen-paczka, in the "ORLEN Paczka map widget" section.

An individual access token is generated for the ORLEN Paczka map widget. To obtain it, contact your ORLEN Paczka account manager and provide the following information:

- the online store address where the map will be embedded,
- the customer number (starting with "PWR"),
- information that the widget will be tested in a local environment (localhost), if applicable – in this case, a separate test token is generated.

The token for the local environment is temporary and expires 3 months after it is generated.

2.4.2 Retrieving Pickup Point Data Through the ORLEN Paczka API

The second option is to build your own map that displays ORLEN Paczka pickup points.

For this purpose, we recommend the `GiveMeAllLocationWithAllDataWithZipCode` method, which retrieves a list of all point types with extended descriptions and postal codes.

The pickup point database is updated once a day in the morning. To obtain current data, we recommend retrieving the point list once a day after 6:00.

Other methods for retrieving the pickup point list are described in **section 4.2**.

Types of points available in ORLEN Paczka, returned in the `<PointType>` parameter:

PointType	Definition
APM	Parcel lockers
PPP	Staffed pickup points
PKN	ORLEN fuel stations
PPK	PointPack partner pickup points (points support shipment collection only)

2.5 Submitting Pre-advice Data and Generating Labels

The process for submitting shipment pre-advice data and generating the corresponding shipping labels is described below. This is a core part of ORLEN Paczka API integration and can be implemented in several ways: by submitting the data and generating the label in a single call, by splitting these actions into separate steps, or by reserving parcel numbers and submitting the pre-advice data later.

2.5.1 Pre-advising Parcels in ORLEN Paczka

ORLEN Paczka provides three methods for submitting pre-advice data and generating labels.

SOLUTION 1

We recommend using the `GenerateLabelBusinessPackListTwo` method to pre-advise a parcel number and generate a label. This method can pre-advise a single parcel or a list of parcels and returns a combined label in PDF, PDF10, EPL, ZPL, or ZPL_300DPI format.

Other methods for submitting pre-advice data and generating labels are described in **section 4.3**.

SOLUTION 2

Pre-advise the parcel number using one method, then generate the label using an additional method.

STEP 1: Call the `GenerateBusinessPackTwo` method to submit pre-advice data to ORLEN Paczka systems. This method does not return a label in the response – it returns a parcel number for which a label must be generated.

STEP 2: To generate a label for the parcel pre-advised in step 1, we recommend using the `LabelPrintDuplicateListTwo` method (**section 4.3.5**).

Other methods for retrieving labels are described in **section 4.3**.

SOLUTION 3

Submit pre-advice data and generate a label for a reserved pool of shipment numbers.

This scenario can be used only with approval from an ORLEN Paczka account manager.

STEP 1: Use the `GenerateNumberWaybills` method (**section 4.3.6**). Parcel numbers reserved this way do not expire. At this stage, the parcel has not yet been pre-advised in ORLEN Paczka systems, which means it has no tracking status and is not visible on the parcel tracking page (www.orlenpaczka.pl/sledz-paczke).

STEP 2: To assign data to the parcel number reserved in step 1, use the `PutCustomerPack` method (**section 4.3.7**). If the parcel is not handed over to the ORLEN Paczka network within 14 calendar days from pre-advice, it will be canceled automatically.

STEP 3: To generate and retrieve a label for the parcel pre-advised in step 2, use the `LabelPrintDuplicateListTwo` method (**section 4.3.5**) or another method described in **section 4.3**.

The `<PackPrice>` field returned by pre-advice methods is directly related to the declared parcel size. Its value is not the final courier service charge – the final price is agreed individually with the ORLEN Paczka account manager before the commercial agreement is signed.

The standard `<PackPrice>` values for each parcel size, applicable to all ORLEN Paczka customers, are shown below.

EXAMPLE

```
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> field returned by pre-advice methods indicates the customer account type:

- `<PackPaid>true</PackPaid>` – means that the commercial agreement has been signed and has POSTPAID status in ORLEN Paczka systems,
- `<PackPaid>false</PackPaid>` – means that the commercial agreement has not been signed and has PREPAID status in ORLEN Paczka systems, which prevents the parcel from being sent.

2.5.2 Shipping Without a Label

Parcels can be pre-advised without a label and a shipping code can be obtained using the `GenerateShippingCode` method. The shipping code is returned in the `<ShippingCode>` field.

To enable generation of label-free shipments, contact your ORLEN Paczka account manager.

Details of the `GenerateShippingCode` method are described in **section 4.3.4**.

The shipping code has 10 digits. It is created from the parcel number by removing the two-digit “23” prefix and the final check digit.

EXAMPLE

2325783830155 → ~~23~~25783830155 → 2578383015

Parcel numbers sent using a shipping code start with “23”.

2.5.3 Declaring ORLEN Paczka Pickup Point Identifiers

In pre-advice methods, the `<DestinationCode>` field must contain the value returned in the `<DestinationCode>` field by the `GiveMeAllLocationWithAllDataWithZipCode` method or another method that returns pickup point identifiers (**section 4.2**).

When submitting pre-advice data, you must declare the full or universal pickup point identifier. This identifier is returned in the `<DestinationCode>` parameter by the methods described in **section 4.2**.

EXAMPLE

```
<DestinationCode>
      WS-116263-C3-01
</DestinationCode>
```

The universal identifier is created by replacing the prefix with XX and the suffix with 00-00.

EXAMPLE

```
XX-116263-00-00
```

Providing the universal identifier generates the shipment number and shipping label correctly – even if the system returns the warning “Saved, but DestinationCode was changed” (**section 6**).

2.5.4 Parcel Sizes

When submitting pre-advice data to ORLEN Paczka systems, the `<BoxSize>` field must be set to **S**, **M**, or **L**. This field specifies the parcel size. If the field is left empty, size M is assigned to the parcel automatically.

Parcel size specification for the ORLEN Paczka service:

Size	Dimensions	Maximum weight
S	8 x 38 x 60 cm	20 kg
M	19 x 38 x 60 cm	20 kg
L	41 x 38 x 60 cm	20 kg

2.5.5 Additional Data for Reporting

The `<TransferDescription>` and `<SenderOrders>` parameters allow you to provide additional reference data, such as an order number, buyer identifier, or information about the parcel contents.

The difference between these parameters is whether the data is printed on the label. Data entered in the `<TransferDescription>` field is not printed on the label, while data entered in the `<SenderOrders>` field is visible on the label.

The `<ExternalSenderPackageNumber>` and `<ExternalPackageNumber>` parameters, available in the `GenerateStandardCustomerReturn` and `GenerateStandardCustomerReturnShippingCode` methods, allow you to provide additional reference data. Data entered in these fields is not printed on the label.

At the customer's request, after consultation with an ORLEN Paczka account manager, a shipment report containing the data provided in these parameters can be prepared.

2.6 Label Validity Period

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

If the shipment is not accepted into the ORLEN Paczka network within this time, it is canceled automatically and receives the appropriate status.

2.7 Generating Courier Handover Protocols

To generate a courier shipment handover protocol, use the `GenerateProtocol` method (**section 4.4.1**). The protocol file is returned in PDF format for the specified parcel numbers pre-advised in ORLEN Paczka.

The method also returns information about invalid parcels (with a status other than “pre-advised”) and valid parcels, together with the protocol number.

2.8 Ordering a Courier Pickup and Checking Pickup Availability for a Specified Address

To order a courier pickup for parcels, follow the steps below.

STEP 1: Check the availability of declared courier time slots for the shipment sender's postal code.

To check the available declared courier time slots, use the `GetAvailablePickups` method (**section 4.5.1**). The method returns a list of the nearest available pickup dates and times for the specified postal code.

The response returns the `<MinimumInterval>` parameter, which specifies the minimum lead time required to order a courier pickup before the end of a given time slot. This value is provided in minutes.

EXAMPLE

```
<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>
```

A value of `<MinimumInterval>120</MinimumInterval>` means that the courier pickup can be ordered no later than 14:00 (16:00 - 2 hours = 14:00).

STEP 2: Request a courier pickup for the shipments.

We recommend the `CallPickupNew` method (**section 4.6.1**) for ordering a courier pickup. It allows you to specify a particular address together with pickup times retrieved from the `GetAvailablePickups` method response.

A courier pickup can also be ordered using the `CallPickup` method (**section 4.9.7**), which automatically retrieves the pickup address from the data assigned to the customer's commercial agreement in the ORLEN Paczka system and defined when the agreement was signed.

If the `GetAvailablePickups` method returned an available 8:00-16:00 time slot in step 1, the following time slots, among others, can be declared in the `CallPickupNew` method call:

EXAMPLE (8:00-10:00 – the request 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>
```

EXAMPLE (12:00-15:00 – the request 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>
```

EXAMPLE (14:00-16:00 – the request 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>
```

Examples of incorrectly declared time slots that do not meet the minimum interval requirement `<MinimumInterval>120</MinimumInterval>`:

EXAMPLE

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

EXAMPLE

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

The time slots above, provided in the API call, are suggested rather than final.

2.9 Retrieving Parcel Tracking Statuses

To retrieve shipment tracking statuses, you can use the methods described in **section 4.7**, Track&Trace, or the universal status pusher (webhook).

2.9.1 Track&Trace

To track a shipment status, use the link below, replacing `@parcelNumber` with the actual shipment number received in the response to the pre-advice request:

```
https://www.orlenpaczka.pl/sledz-paczke/?numer=@parcelNumber
```

Only shipments generated in the ORLEN Paczka production environment can be tracked – both through the API and through WebTrucker (manager.orlenpaczka.pl). Shipments

generated in the test environment are not visible on the production shipment tracking page.

2.9.2 Universal Status Pusher (webhook)

The ORLEN Paczka API provides the `WebhookRegister`, `WebhookUnregister`, and `WebhookStatus` methods, which enable automatic receipt of information about parcel status changes without the need to retrieve them periodically.

A detailed description of how to use and configure the universal pusher is available in separate documentation at https://www.orlenpaczka.pl/integracja-z-orlen-paczka/, under "Dokumentacja uniwersalnego Pushera statusów".

2.10 Consumer Returns

To use the consumer return service, select one of the scenarios below.

Consumer return linked to the original parcel – SHIPPING WITH A LABEL

The `GenerateCustomerReturn` method is used to generate returns with a label in the consumer return service. The return is linked to the original shipment. The method is described in **section 4.8.1**.

Consumer return linked to the original parcel – SHIPPING WITHOUT A LABEL

The `GenerateCustomerReturnShippingCode` method is used to generate returns without a label in the consumer return service. The return is linked to the original shipment. The method is described in **section 4.8.2**.

Consumer return not linked to the original parcel – SHIPPING WITH A LABEL

The `GenerateStandardCustomerReturn` method is used to generate returns with a label in the consumer return service. The return is not linked to the original shipment. The method is described in **section 4.8.3**.

A shipment generated using this method is sent to the company registered office address or another address specified for returns, as agreed with the ORLEN Paczka account manager.

Consumer return not linked to the original parcel – SHIPPING WITHOUT A LABEL

The `GenerateStandardCustomerReturnShippingCode` method is used to generate returns without a label in the consumer return service. The return is not linked to the original shipment. The method is described in **section 4.8.4**.

A shipment generated using this method is sent to the company registered office address or another address specified for returns, as agreed with the ORLEN Paczka account manager.

2.11 Original Parcel Number

To obtain the original parcel number from the return parcel number, use the `GiveMasterPack` method (**section 4.7.8**).

2.12 Cash on Delivery (COD) Parcel

As of January 2, 2025, the cash on delivery (COD) service has been disabled for ORLEN Paczka shipments.

The `<CashOnDelivery>` and `<AmountCashOnDelivery>` parameters must be removed from ORLEN Paczka API calls.

2.13 Shipment Insurance

As of December 4, 2024, the shipment insurance service has been disabled for ORLEN Paczka shipments.

As a result, any attempt to generate a label with the insurance option enabled will return an error message. To ensure correct label generation and compliance with the current ORLEN Paczka API specification, you must:

- remove the `<Insurance>` and `<PackValue>` parameters from ORLEN Paczka API calls, or
- set the `<Insurance>` parameter to F (false) and `<PackValue>` in grosz, e.g. PLN 32,99 = `<PackValue>3299</PackValue>`.

Under the current ORLEN Paczka terms and conditions, the insurance service has been replaced by default liability for shipments. It covers a value of up to PLN 5 000,00 and is pre-advised automatically – no additional insurance needs to be added when generating an order.

2.14 Allegro SMART! Program

The `GenerateLabelBusinessPackAllegro`, `GenerateLabelBusinessPackListAllegro`, `GenerateOrlenPaczkaLabel`, `GenerateOrlenPaczkaShippingCode`, `GenerateOrlenPaczkaShippingCodeReturn2Home`, `GenerateOrlenPaczkaReturn2Home`, `ORLENPaczkaPoints`, `ORLENPaczkaPointsUpdate`, and `ORLENPaczkaPointsLockersSchema` methods are intended exclusively for use by Allegro.

The methods above should not be used in solutions created by other customers.

A customer sending shipments under their own direct agreement – using the ORLEN Paczka API through any plugin integrating with ORLEN Paczka services – will be billed according to the ORLEN Paczka price list.

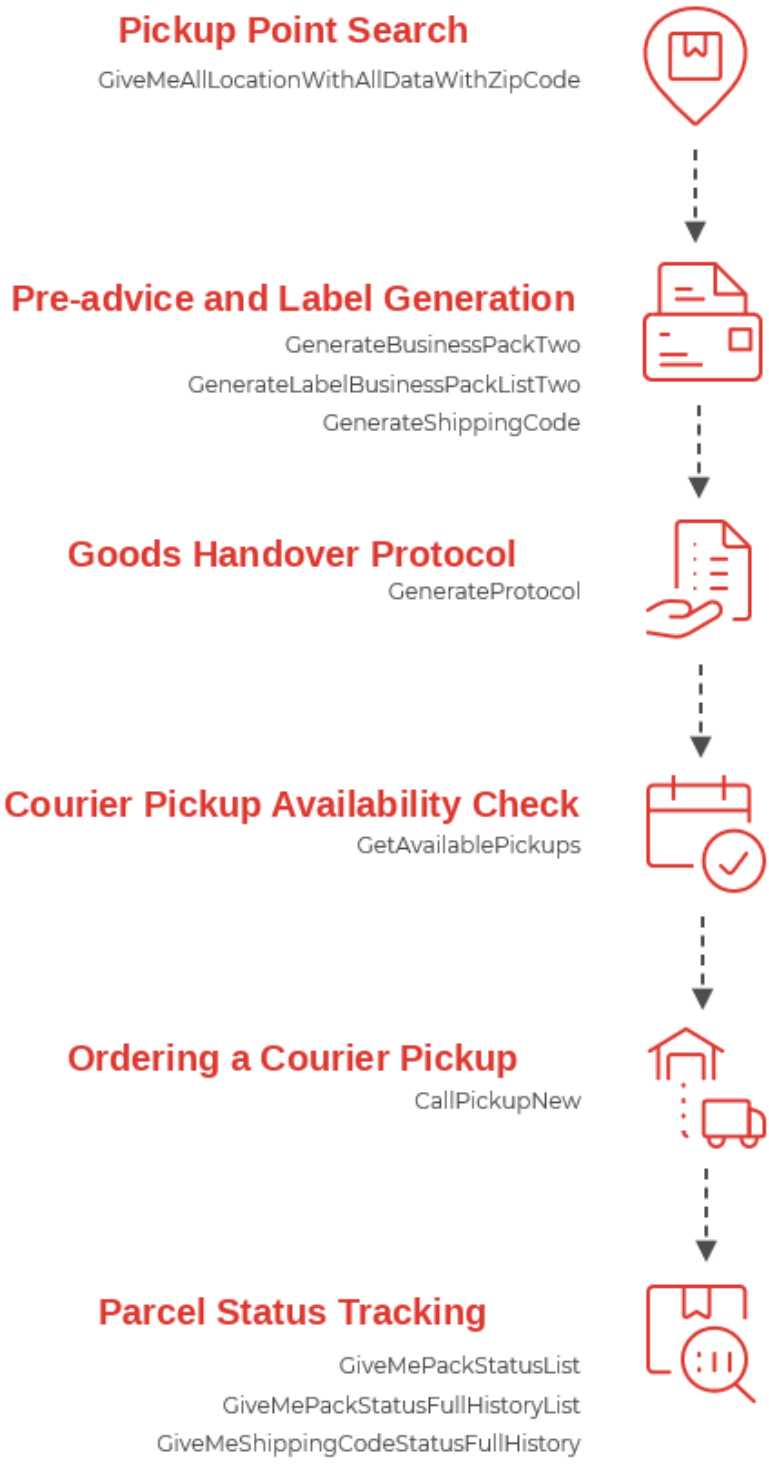
The customer's direct agreement, concluded with an ORLEN Paczka account manager, uses different billing rules than the Allegro broker agreement.

For shipments to qualify for the Allegro SMART program and be billed according to the program price list, shipping labels must be generated in the “Wysyłam z Allegro” portal or through an Allegro integration that uses the Allegro broker agreement.

3 Quick Start with the ORLEN Paczka API

The diagram covers the most commonly performed operations. For each process, it lists the methods recommended by ORLEN Paczka, making it easier to plan your integration and avoid outdated or non-recommended solutions.

A detailed description of each method, including parameters and call examples, is provided later in the documentation.



4 Methods

The request and response examples show a simplified structure of the business data. SOAP protocol elements, including `Envelope`, `Body`, `...Response` containers, and some `...Result` containers, may be omitted. The examples should not be treated as complete XML ready to send. The full structure of SOAP messages is defined in the ASMX method definitions and the WSDL. In full SOAP messages, the operation element uses the https://91.242.220.103/WebServicePwR `namespace`. In the simplified examples below, the `namespace` may be omitted. It must not be replaced with the API host address – use the value specified in the WSDL.

After a timeout or connection interruption, the result of an operation that creates a parcel or courier pickup must be treated as unknown. Automatically retrying the call may create a duplicate. Before retrying, check whether the original request was processed, if possible.

The `<Err>` field must be interpreted as a numeric code. Depending on the method, the code may be returned with or without leading zeros; for example, `0` and `000` both indicate success. The integration should not compare codes as fixed-length text.

4.1 Administration

4.1.1 Ping

Field	Value
Business process	Administration
Usage recommendation	Check the connection status with the ORLEN Paczka API
Additional information	-

4.1.1.1 Request – Ping

Field	Description	Required	Characters
-	-	-	-

REQUEST EXAMPLE – Ping

```
<Ping xmlns="https://91.242.220.103/WebServicePwR" />
```

4.1.1.2 Response – Ping

Field	Description
`<PingResult>`	Connection status (true / false)

RESPONSE EXAMPLE – Ping

```
<PingResponse>
    <PingResult>true</PingResult>
</PingResponse>
```

4.1.2 CreateCustomer

Field	Value
Business process	Administration
Usage recommendation	Creating a new customer under a direct agreement.
Additional information	Access to the method is available only with approval from an ORLEN Paczka account manager.

4.1.2.1 Request – CreateCustomer

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char
<Nip>	Customer NIP	No	10 char
<City>	City	No	30 char
<FirstName>	First name	No	30 char
<LastName>	Last name	No	30 char
<CompanyName>	Company name	No	70 char
<StreetName>	Street name	No	30 char
<BuildingNumber>	Building number	No	10 char
<PostCode>	Postal code	No	6 char
<PhoneNumber>	Phone number	**Yes**	9 char / 12 char
<EMail>	Email address	No	60 char

REQUEST EXAMPLE – CreateCustomer

```
<CreateCustomer>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <Nip>1234567890</Nip>
    <City>Warszawa</City>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <PostCode>03-236</PostCode>
    <PhoneNumber>111222333</PhoneNumber>
    <EMail>respons@orlenpaczka.pl</EMail>
</CreateCustomer>
```

4.1.2.2 Response – CreateCustomer

Field	Description
<Err>	Error number
<ErrDes>	Error description
<NewCustomerPartnerID>	Identifier of the created customer
<NewCustomerPartnerKey>	Password of the created customer

RESPONSE EXAMPLE – CreateCustomer

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

4.1.3 CreateBrokerBranch

Field	Value
Business process	Administration
Usage recommendation	Creating a new customer under a direct agreement.
Additional information	Access to the method is available only with approval from an ORLEN Paczka account manager. Fields starting with `<Return>` specify the address used for operational returns (e.g. when the recipient does not collect the parcel) and for consumer returns (withdrawal from a distance contract).

4.1.3.1 Request – CreateBrokerBranch

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<Nip>`	Customer NIP	**Yes**	13 char
`<City>`	City	**Yes**	30 char
`<FirstName>`	First name	No	30 char
`<LastName>`	Last name	No	30 char
`<CompanyName>`	Company name	No	70 char
`<StreetName>`	Street name	No	30 char
`<BuildingNumber>`	Building number	No	10 char
`<PostCode>`	Postal code	No	10 char
`<PhoneNumber>`	Phone number	**Yes**	16 char
`<EMail>`	Email address	No	60 char
`<ReturnName>`	Company name for returns	No	70 char
`<ReturnAddress>`	Return address	No	50 char
`<ReturnCity>`	City for returns	No	30 char
`<ReturnPostCode>`	Postal code for returns	No	10 char
`<ReturnCountry>`	Country for returns	No	2 char

REQUEST EXAMPLE – CreateBrokerBranch

```
<CreateBrokerBranch>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <Nip>5243023989</Nip>
    <City>Warszawa</City>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17A</BuildingNumber>
    <PostCode>03-236</PostCode>
    <PhoneNumber>111222333</PhoneNumber>
```

REQUEST EXAMPLE – CreateBrokerBranch

```
    <EMail>branch@orlenpaczka.pl</EMail>
    <ReturnName>Test</ReturnName>
    <ReturnAddress>Testowa 1</ReturnAddress>
    <ReturnCity>Warszawa</ReturnCity>
    <ReturnPostCode>00-001</ReturnPostCode>
    <ReturnCountry>PL</ReturnCountry>
</CreateBrokerBranch>
```

4.1.3.2 Response – CreateBrokerBranch

Field	Description
<Err>	Error number
<ErrDes>	Error description
<NewCustomerPartnerID>	Identifier of the created customer
<NewCustomerPartnerKey>	Password of the created customer

RESPONSE EXAMPLE – CreateBrokerBranch
Positive path – new customer created

```
<CreateBrokerBranchResponse>
    <CreateBrokerBranchResult>
        <Err>0</Err>
        <ErrDes />
        <Data>
            <NewCustomerPartnerID>XXXXXXXXXX</NewCustomerPartnerID>
            <NewCustomerPartnerKey>YYYYYYYYYY</NewCustomerPartnerKey>
        </Data>
    </CreateBrokerBranchResult>
</CreateBrokerBranchResponse>
```

RESPONSE EXAMPLE – CreateBrokerBranch
Negative path – data validation and no new customer created

```
<CreateBrokerBranchResponse>
    <CreateBrokerBranchResult>
        <Err>162</Err>
        <ErrDes>missing Nip</ErrDes>
    </CreateBrokerBranchResult>
</CreateBrokerBranchResponse>
```

4.1.4 GivePartnerStatus

Field	Value
Business process	Administration
Usage recommendation	Retrieving ORLEN Paczka customer information
Additional information	-

4.1.4.1 Request – GivePartnerStatus

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – GivePartnerStatus

```
<GivePartnerStatus>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GivePartnerStatus>
```

4.1.4.2 Response – GivePartnerStatus

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PaymentType>	Payment type (POSTPAID or PREPAID)
<DeliveryRUCH>	Parcel pickup (true / false)
<PackPrice>	Estimated price per parcel (in grosz)
<FirstName>	First name
<LastName>	Last name
<CompanyName>	Company name
<StreetName>	Street name
<BuildingNumber>	Building and unit number
<PostCode>	Postal code
<City>	City
<Nip>	NIP number
<PhoneNumber>	Phone number
<EMail>	Email address
<CashOnDelivery>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**
<Insurance>	
<CODMinPrice>	
<CODProvision>	
<InsPrice100>	
<InsPrice200>	

Field	Description
<InsPrice300>	
<InsPrice500>	
<InsPrice1000>	
<InsPrice1500>	
<InsPrice2000>	
<InsPrice2500>	
<InsPrice5000>	

RESPONSE EXAMPLE – GivePartnerStatus

```
<GivePartnerStatus>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PaymentType>POSTPAID</PaymentType>
    <CashOnDelivery>true</CashOnDelivery>
    <Insurance>false</Insurance>
    <DeliveryRUCH>false</DeliveryRUCH>
    <PackPrice>1099</PackPrice>
    <CODProvision>0</CODProvision>
    <CODMinPrice>0</CODMinPrice>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <PostCode>03-236</PostCode>
    <City>Warszawa</City>
    <Nip>1111451096</Nip>
    <PhoneNumber>111222333</PhoneNumber>
    <EMail>test@orlenpaczka.pl</EMail>
    <InsPrice100>0</InsPrice100>
    <InsPrice200>0</InsPrice200>
    <InsPrice300>0</InsPrice300>
    <InsPrice500>0</InsPrice500>
    <InsPrice1000>0</InsPrice1000>
    <InsPrice1500>0</InsPrice1500>
    <InsPrice2000>0</InsPrice2000>
    <InsPrice2500>0</InsPrice2500>
    <InsPrice5000>0</InsPrice5000>
</GivePartnerStatus>
```

4.1.5 GivePartnerStatusWithAllData

Field	Value
Business process	Administration
Usage recommendation	Retrieving extended ORLEN Paczka customer information
Additional information	-

4.1.5.1 Request – GivePartnerStatusWithAllData

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – GivePartnerStatusWithAllData

```
<GivePartnerStatusWithAllData>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GivePartnerStatusWithAllData>
```

4.1.5.2 Response – GivePartnerStatusWithAllData

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PaymentType>	Payment type (POSTPAID or PREPAID)
<DeliveryRUCH>	Parcel pickup (true / false)
<PackPrice>	Estimated price per parcel (in grosz)
<FirstName>	First name
<LastName>	Last name
<CompanyName>	Company name
<StreetName>	Street name
<BuildingNumber>	Building and unit number
<PostCode>	Postal code
<City>	City
<Nip>	NIP number
<PhoneNumber>	Phone number
<EMail>	Email address
<OBSZAR>	Logistics area
<MIKROREJON>	Logistics microregion
<SORTOWNIA>	Logistics sorting facility
<PRESORT>	Address logistics presort
<CZAS>	Address availability time parameter

Field	Description
<KURIER_ZWROTY>	Return courier
<CashOnDelivery>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**
<Insurance>	
<CODMinPrice>	
<CODProvision>	
<InsPrice100>	
<InsPrice200>	
<InsPrice300>	
<InsPrice500>	
<InsPrice1000>	
<InsPrice1500>	
<InsPrice2000>	
<InsPrice2500>	
<InsPrice5000>	

RESPONSE EXAMPLE – GivePartnerStatusWithAllData

```
<GivePartnerStatusWithAllData>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PaymentType>POSTPAID</PaymentType>
    <CashOnDelivery>true</CashOnDelivery>
    <Insurance>true</Insurance>
    <DeliveryRUCH>false</DeliveryRUCH>
    <PackPrice>1099</PackPrice>
    <CODProvision>0</CODProvision>
    <CODMinPrice>0</CODMinPrice>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <PostCode>03-236</PostCode>
    <City>Warszawa</City>
    <Nip>1111451096</Nip>
    <PhoneNumber>111222333</PhoneNumber>
    <EMail>test@orlenpaczka.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>
    <MIKROREJON>KLJ2</MIKROREJON>
    <SORTOWNIA></SORTOWNIA>
    <PRESORT>11</PRESORT>
    <CZAS>JJ </CZAS>
    <KURIER_ZWROTY>WSC3</KURIER_ZWROTY>
```

RESPONSE EXAMPLE – GivePartnerStatusWithAllData

```
</GivePartnerStatusWithAllData>
```

4.1.6 GivePaymentType

Field	Value
Business process	Administration
Usage recommendation	Retrieving information about the ORLEN Paczka customer payment type
Additional information	-

4.1.6.1 Request – GivePaymentType

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – GivePaymentType

```
<GivePaymentType>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GivePaymentType>
```

4.1.6.2 Response – GivePaymentType

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

RESPONSE EXAMPLE – GivePaymentType

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

4.1.7 GiveShippingCodeStatus

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Checking whether a label-free shipping code has been used
Additional information	Shipping codes should be provided in `<string>` elements of the `<Codes>` collection.

4.1.7.1 Request – GiveShippingCodeStatus

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char
<Codes>	Shipping code	**Yes**	10 char

REQUEST EXAMPLE – GiveShippingCodeStatus

```
<GiveShippingCodeStatus>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <Codes>
       <string>7916642107</string>
       <string>6093055548</string>
       <string>1057675697</string>
    </Codes>
</GiveShippingCodeStatus>
```

4.1.7.2 Response – GiveShippingCodeStatus

Field	Description
<ShippingCode>	Label-free shipping code
<PackCode>	Parcel number linked to the shipping code
<CodeStatus>	Shipping code status

RESPONSE EXAMPLE – GiveShippingCodeStatus

```
<GiveShippingCodeStatus>
    <ShippingCode>7916642107</ShippingCode>
    <PackCode>3379166421071</PackCode>
    <CodeStatus>code is cancelled</CodeStatus>
</GiveShippingCodeStatus>
<GiveShippingCodeStatus>
    <ShippingCode>6093055548</ShippingCode>
    <PackCode>3360930555481</PackCode>
    <CodeStatus>Code used</CodeStatus>
</GiveShippingCodeStatus>
<GiveShippingCodeStatus>
    <ShippingCode>1057675697</ShippingCode>
    <PackCode>3310576756974</PackCode>
    <CodeStatus>Code not used</CodeStatus>
</GiveShippingCodeStatus>
```

4.2 Pickup Point Search

4.2.1 GiveMeAllLocationWithAllDataWithZipCode

Field	Value
Business process	Pickup point search
Usage recommendation	Search for extended data about ORLEN Paczka network pickup points (APM, PPP, PKN, and PPK), including their postal codes
Additional information	-

4.2.1.1 Request – GiveMeAllLocationWithAllDataWithZipCode

Field	Description	Required	Characters
-	-	-	-

REQUEST EXAMPLE – GiveMeAllLocationWithAllDataWithZipCode

```
<GiveMeAllLocationWithAllDataWithZipCode
xmlns="https://91.242.220.103/WebServicePwR" />
```

4.2.1.2 Response – GiveMeAllLocationWithAllDataWithZipCode

Field	Description
<DestinationCode>	Pickup point code (after verification)
<StreetName>	Street name
<BuildingNumber>	Building and unit number
<City>	City
<Zipcode>	Postal code
<District>	District
<Longitude>	Longitude
<Latitude>	Latitude
<Province>	Province
<OpeningHours>	Pickup point opening hours
<Location>	Pickup point location description
<PSD>	Unique 6-digit pickup point ID
<Available>	Point availability in the service: T or N
<Obszar>	Pickup point area
<Mikrorejon>	Pickup point microregion
<Skrotnrpok>	Abbreviated pickup point ID number
<Sortownia>	Sorting facility assigned to the pickup point
<Presort>	Pickup point presort
<Czas>	Pickup point availability time parameter
<PointType>	Point type (section 2.4.2)
<NearestPoints>	Collection of up to three nearest pickup points determined relative to the base point

Field	Description
<NearestPoint>/<Name>	Name or address of the nearest pickup point
<NearestPoint>/<Distance>	Distance from the base pickup point in meters
<CashOnDelivery>	**The cash on delivery (COD) service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**

RESPONSE EXAMPLE – GiveMeAllLocationWithAllDataWithZipCode

```
<LocationWithAllData2>
    <DestinationCode>KL-895926-J2-55</DestinationCode>
    <StreetName>CHROBREGO</StreetName>
    <BuildingNumber>6</BuildingNumber>
    <City>Jędrzejów</City>
    <Zipcode>28-300</Zipcode>
    <District>Jędrzejów</District>
    <Longitude>20.2861596</Longitude>
    <Latitude>50.6426837</Latitude>
    <Province>świętokrzyskie</Province>
    <CashOnDelivery>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:00:00-24:00, So:00:00-24:00, Nd:00:00-
24:00</OpeningHours>
    <Location>Automat na terenie biuro ubezpieczeniowe</Location>
    <PSD>895926</PSD>
    <Available>T</Available>
    <Mikrorejon>KLJ2</Mikrorejon>
    <Skrotnrpok>J2-55</Skrotnrpok>
    <Presort>11</Presort>
    <Czas>JJ</Czas>
    <PointType>APM</PointType>
    <NearestPoints>
        <NearestPoint>
            <Name>Jędrzejów Przypkowskiego 41A/1</Name>
            <Distance>142</Distance>
        </NearestPoint>
        <NearestPoint>
            <Name>Jędrzejów Armii Krajowej 1A</Name>
            <Distance>506</Distance>
        </NearestPoint>
        <NearestPoint>
            <Name>Jędrzejów Reymonta 12A</Name>
            <Distance>542</Distance>
        </NearestPoint>
    </NearestPoints>
</LocationWithAllData2>
```

4.2.2 ORLENPaczkaWeekendOpenPUDOPoints

Field	Value
Business process	Pickup point search
Usage recommendation	Search for ORLEN Paczka pickup point data for points available on Saturday and/or Sunday
Additional information	The method does not return parcel locker (APM) data because parcel lockers are available 24/7. The method may return pickup points available on Saturday and Sunday or on Saturday only.

4.2.2.1 Request – ORLENPaczkaWeekendOpenPUDOPoints

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – ORLENPaczkaWeekendOpenPUDOPoints

```
<ORLENPaczkaWeekendOpenPUDOPoints>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</ORLENPaczkaWeekendOpenPUDOPoints>
```

4.2.2.2 Response – ORLENPaczkaWeekendOpenPUDOPoints

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PSD>	Pickup point code
<PointType>	Pickup point type (section 2.4.2)
<Address>	Pickup point address
<PostCode>	Pickup point postal code
<City>	Pickup point city
<Longitude>	Pickup point longitude
<Latitude>	Pickup point latitude
<Province>	Pickup point province
<PhysicalName>	Pickup point name (e.g. store name)
<PhysicalLocation>	Pickup point location description
<OpenHoursSaStart>	Pickup point opening time – Saturday
<OpenHoursSaEnd>	Pickup point closing time – Saturday
<OpenHoursSuStart>	Pickup point opening time – Sunday
<OpenHoursSuEnd>	Pickup point closing time – Sunday
<IsSaturdayPickupAvailable>	Whether the pickup point is available on Saturday (true – active; false – inactive)

Field	Description
<IsSundayPickupAvailable>	Whether the pickup point is available on Sunday (true – active; false – inactive)
<PointLastModifiedTimestamp>	Date and time of the last pickup point data update

RESPONSE EXAMPLE – ORLENPaczkaWeekendOpenPUDOPoints

```
<ORLENPaczkaWeekendOpenPUDOPoints>
    <Err>0</Err>
    <ErrDes />
    <Data>
        <WeekendOpenPUDOPointsDto>
            <PSD>738964</PSD>
            <PointType>PPP</PointType>
            <Address>Grunwaldzka 116</Address>
            <PostCode>32-540</PostCode>
            <City>Trzebinia</City>
            <Longitude>19.4217432000000</Longitude>
            <Latitude>50.1983243000000</Latitude>
            <Province>małopolskie</Province>
            <PhysicalName>Obok sklepu</PhysicalName>
            <OpenHoursSaStart>06:00:00</OpenHoursSaStart>
            <OpenHoursSaEnd>21:00:00</OpenHoursSaEnd>
            <OpenHoursSuStart>00:00:00</OpenHoursSuStart>
            <OpenHoursSuEnd>00:00:00</OpenHoursSuEnd>
            <IsSaturdayPickupAvailable>true</IsSaturdayPickupAvailable>
            <IsSundayPickupAvailable>false</IsSundayPickupAvailable>
            <PointLastModifiedTimestamp>2026-07-
17T01:55:25.03</PointLastModifiedTimestamp>
        </WeekendOpenPUDOPointsDto>
    </Data>
    </ORLENPaczkaWeekendOpenPUDOPoints>
```

4.2.3 GiveMeAllRUCHWithFilled

Field	Value
Business process	Pickup Point Search
Usage recommendation	Search for extended data about pickup points in the ORLEN Paczka network (APM, PPP, PKN, and PPK).
Additional information	The pickup point capacity indicator has been disabled. This is due to a process change – if a pickup point is full, the parcel is automatically redirected to the nearest available pickup point.

4.2.3.1 Request – GiveMeAllRUCHWithFilled

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMeAllRUCHWithFilled

```
<GiveMeAllRUCHWithFilled>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMeAllRUCHWithFilled>
```

4.2.3.2 Response – GiveMeAllRUCHWithFilled

Field	Description
<Err>	Error number
<ErrDes>	Error description
<DestinationCode>	Pickup point code
<StreetName>	Pickup point street name
<BuilingNumber>	Pickup point building number
<City>	Pickup point city
<District>	Pickup point district
<Latitude>	Pickup point latitude
<Longitude>	Pickup point longitude
<Province>	Pickup point province
<OpeningHours>	Pickup point opening hours
<Location>	Pickup point location description
<PSD>	Unique 6-digit pickup point ID
<PointType>	Point type (section 2.4.2)
<BuildingNumber>	Pickup point building number
<Available>	Point availability in the service: true or false
<ZipCode>	Pickup point postal code
<CashOnDelivery>	

Field	Description
<Filled>	**Cash on delivery (COD) and pickup point capacity display have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**
<Suggestions>	

RESPONSE EXAMPLE – GiveMeAllRUCHWithFilled

```
<GiveMeAllRUCHWithFilled>
    <DestinationCode>LB-358420-AB-82</DestinationCode>
    <StreetName>GOSPODARCZA</StreetName>
    <BuilingNumber>28</BuilingNumber>
    <City>Lublin</City>
    <District>Lublin</District>
    <Latitude>51.2458210</Latitude>
    <Longitude>22.6006520</Longitude>
    <Province>lubelskie</Province>
    <CashOnDelivery>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:00:00-24:00, So:00:00-24:00, Nd:00:00-
24:00</OpeningHours>
    <Location>Automat paczkowy na terenie osiedla Tatry</Location>
    <PSD>358420</PSD>
    <PointType>APM</PointType>
    <Filled>false</Filled>
    <BuildingNumber>28</BuildingNumber>
    <Available>true</Available>
    <ZipCode>20-213</ZipCode>
    <Suggestions />
</GiveMeAllRUCHWithFilled>
```

4.2.4 GiveMeAllRUCHWithFilledD1

Field	Value
Business process	Pickup Point Search
Usage recommendation	Search for extended data about pickup points in the ORLEN Paczka network (APM, PPP, PKN, and PPK) that meet D+1 delivery criteria.
Additional information	The pickup point capacity indicator has been disabled. This is due to a process change – if a pickup point is full, the parcel is automatically redirected to the nearest available pickup point.

4.2.4.1 Request – GiveMeAllRUCHWithFilledD1

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMeAllRUCHWithFilledD1

```
<GiveMeAllRUCHWithFilledD1>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMeAllRUCHWithFilledD1>
```

4.2.4.2 Response – GiveMeAllRUCHWithFilledD1

Field	Description
<Err>	Error number
<ErrDes>	Error description
<DestinationCode>	Pickup point code
<StreetName>	Pickup point street name
<BuilingNumber>	Pickup point building number
<City>	Pickup point city
<District>	Pickup point district
<Latitude>	Pickup point latitude
<Longitude>	Pickup point longitude
<Province>	Pickup point province
<OpeningHours>	Pickup point opening hours
<Location>	Pickup point location description
<PSD>	Unique 6-digit pickup point ID
<PointType>	Point type (**section 2.4.2**)
<BuildingNumber>	Pickup point building number
<Available>	Point availability in the service: true or false
<ZipCode>	Pickup point postal code

Field	Description
<CashOnDelivery>	**Cash on delivery (COD) and pickup point capacity display have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**
<Filled>	
<Suggestions>	

RESPONSE EXAMPLE – GiveMeAllRUCHWithFilledD1

```
<GiveMeAllRUCHWithFilledD1>
    <DestinationCode>LB-358420-AB-82</DestinationCode>
    <StreetName>GOSPODARCZA</StreetName>
    <BuilingNumber>28</BuilingNumber>
    <City>Lublin</City>
    <District>Lublin</District>
    <Latitude>51.2458210</Latitude>
    <Longitude>22.6006520</Longitude>
    <Province>lubelskie</Province>
    <CashOnDelivery>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:00:00-24:00, So:00:00-24:00, Nd:00:00-
24:00</OpeningHours>
    <Location>Automat paczkowy na terenie osiedla Tatry</Location>
    <PSD>358420</PSD>
    <PointType>APM</PointType>
    <Filled>false</Filled>
    <BuildingNumber>28</BuildingNumber>
    <Available>true</Available>
    <ZipCode>20-213</ZipCode>
    <Suggestions />
</GiveMeAllRUCHWithFilledD1>
```

4.3 Pre-advice and Label Generation

4.3.1 GenerateLabelBusinessPackListTwo

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Pre-advice of individual parcels or a list of shipments and creation of a PDF, PDF10, EPL, ZPL, or ZPL_300DPI label.
Additional information	`<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field. The maximum number of labels returned at one time is 50.

4.3.1.1 Request – GenerateLabelBusinessPackListTwo

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<AutoDestinationChange>`	Technical field	No	null
`<Format>`	Label format	**Yes**	PDF; PDF10; EPL; ZPL; ZPL_300DPI
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	No	30 char
`<LastName>`	Recipient last name	No	30 char
`<CompanyName>`	Recipient company name	No	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char

Field	Description	Required	Characters
<SenderEMail>	Sender email address	**Yes**	60 char
<SenderFirstName>	Sender first name	Conditional	30 char
<SenderLastName>	Sender last name	Conditional	30 char
<SenderCompanyName>	Sender company name	Conditional	70 char
<SenderStreetName>	Sender street name	**Yes**	30 char
<SenderBuildingNumber>	Sender building number	**Yes**	10 char
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Building number in the return address	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<ReturnPack>	Return parcel	No	T = return parcel
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<TransferDescription>	Reference number	No	70 char
<PrintType>	Label type	**Yes**	1 – full address; 2 – anonymous address; 3 – recipient and sender name only

Field	Description	Required	Characters
<ConfirmTermsOfUse>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<ConfirmMarketing>			
<CofirmMailing>			
<CashOnDelivery>			
<AmountCashOnDelivery>			
<PackValue>			
<Insurance>			

REQUEST EXAMPLE – GenerateLabelBusinessPackListTwo
Call for a single parcel

```
<GenerateLabelBusinessPackListTwo>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <AutoDestinationChange></AutoDestinationChange>
    <Format>ZPL</Format>
    <BusinessPackList>
        <BusinessPack>
            <DestinationCode>NS-358394-N4-35</DestinationCode>
            <AlternativeDestinationCode></AlternativeDestinationCode>
            <BoxSize>S</BoxSize>
            <EMail>receiver@orlenpaczka.pl</EMail>
            <FirstName>Jan</FirstName>
            <LastName>Nowak</LastName>
            <CompanyName>ORLEN Paczka</CompanyName>
            <StreetName>Annopol</StreetName>
            <BuildingNumber>17a</BuildingNumber>
            <FlatNumber></FlatNumber>
            <City>Warszawa</City>
            <PostCode>03-236</PostCode>
            <PhoneNumber>111222333</PhoneNumber>
            <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
            <SenderFirstName>Adam</SenderFirstName>
            <SenderLastName>Tester</SenderLastName>
            <SenderCompanyName>Test</SenderCompanyName>
            <SenderStreetName>Testowa</SenderStreetName>
            <SenderBuildingNumber>1a</SenderBuildingNumber>
            <SenderFlatNumber>2</SenderFlatNumber>
            <SenderCity>Warszawa</SenderCity>
            <SenderPostCode>00-001</SenderPostCode>
            <SenderPhoneNumber>444555666</SenderPhoneNumber>
            <SenderOrders></SenderOrders>
            <ReturnDestinationCode></ReturnDestinationCode>
            <ReturnEMail></ReturnEMail>
            <ReturnFirstName></ReturnFirstName>
            <ReturnLastName></ReturnLastName>
            <ReturnCompanyName></ReturnCompanyName>
            <ReturnStreetName></ReturnStreetName>
            <ReturnBuildingNumber></ReturnBuildingNumber>
            <ReturnFlatNumber></ReturnFlatNumber>
            <ReturnCity></ReturnCity>
            <ReturnPostCode></ReturnPostCode>
            <ReturnPhoneNumber></ReturnPhoneNumber>
            <ReturnPack></ReturnPack>
            <TransferDescription></TransferDescription>
```

REQUEST EXAMPLE – GenerateLabelBusinessPackListTwo
Call for a single parcel

```
            <PrintAdress>1</PrintAdress>
            <ReturnAvailable></ReturnAvailable>
            <ReturnQuantity></ReturnQuantity>
            <PrintType>1</PrintType>
        </BusinessPack>
    </BusinessPackList>
</GenerateLabelBusinessPackListTwo>
```

REQUEST EXAMPLE – GenerateLabelBusinessPackListTwo
Call for multiple parcels

```
<GenerateLabelBusinessPackListTwo>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <AutoDestinationChange></AutoDestinationChange>
    <Format>ZPL</Format>
    <BusinessPackList>
        <BusinessPack>
            <DestinationCode>NS-358394-N4-35</DestinationCode>
            <AlternativeDestinationCode></AlternativeDestinationCode>
            <BoxSize>S</BoxSize>
            <EMail>receiver@orlenpaczka.pl</EMail>
            <FirstName>Jan</FirstName>
            <LastName>Nowak</LastName>
            <CompanyName>ORLEN Paczka</CompanyName>
            <StreetName>Annopol</StreetName>
            <BuildingNumber>17a</BuildingNumber>
            <FlatNumber></FlatNumber>
            <City>Warszawa</City>
            <PostCode>03-236</PostCode>
            <PhoneNumber>111222333</PhoneNumber>
            <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
            <SenderFirstName>Adam</SenderFirstName>
            <SenderLastName>Tester</SenderLastName>
            <SenderCompanyName>Test</SenderCompanyName>
            <SenderStreetName>Testowa</SenderStreetName>
            <SenderBuildingNumber>1a</SenderBuildingNumber>
            <SenderFlatNumber>2</SenderFlatNumber>
            <SenderCity>Warszawa</SenderCity>
            <SenderPostCode>00-001</SenderPostCode>
            <SenderPhoneNumber>444555666</SenderPhoneNumber>
            <SenderOrders></SenderOrders>
            <ReturnDestinationCode></ReturnDestinationCode>
            <ReturnEMail></ReturnEMail>
            <ReturnFirstName></ReturnFirstName>
            <ReturnLastName></ReturnLastName>
            <ReturnCompanyName></ReturnCompanyName>
            <ReturnStreetName></ReturnStreetName>
            <ReturnBuildingNumber></ReturnBuildingNumber>
            <ReturnFlatNumber></ReturnFlatNumber>
            <ReturnCity></ReturnCity>
            <ReturnPostCode></ReturnPostCode>
            <ReturnPhoneNumber></ReturnPhoneNumber>
            <ReturnPack></ReturnPack>
            <TransferDescription></TransferDescription>
            <PrintAdress>1</PrintAdress>
            <ReturnAvailable></ReturnAvailable>
            <ReturnQuantity></ReturnQuantity>
```

REQUEST EXAMPLE – GenerateLabelBusinessPackListTwo
Call for multiple parcels

```
            <PrintType>1</PrintType>
        </BusinessPack>
        <BusinessPack>
            <DestinationCode>NS-358394-N4-35</DestinationCode>
            <AlternativeDestinationCode></AlternativeDestinationCode>
            <BoxSize>S</BoxSize>
            <EMail>receiver@orlenpaczka.pl</EMail>
            <FirstName>Jan</FirstName>
            <LastName>Nowak</LastName>
            <CompanyName>ORLEN Paczka</CompanyName>
            <StreetName>Annopol</StreetName>
            <BuildingNumber>17a</BuildingNumber>
            <FlatNumber></FlatNumber>
            <City>Warszawa</City>
            <PostCode>03-236</PostCode>
            <PhoneNumber>111222333</PhoneNumber>
            <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
            <SenderFirstName>Adam</SenderFirstName>
            <SenderLastName>Tester</SenderLastName>
            <SenderCompanyName>Test</SenderCompanyName>
            <SenderStreetName>Testowa</SenderStreetName>
            <SenderBuildingNumber>1a</SenderBuildingNumber>
            <SenderFlatNumber>2</SenderFlatNumber>
            <SenderCity>Warszawa</SenderCity>
            <SenderPostCode>00-001</SenderPostCode>
            <SenderPhoneNumber>444555666</SenderPhoneNumber>
            <SenderOrders></SenderOrders>
            <ReturnDestinationCode></ReturnDestinationCode>
            <ReturnEMail></ReturnEMail>
            <ReturnFirstName></ReturnFirstName>
            <ReturnLastName></ReturnLastName>
            <ReturnCompanyName></ReturnCompanyName>
            <ReturnStreetName></ReturnStreetName>
            <ReturnBuildingNumber></ReturnBuildingNumber>
            <ReturnFlatNumber></ReturnFlatNumber>
            <ReturnCity></ReturnCity>
            <ReturnPostCode></ReturnPostCode>
            <ReturnPhoneNumber></ReturnPhoneNumber>
            <ReturnPack></ReturnPack>
            <TransferDescription></TransferDescription>
            <PrintAdress>1</PrintAdress>
            <ReturnAvailable></ReturnAvailable>
            <ReturnQuantity></ReturnQuantity>
            <PrintType>1</PrintType>
        </BusinessPack>
    </BusinessPackList>
</GenerateLabelBusinessPackListTwo>
```

4.3.1.2 Response – GenerateLabelBusinessPackListTwo

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code (after verification)

Field	Description
<DestinationId>	Pickup point identifier
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)
<NameCL>	ORLEN Paczka logistics center name
<NrCL>	ORLEN Paczka logistics center number
<OriginDestinationCode>	Pickup point code (from the request)
<AutoChangeDestinationConfirm>	Technical field
<LabelData>	Base64-encoded label

RESPONSE EXAMPLE – GenerateLabelBusinessPackListTwo
Response for a single parcel

```
<GenerateLabelBusinessPackListTwo>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009509797</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <NameCL>NOWY SĄCZ</NameCL>
    <NrCL>4902</NrCL>
    <OriginDestinationCode>NS-358394-N4-35</OriginDestinationCode>
    <AutoChangeDestinationConfirm>0</AutoChangeDestinationConfirm>
</GenerateLabelBusinessPackListTwo>
<LabelData>Base64-encoded label</LabelData>
```

RESPONSE EXAMPLE – GenerateLabelBusinessPackListTwo
Response for multiple parcels

```
<GenerateLabelBusinessPackListTwo>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009509797</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <NameCL>NOWY SĄCZ</NameCL>
    <NrCL>4902</NrCL>
    <OriginDestinationCode>NS-358394-N4-35</OriginDestinationCode>
    <AutoChangeDestinationConfirm>0</AutoChangeDestinationConfirm>
</GenerateLabelBusinessPackListTwo>
<GenerateLabelBusinessPackListTwo>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009648205</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <NameCL>NOWY SĄCZ</NameCL>
    <NrCL>4902</NrCL>
    <OriginDestinationCode>NS-358394-N4-35</OriginDestinationCode>
```

RESPONSE EXAMPLE – GenerateLabelBusinessPackListTwo
Response for multiple parcels

```
    <AutoChangeDestinationConfirm>0</AutoChangeDestinationConfirm>
</GenerateLabelBusinessPackListTwo>
<LabelData>Base64-encoded label</LabelData>
```

4.3.2 GenerateBusinessPackTwo

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Shipment pre-advice with the option to generate the label using another method
Additional information	`<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<FirstName>` and `<LastName>` fields are required if no value is provided in the `<CompanyName>` field. The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field.

4.3.2.1 Request – GenerateBusinessPackTwo

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	Conditional	30 char
`<LastName>`	Recipient last name	Conditional	30 char
`<CompanyName>`	Recipient company name	Conditional	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char

Field	Description	Required	Characters
<SenderEMail>	Sender email address	**Yes**	60 char
<SenderFirstName>	Sender first name	Conditional	30 char
<SenderLastName>	Sender last name	Conditional	30 char
<SenderCompanyName>	Sender company name	Conditional	70 char
<SenderStreetName>	Sender street name	**Yes**	30 char
<SenderBuildingNumber>	Sender building number	**Yes**	10 char
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Building number in the return address	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<TransferDescription>	Reference data	No	70 char
<ReturnPack>	Return parcel	No	T = return parcel
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<AutoDestinationChange>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<ConfirmTermsOfUse>			
<ConfirmMarketing>			
<CofirmMailing>			
<CashOnDelivery>			
<AmountCashOnDelivery>			
<PackValue>			

Field	Description	Required	Characters
<Insurance>			

REQUEST EXAMPLE – GenerateBusinessPackTwo

```
<GenerateBusinessPackTwo>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PhoneNumber>111222333</PhoneNumber>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <AlternativeDestinationCode></AlternativeDestinationCode>
    <BoxSize>S</BoxSize>
    <EMail>receiver@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Joanna</SenderFirstName>
    <SenderLastName>Testerska</SenderLastName>
    <SenderCompanyName>Test</SenderCompanyName>
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
    <SenderFlatNumber>2</SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ReturnDestinationCode></ReturnDestinationCode>
    <ReturnEMail></ReturnEMail>
    <ReturnFirstName></ReturnFirstName>
    <ReturnLastName></ReturnLastName>
    <ReturnCompanyName></ReturnCompanyName>
    <ReturnStreetName></ReturnStreetName>
    <ReturnBuildingNumber></ReturnBuildingNumber>
    <ReturnFlatNumber></ReturnFlatNumber>
    <ReturnCity></ReturnCity>
    <ReturnPostCode></ReturnPostCode>
    <ReturnPhoneNumber></ReturnPhoneNumber>
    <ReturnPack></ReturnPack>
    <TransferDescription></TransferDescription>
    <PrintAdress>1</PrintAdress>
    <ReturnAvailable></ReturnAvailable>
    <ReturnQuantity></ReturnQuantity>
</GenerateBusinessPackTwo>
```

4.3.2.2 Response – GenerateBusinessPackTwo

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code (after verification)
<DestinationId>	Pickup point identifier
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)
<NameCL>	ORLEN Paczka logistics center name
<NrCL>	ORLEN Paczka logistics center number
<OriginDestinationCode>	Pickup point code (from the request)
<AutoChangeDestinationConfirm>	Technical field

RESPONSE EXAMPLE – GenerateBusinessPackTwo

```
<GenerateBusinessPackTwo>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009507670</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <NameCL>NOWY SĄCZ</NameCL>
    <NrCL>4902</NrCL>
    <OriginDestinationCode>NS-358394-N4-35</OriginDestinationCode>
    <AutoChangeDestinationConfirm>0</AutoChangeDestinationConfirm>
</GenerateBusinessPackTwo>
```

4.3.3 GenerateBusinessPack

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Shipment pre-advice with the option to generate the label using another method
Additional information	`<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<FirstName>` and `<LastName>` fields are required if no value is provided in the `<CompanyName>` field. The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field.

4.3.3.1 Request – GenerateBusinessPack

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	Conditional	30 char
`<LastName>`	Recipient last name	Conditional	30 char
`<CompanyName>`	Recipient company name	Conditional	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<SenderEMail>`	Sender email address	**Yes**	60 char
`<SenderFirstName>`	Sender first name	Conditional	30 char
`<SenderLastName>`	Sender last name	Conditional	30 char
`<SenderCompanyName>`	Sender company name	Conditional	70 char
`<SenderStreetName>`	Sender street name	**Yes**	30 char
`<SenderBuildingNumber>`	Sender building number	**Yes**	10 char

Field	Description	Required	Characters
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Building number in the return address	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<TransferDescription>	Reference data	No	70 char
<ReturnPack>	Return parcel	No	T = return parcel
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<CashOnDelivery>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<AmountCashOnDelivery>			
<PackValue>			
<Insurance>			

REQUEST EXAMPLE – GenerateBusinessPack

```
<GenerateBusinessPack>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PhoneNumber>111222333</PhoneNumber>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <AlternativeDestinationCode></AlternativeDestinationCode>
    <BoxSize>S</BoxSize>
```

REQUEST EXAMPLE – GenerateBusinessPack

```
    <EMail>receiver@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Joanna</SenderFirstName>
    <SenderLastName>Kowalska</SenderLastName>
    <SenderCompanyName>Test</SenderCompanyName>
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
    <SenderFlatNumber>2</SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ReturnDestinationCode></ReturnDestinationCode>
    <ReturnEMail></ReturnEMail>
    <ReturnFirstName></ReturnFirstName>
    <ReturnLastName></ReturnLastName>
    <ReturnCompanyName></ReturnCompanyName>
    <ReturnStreetName></ReturnStreetName>
    <ReturnBuildingNumber></ReturnBuildingNumber>
    <ReturnFlatNumber></ReturnFlatNumber>
    <ReturnCity></ReturnCity>
    <ReturnPostCode></ReturnPostCode>
    <ReturnPhoneNumber></ReturnPhoneNumber>
    <ReturnPack></ReturnPack>
    <TransferDescription></TransferDescription>
    <PrintAdress>1</PrintAdress>
    <ReturnAvailable></ReturnAvailable>
    <ReturnQuantity></ReturnQuantity>
</GenerateBusinessPack>
```

4.3.3.2 Response – GenerateBusinessPack

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code (confirmation or change)
<DestinationId>	Pickup point identifier
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)

RESPONSE EXAMPLE – GenerateBusinessPack

```
<GenerateBusinessPack>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>2900000000230</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>142450</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
</GenerateBusinessPack>
```

4.3.4 GenerateShippingCode

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Shipment pre-advice with the option to send without a label (generate a shipping code)
Additional information	The <PackCode> field in the response will be enabled on November 4, 2026. `<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<FirstName>` and `<LastName>` fields are required if no value is provided in the `<CompanyName>` field. The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field.

4.3.4.1 Request – GenerateShippingCode

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	Conditional	30 char
`<LastName>`	Recipient last name	Conditional	30 char
`<CompanyName>`	Recipient company name	Conditional	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<SenderEMail>`	Sender email address	**Yes**	60 char
`<SenderFirstName>`	Sender first name	Conditional	30 char
`<SenderLastName>`	Sender last name	Conditional	30 char
`<SenderCompanyName>`	Sender company name	Conditional	70 char
`<SenderStreetName>`	Sender street name	**Yes**	30 char

Field	Description	Required	Characters
<SenderBuildingNumber>	Sender building number	**Yes**	10 char
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Building number in the return address	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<TransferDescription>	Reference data	No	70 char
<ReturnPack>	Return parcel	No	T = return parcel
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<AutoDestinationChange>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<ConfirmTermsOfUse>			
<ConfirmMarketing>			
<CofirmMailing>			
<CashOnDelivery>			
<AmountCashOnDelivery>			
<PackValue>			
<Insurance>			

REQUEST EXAMPLE – GenerateShippingCode

```
<GenerateShippingCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
```

REQUEST EXAMPLE – GenerateShippingCode

```
    <PhoneNumber>111222333</PhoneNumber>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <AlternativeDestinationCode></AlternativeDestinationCode>
    <BoxSize>S</BoxSize>
    <EMail>receiver@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Adam</SenderFirstName>
    <SenderLastName>Tester</SenderLastName>
    <SenderCompanyName>Testowa</SenderCompanyName>
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
    <SenderFlatNumber>2</SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ReturnDestinationCode></ReturnDestinationCode>
    <ReturnEMail></ReturnEMail>
    <ReturnFirstName></ReturnFirstName>
    <ReturnLastName></ReturnLastName>
    <ReturnCompanyName></ReturnCompanyName>
    <ReturnStreetName></ReturnStreetName>
    <ReturnBuildingNumber></ReturnBuildingNumber>
    <ReturnFlatNumber></ReturnFlatNumber>
    <ReturnCity></ReturnCity>
    <ReturnPostCode></ReturnPostCode>
    <ReturnPhoneNumber></ReturnPhoneNumber>
    <ReturnPack></ReturnPack>
    <TransferDescription></TransferDescription>
    <PrintAdress>1</PrintAdress>
    <ReturnAvailable></ReturnAvailable>
    <ReturnQuantity></ReturnQuantity>
</GenerateShippingCode>
```

4.3.4.2 Response – GenerateShippingCode

Field	Description
<Err>	Error number
<ErrDes>	Error description
<ShippingCode>	Label-free shipping code
<PackCode>	Parcel number
<DestinationCode>	Pickup point code (after verification)
<DestinationId>	Pickup point identifier
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)

Field	Description
<ShippingCodeExpirationDate>	Shipping code expiration date

RESPONSE EXAMPLE – GenerateShippingCode

```
<GenerateShippingCode>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <ShippingCode>1618885846</ShippingCode>
    <PackCode>3316188858461</PackCode>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>0</PackPrice>
    <PackPaid>true</PackPaid>
    <ShippingCodeExpirationDate>2027-01-
01T00:00:00+01:00</ShippingCodeExpirationDate>
</GenerateShippingCode>
```

4.3.5 LabelPrintDuplicateListTwo

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Retrieve labels again for multiple parcels in PDF, PDF10, EPL, or ZPL format
Additional information	Parcel numbers should be provided in `<string>` elements of the `<PackCodeList>` collection. The maximum number of parcels in the `<PackCodeList>` collection is 50.

4.3.5.1 Request – LabelPrintDuplicateListTwo

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char
<Format>	Label format	**Yes**	PDF; PDF10; EPL; ZPL; ZPL_300DPI
<PackCodeList>	Parcel number	**Yes**	13 char

REQUEST EXAMPLE – LabelPrintDuplicateListTwo

```
<LabelPrintDuplicateListTwo>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <Format>ZPL</Format>
    <PackCodeList>
        <string>3401009538179</string>
        <string>3401009538186</string>
    </PackCodeList>
</LabelPrintDuplicateListTwo>
```

4.3.5.2 Response – LabelPrintDuplicateListTwo

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode>	Parcel number
<LabelData>	Base64-encoded label

RESPONSE EXAMPLE – LabelPrintDuplicateListTwo

```
<LabelPrintDuplicateListTwo>
    <PackCode>3401009538179</PackCode>
    <Err>0</Err>
    <ErrDes>saved</ErrDes>
</LabelPrintDuplicateListTwo>
<LabelPrintDuplicateListTwo>
    <PackCode>3401009538186</PackCode>
    <Err>0</Err>
    <ErrDes>saved</ErrDes>
</LabelPrintDuplicateListTwo>
<LabelData>Base64-encoded label</LabelData>
```

4.3.6 GenerateNumberWaybills

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Reserving Parcel Numbers
Additional information	The maximum pool size in a single call is 10 000. Reserved parcel numbers do not expire. Pre-advice data for reserved numbers should be submitted using the `PutCustomerPack` method.

4.3.6.1 Request – GenerateNumberWaybills

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char
<NumberOfWaybills>	Number of parcels to reserve	**Yes**	5 char

REQUEST EXAMPLE – GenerateNumberWaybills

```
<GenerateNumberWaybills>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <NumberOfWaybills>2</NumberOfWaybills>
</GenerateNumberWaybills>
```

4.3.6.2 Response – GenerateNumberWaybills

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode>	Reserved parcel number

RESPONSE EXAMPLE – GenerateNumberWaybills

```
<GenerateNumberWaybills>
    <Err>000</Err>
    <ErrDes>ok</ErrDes>
</GenerateNumberWaybills>
<GenerateNumberWaybills1>
    <PackCode>3401009510380</PackCode>
</GenerateNumberWaybills1>
<GenerateNumberWaybills1>
    <PackCode>3401009510397</PackCode>
</GenerateNumberWaybills1>
```

4.3.7 PutCustomerPack

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Pre-advising parcels from a pool of numbers generated using the `GenerateNumberWaybills` method
Additional information	Pre-advice data can be submitted only once for a given reserved parcel number. `<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field.

4.3.7.1 Request – PutCustomerPack

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PackCode>`	Parcel number	**Yes**	13 char
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	No	30 char
`<LastName>`	Recipient last name	No	30 char
`<CompanyName>`	Recipient company name	No	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<SenderEMail>`	Sender email address	**Yes**	60 char
`<SenderFirstName>`	Sender first name	Conditional	30 char
`<SenderLastName>`	Sender last name	Conditional	30 char
`<SenderCompanyName>`	Sender company name	Conditional	70 char
`<SenderStreetName>`	Sender street name	**Yes**	30 char

Field	Description	Required	Characters
<SenderBuildingNumber>	Sender building number	**Yes**	10 char
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Return building number	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<TransferDescription>	Reference data	No	70 char
<ReturnPack>	Return parcel	No	T = return parcel
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<CashOnDelivery>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<AmountCashOnDelivery>			
<Insurance>			
<PackValue>			

REQUEST EXAMPLE – PutCustomerPack

```
<PutCustomerPack>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PackCode>3401009513633</PackCode>
    <PhoneNumber>111222333</PhoneNumber>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
```

REQUEST EXAMPLE – PutCustomerPack

```
    <AlternativeDestinationCode></AlternativeDestinationCode>
    <BoxSize>S</BoxSize>
    <EMail>receiver@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName></CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Adam</SenderFirstName>
    <SenderLastName>Tester</SenderLastName>
    <SenderCompanyName>Test</SenderCompanyName>
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
    <SenderFlatNumber>2</SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ReturnDestinationCode></ReturnDestinationCode>
    <ReturnEMail></ReturnEMail>
    <ReturnFirstName></ReturnFirstName>
    <ReturnLastName></ReturnLastName>
    <ReturnCompanyName></ReturnCompanyName>
    <ReturnStreetName></ReturnStreetName>
    <ReturnBuildingNumber></ReturnBuildingNumber>
    <ReturnFlatNumber></ReturnFlatNumber>
    <ReturnCity></ReturnCity>
    <ReturnPostCode></ReturnPostCode>
    <ReturnPhoneNumber></ReturnPhoneNumber>
    <ReturnPack></ReturnPack>
    <TransferDescription></TransferDescription>
    <PrintAdress>1</PrintAdress>
    <ReturnAvailable></ReturnAvailable>
    <ReturnQuantity></ReturnQuantity>
</PutCustomerPack>
```

4.3.7.2 Response – PutCustomerPack

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code (after verification)
<DestinationId>	Pickup point identifier
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)

RESPONSE EXAMPLE – PutCustomerPack

```
<PutCustomerPack>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009513633</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
</PutCustomerPack>
```

4.3.8 PutCustomerPackCanceled

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Canceling a previously pre-advised parcel
Additional information	-

4.3.8.1 Request – PutCustomerPackCanceled

Field	Description	Required	Characters
<PackCode>	Parcel number	**Yes**	13 char
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – PutCustomerPackCanceled

```
<PutCustomerPackCanceled>
    <PackCode>3401009538315</PackCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</PutCustomerPackCanceled>
```

4.3.8.2 Response – PutCustomerPackCanceled

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode>	Canceled parcel number

RESPONSE EXAMPLE – PutCustomerPackCanceled

```
<CustomerPackCanceled>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode>3401009538315</PackCode>
</CustomerPackCanceled>
```

4.3.9 PutCustomerShippingCodeCanceled

Field	Value
Business process	Pre-advice and Label Generation
Usage recommendation	Canceling a shipping code
Additional information	-

4.3.9.1 Request – PutCustomerShippingCodeCanceled

Field	Description	Required	Characters
<ShippingCode>	Label-free shipping code	**Yes**	10 char
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – PutCustomerShippingCodeCanceled

```
<PutCustomerShippingCodeCanceled>
    <ShippingCode>4216685168</ShippingCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</PutCustomerShippingCodeCanceled>
```

4.3.9.2 Response – PutCustomerShippingCodeCanceled

Field	Description
<Err>	Error code
<ErrDes>	Error description
<ShippingCode>	Label-free shipping code

RESPONSE EXAMPLE – PutCustomerShippingCodeCanceled

```
<PutCustomerShippingCodeCanceled>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <ShippingCode>4216685168</ShippingCode>
</PutCustomerShippingCodeCanceled>
```

4.4 Goods Handover Protocol

4.4.1 GenerateProtocol

Field	Value
Business process	Goods Handover Protocol
Usage recommendation	Generating a Goods Handover Protocol
Additional information	The protocol is generated in PDF format for the specified parcel numbers. The method returns information about parcels with a status other than pre-advised. The maximum number of parcels declared on one protocol is 571. The time returned in the `<DATA_MOD>` field is in UTC+1 (local Polish time).

4.4.1.1 Request – GenerateProtocol

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<parcels>`	Field for specifying `<unsignedLong>` tags containing parcel numbers	**Yes**	-
`<unsignedLong>`	Parcel number	**Yes**	13 char

REQUEST EXAMPLE – GenerateProtocol

```
<GenerateProtocol>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <parcels>
        <unsignedLong>2100123123123</unsignedLong>
        <unsignedLong>2100123123124</unsignedLong>
    </parcels>
</GenerateProtocol>
```

4.4.1.2 Response – GenerateProtocol

Field	Description
`<Err>`	Error number – when `<Err>` is `210`, the parcel has a status other than pre-advised, which means it will not be added to the protocol
`<ErrDes>`	Error description
`<ProtocolCode>`	Protocol number (if `<Err>` is `0`)
`<PackCodeRUCH>`	Parcel number
`<DATA_MOD>`	Status modification date
`<status>`	Status number
`<status_opis>`	Status description

Field	Description
<LabelData>	Protocol in PDF format – Base64 encoding
<Table>	Area containing the status of a single parcel

RESPONSE EXAMPLE – GenerateProtocol
All parcels were added to the handover protocol

```
<GenerateProtocolResult>
  <Table>
    <Err>0</Err>
    <ErrDes>OK</ErrDes>
    <ProtocolCode>1000185903089</ProtocolCode>
    <PackCodeRUCH>2100123123123</PackCodeRUCH>
    <DATA_MOD>2024-10-22T13:12:55+02:00</DATA_MOD>
    <status>200</status>
    <status_opis>Zaawizowana w PwR</status_opis>
  </Table>
  <Table>
    <Err>0</Err>
    <ErrDes>OK</ErrDes>
    <ProtocolCode>1000185903089</ProtocolCode>
    <PackCodeRUCH>2100123123124</PackCodeRUCH>
    <DATA_MOD>2024-10-22T13:12:55+02:00</DATA_MOD>
    <status>200</status>
    <status_opis>Zaawizowana w PwR</status_opis>
  </Table>
</GenerateProtocolResult>
<LabelData>Base64-encoded label</LabelData>
```

RESPONSE EXAMPLE – GenerateProtocol
Some parcels were not added to the handover protocol

```
<GenerateProtocolResult>
  <Table>
    <Err>210</Err>
    <ErrDes>NIE DODANE DO PROTOKOŁU</ErrDes>
    <PackCodeRUCH>2100123123123</PackCodeRUCH>
    <DATA_MOD>2024-10-22T13:15:12+02:00</DATA_MOD>
    <status>0</status>
    <status_opis>Brak Paczki</status_opis>
  </Table>
  <Table>
    <Err>0</Err>
    <ErrDes>OK</ErrDes>
    <ProtocolCode>1000185903089</ProtocolCode>
    <PackCodeRUCH>2100123123124</PackCodeRUCH>
    <DATA_MOD>2024-10-22T13:12:55+02:00</DATA_MOD>
    <status>200</status>
    <status_opis>Zaawizowana w PwR</status_opis>
  </Table>
</GenerateProtocolResult>
<LabelData>Base64-encoded label</LabelData>
```

4.5 Courier Vehicle Availability Check

4.5.1 GetAvailablePickups

Field	Value
Business process	Checking Courier Pickup Availability
Usage recommendation	Retrieving a list of the nearest available courier pickup dates and times for the specified location
Additional information	The time returned in the `<MinReadyDate>` and `<MaxPickupDate>` fields is in UTC+1 (local Polish time).

4.5.1.1 Request – GetAvailablePickups

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PostCode>`	Postal code	**Yes**	6 char
`<City>`	City	No	30 char
`<Street>`	Street name	No	30 char
`<BuildingNo>`	Building and unit number	No	10 char
`<Email>`	Email address	No	60 char
`<PartnerName>`	Sender company name	No	30 char
`<PersonName>`	Customer first name	No	30 char
`<PersonSurname>`	Customer last name	No	30 char
`<Telephone>`	Customer phone number	No	9 char
`<ReadyDate>`	Date and time used to check courier pickup availability	**Yes**	YYYY-MM-DDThh:mm:ss

REQUEST EXAMPLE – GetAvailablePickups

```
<GetAvailablePickups>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PostCode>03-236</PostCode>
    <City></City>
    <Street></Street>
    <BuildingNo></BuildingNo>
    <Email></Email>
    <PartnerName></PartnerName>
    <PersonName></PersonName>
    <PersonSurname></PersonSurname>
    <Telephone></Telephone>
    <ReadyDate>2024-10-23T12:40:53.481Z</ReadyDate>
</GetAvailablePickups>
```

4.5.1.2 Response – GetAvailablePickups

Field	Description
<Err>	Error number
<ErrDes>	Error description
<Date>	Courier pickup day
<MinReadyDate>	Start date and time of courier pickup availability
<MaxPickupDate>	End date and time of courier pickup availability
<MinimumInterval>	Minimum interval between the declared values in the <MinReadyDate> and <MaxPickupDate> fields
<CourierZone>	Courier zone assigned to the available courier pickup time slot

RESPONSE EXAMPLE – GetAvailablePickups

```
<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>
            <CourierZone>A</CourierZone>
        </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>
            <CourierZone>A</CourierZone>
        </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>
            <CourierZone>A</CourierZone>
        </AvailablePickupDay>
    </Data>
</GetAvailablePickupsResult>
```

4.6 Ordering a Courier Pickup

4.6.1 CallPickupNew

Field	Value
Business process	Ordering a Courier Pickup
Usage recommendation	Ordering a courier pickup from a specified address
Additional information	The request must include the `<ReadyDate>` and `<PickupDate>` parameters defining the courier pickup time slot. Available courier arrival time slots are returned in the response from the `GetAvailablePickups` method (section 4.5.1). The time returned in the `<ReadyDate>` and `<PickupDate>` fields is in UTC+1 (local Polish time). Parcel numbers should be provided in `<string>` elements of the `<PackList>` collection. If the `<ReadyDate>` field contains a time earlier than the current time, the current pickup readiness time is set. A courier pickup can be canceled by contacting an ORLEN Paczka account manager or the ORLEN Paczka helpline and providing the courier pickup number.

4.6.1.1 Request – CallPickupNew

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PackList>`	Parcel numbers to be collected	**Yes**	13 char
`<ReadyDate>`	Start date and time when the parcel is ready for collection	**Yes**	YYYY-MM-DDThh:mm:ss
`<PickupDate>`	End date and time when the parcel is ready for collection	**Yes**	YYYY-MM-DDThh:mm:ss
`<PostCode>`	Postal code	**Yes**	6 char
`<City>`	City	**Yes**	30 char
`<Street>`	Street name	**Yes**	30 char
`<BuildingNo>`	Building and unit number	No	10 char
`<Email>`	Email address	**Yes**	60 char
`<PartnerName>`	Sender company name	**Yes**	30 char
`<PersonName>`	Customer first name	No	30 char
`<PersonSurname>`	Customer last name	No	30 char
`<Telephone>`	Customer phone number	No	9 char
`<ExternalNumber>`	External courier pickup identifier	No	30 char

REQUEST EXAMPLE – CallPickupNew

```
<CallPickupNew>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</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@orlenpaczka.pl</Email>
    <PartnerName>Test</PartnerName>
    <PersonName>Jan</PersonName>
    <PersonSurname>Testowy</PersonSurname>
    <Telephone></Telephone>
    <ExternalNumber></ExternalNumber>
</CallPickupNew>
```

4.6.1.2 Response – CallPickupNew

Field	Description
<Err>	Error number
<ErrDes>	Error description
<Data>	Courier pickup request number

RESPONSE EXAMPLE – CallPickupNew

```
<CallPickupNewResult>
    <Err>0</Err>
    <ErrDes>Success</ErrDes>
    <Data>14681063</Data>
</CallPickupNewResult>
```

4.7 Parcel Status Tracking

4.7.1 GiveMePackStatusList

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving the Latest Status for Multiple Parcels
Additional information	The method does not return tracking statuses older than 90 days or statuses for parcels linked to the queried parcel (e.g. consumer returns). Parcel numbers should be provided in `<string>` elements of the `<PackCodes>` collection. The maximum number of parcels in the `<PackCodes>` collection is 1 000. The time returned in the `<Data>` field is in UTC+1 (local Polish time). Ignore the “Z” character at the end of the time format.

4.7.1.1 Request – GiveMePackStatusList

Field	Description	Required	Characters
`<PackCodes>`	Parcel number	**Yes**	13 char
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMePackStatusList

```
<GiveMePackStatusList>
    <PackCodes>
        <string>2100123123123</string>
        <string>2100123123125</string>
    </PackCodes>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMePackStatusList>
```

4.7.1.2 Response – GiveMePackStatusList

Field	Description
`<PackCode>`	Parcel number
`<Trans>`	Parcel status
`<Trans_Des>`	Status description
`<Data>`	Date from which the parcel has the returned status
`<Destination>`	Pickup point code
`<PackStatus>`	Area containing the status of a single parcel

RESPONSE EXAMPLE – GiveMePackStatusList

```
<GiveMePackStatusList>
  <PackStatus>
     <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>
  </PackStatus>
  <PackStatus>
     <PackCode>2100123123125</PackCode>
     <Trans>200</Trans>
     <Trans_Des>Zaawizowana do PwR</Trans_Des>
     <Data>2024-10-22T09:45:23.1436745Z</Data>
     <Destination>NS-358394-N4-35</Destination>
  </PackStatus>
</GiveMePackStatusList>
```

4.7.2 GiveMePackStatusFullHistoryList

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving Full Status History for Multiple Parcels
Additional information	The method does not return tracking statuses older than 90 days or statuses for parcels linked to the queried parcel (e.g. consumer returns). Parcel numbers should be provided in `<string>` elements of the `<PackCodes>` collection. The maximum number of parcels in the `<PackCodes>` collection is 1000. The time returned in the `<Data>` field is in UTC+1 (local Polish time). Ignore the "Z" character at the end of the time format.

4.7.2.1 Request – GiveMePackStatusFullHistoryList

Field	Description	Required	Characters
<PackCodes>	Parcel number	**Yes**	13 char
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMePackStatusFullHistoryList

```
<GiveMePackStatusFullHistoryList>
    <PackCodes>
        <string>3401009538483</string>
        <string>3401009538490</string>
    </PackCodes>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMePackStatusFullHistoryList>
```

4.7.2.2 Response – GiveMePackStatusFullHistoryList

Field	Description
<PackCode>	Parcel number
<Trans>	Parcel status
<Trans_Des>	Status description
<Data>	Date from which the parcel has the returned status
<Destination>	Pickup point code
<Cl_From>	Number of the logistics center in the area where the parcel was sent
<Cl_From_Des>	Name of the logistics center in the area where the parcel was sent
<Cl>	Number of the logistics center in the area where the shipment was pre-advised
<Cl_Des>	Name of the logistics center in the area where the shipment was pre-advised

Field	Description
<Attribute>	Parcel attribute (mobilny – mobile point, alternatywny – alternative point, zmieniony – changed point, powrot – shipment return to the sending point, 2_powrot – shipment return to the sending warehouse, zwrot – return parcel to the sender warehouse)
<StreetName>	Pickup point street name
<City>	Pickup point city
<OpeningHours>	Pickup point opening hours
<Location>	Pickup point location description
<PackStatus>	Area containing the status of a single parcel

RESPONSE EXAMPLE – GiveMePackStatusFullHistoryList

```
<GiveMePackStatusFullHistoryList>
    <PackStatus>
        <PackCode>3401009538483</PackCode>
        <Trans>201</Trans>
        <Trans_Des>Anulowane awizo</Trans_Des>
        <Data>2026-07-24T14:10:17Z</Data>
        <Destination>NS-358394-N4-35</Destination>
        <Cl_From>0000</Cl_From>
        <Cl_From_Des>SC</Cl_From_Des>
        <Cl>4902</Cl>
        <Cl_Des>NOWY SĄCZ</Cl_Des>
        <Attribute />
        <StreetName>NAŚCISZOWSKA</StreetName>
        <City>Nowy Sącz</City>
        <OpeningHours>Pn-Pt:09:00-18:00, So:  nieczynne,
Nd:  nieczynne</OpeningHours>
        <Location>Furgonetka Regionalna Telewizja Kablowa naprzeciwko
składu materiałów budowlanych</Location>
    </PackStatus>
    <PackStatus>
        <PackCode>3401009538490</PackCode>
        <Trans>200</Trans>
        <Trans_Des>Zaawizowana do PwR</Trans_Des>
        <Data>2026-07-24T13:46:48.1152607Z</Data>
        <Destination>NS-358394-N4-35</Destination>
        <Cl_From>0000</Cl_From>
        <Cl_From_Des>SC</Cl_From_Des>
        <Cl>4902</Cl>
        <Cl_Des>NOWY SĄCZ</Cl_Des>
        <Attribute />
        <StreetName>NAŚCISZOWSKA</StreetName>
        <City>Nowy Sącz</City>
        <OpeningHours>Pn-Pt:09:00-18:00, So:  nieczynne,
Nd:  nieczynne</OpeningHours>
        <Location>Furgonetka Regionalna Telewizja Kablowa naprzeciwko
składu materiałów budowlanych</Location>
    </PackStatus>
    <PackStatus>
        <PackCode>3401009538483</PackCode>
        <Trans>200</Trans>
        <Trans_Des>Zaawizowana do PwR</Trans_Des>
```

RESPONSE EXAMPLE – GiveMePackStatusFullHistoryList

```
        <Data>2026-07-24T13:46:39.9822849Z</Data>
        <Destination>NS-358394-N4-35</Destination>
        <Cl_From>0000</Cl_From>
        <Cl_From_Des>SC</Cl_From_Des>
        <Cl>4902</Cl>
        <Cl_Des>NOWY SĄCZ</Cl_Des>
        <Attribute />
        <StreetName>NAŚCISZOWSKA</StreetName>
        <City>Nowy Sącz</City>
        <OpeningHours>Pn-Pt:09:00-18:00, So:  nieczynne,
Nd:  nieczynne</OpeningHours>
        <Location>Furgonetka Regionalna Telewizja Kablowa naprzeciwko
składu materiałów budowlanych</Location>
    </PackStatus>
</GiveMePackStatusFullHistoryList>
```

4.7.3 GiveMeShippingCodeStatusFullHistory

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving Full Status History Based on a Shipping Code
Additional information	The time returned in the `<Data>` field is in UTC+1 (local Polish time). Ignore the "Z" character at the end of the time format.

4.7.3.1 Request – GiveMeShippingCodeStatusFullHistory

Field	Description	Required	Characters
`<ShippingCode>`	Label-free shipping code	**Yes**	10 char
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMeShippingCodeStatusFullHistory

```
<GiveMeShippingCodeStatusFullHistory>
    <ShippingCode>2127392584</ShippingCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMeShippingCodeStatusFullHistory>
```

4.7.3.2 Response – GiveMeShippingCodeStatusFullHistory

Field	Description
`<PackCode>`	Parcel number
`<Trans>`	Parcel status
`<Trans_Des>`	Status description
`<Data>`	Date from which the parcel has the returned status
`<Destination>`	Pickup point code
`<PartnerId>`	API identifier used to create the shipping code
`<PsdDo>`	Pickup point identifier
`<Cl_From>`	Number of the logistics center in the area where the parcel was sent
`<Cl_From_Des>`	Name of the logistics center in the area where the parcel was sent
`<Cl>`	Number of the logistics center in the area where the shipment was delivered
`<Cl_Des>`	Name of the logistics center in the area where the shipment was delivered
`<Attribute>`	Parcel attribute (mobilny – mobile point, alternatywny – alternative point, zmieniony – changed point, powrot – shipment return to the sending point, 2_powrot – shipment return to the sending warehouse, zwrot – return parcel to the sender warehouse)
`<StreetName>`	Pickup point street name

Field	Description
<City>	Pickup point city
<OpeningHours>	Pickup point opening hours
<Location>	Pickup point location description

RESPONSE EXAMPLE – GiveMeShippingCodeStatusFullHistory

```
<GiveMeShippingCodeStatusFullHistory>
    <PackCode>3321273925843</PackCode>
    <Trans>200</Trans>
    <Trans_Des>Zaawizowana do PwR</Trans_Des>
    <Data>2026-07-24T14:16:31.9917171Z</Data>
    <Destination>NS-358394-N4-35</Destination>
    <PartnerId>username</PartnerId>
    <PsdDo>358394</PsdDo>
    <Cl_From>0000</Cl_From>
    <Cl_From_Des>SC</Cl_From_Des>
    <Cl>4902</Cl>
    <Cl_Des>NOWY SĄCZ</Cl_Des>
    <Attribute />
    <StreetName>NAŚCISZOWSKA</StreetName>
    <City>Nowy Sącz</City>
    <OpeningHours>Pn-Pt:09:00-18:00, So:  nieczynne,
Nd:  nieczynne</OpeningHours>
    <Location>Furgonetka Regionalna Telewizja Kablowa naprzeciwko składu
materiałów budowlanych</Location>
</GiveMeShippingCodeStatusFullHistory>
```

4.7.4 GiveMePackStatus

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving the Latest Status of a Single Parcel.
Additional information	The method does not return tracking statuses older than 90 days or statuses for parcels linked to the queried parcel (e.g. consumer returns). The `<PhoneNumber>` and `<PackCode>` fields are conditionally required – at least one of them, or both, must be provided. `<PhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The time returned in the `<Data>` field is in UTC+1 (local Polish time). Ignore the "Z" character at the end of the time format.

4.7.4.1 Request – GiveMePackStatus

Field	Description	Required	Characters
<PhoneNumber>	Recipient phone number	Conditional	9 char / 12 char
<PackCode>	Parcel number	Conditional	13 char
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMePackStatus

```
<GiveMePackStatus>
    <PhoneNumber>111222333</PhoneNumber>
    <PackCode>3401009538780</PackCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMePackStatus>
```

4.7.4.2 Response – GiveMePackStatus

Field	Description
<PackCode>	Parcel number
<Trans>	Parcel status
<Trans_Des>	Status description
<Data>	Date from which the parcel has the returned status
<Destination>	Pickup point code

RESPONSE EXAMPLE – GiveMePackStatus

```
<GiveMePackStatus>
    <PackCode>3401009538780</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>
</GiveMePackStatus>
```

4.7.5 GiveMePackStatusFullHistory

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving the full parcel status history, including operational returns (return to sender)
Additional information	The method does not return tracking statuses older than 90 days or statuses for parcels linked to the queried parcel (e.g. consumer returns). The time returned in the `<Data>` field is in UTC+1 (local Polish time). Ignore the "Z" character at the end of the time format.

4.7.5.1 Request – GiveMePackStatusFullHistory

Field	Description	Required	Characters
`<PackCode>`	Parcel number	**Yes**	13 char
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMePackStatusFullHistory

```
<GiveMePackStatusFullHistory>
    <PackCode>3401009538797</PackCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMePackStatusFullHistory>
```

4.7.5.2 Response – GiveMePackStatusFullHistory

Field	Description
`<PackCode>`	Parcel number
`<Trans>`	Parcel status
`<Trans_Des>`	Status description
`<Data>`	Date from which the parcel has the returned status
`<Destination>`	Pickup point code
`<Cl_From>`	Number of the logistics center in the area where the parcel was sent
`<Cl_From_Des>`	Name of the logistics center in the area where the parcel was sent
`<Cl>`	Number of the logistics center in the area where the shipment was pre-advised
`<Cl_Des>`	Name of the logistics center in the area where the shipment was pre-advised
`<Attribute>`	Parcel attribute (mobilny – mobile point, alternatywny – alternative point, zmieniony – changed point, powrot – shipment return to the sending point, 2_powrot – shipment return to the sending warehouse, zwrot – return parcel to the sender warehouse)

Field	Description
<StreetName>	Pickup point street name
<City>	Pickup point city
<OpeningHours>	Pickup point opening hours
<Location>	Pickup point location description

RESPONSE EXAMPLE – GiveMePackStatusFullHistory

```
<GiveMePackStatusFullHistory>
    <PackCode>3401009538797</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>
    <Cl_From>0000</Cl_From>
    <Cl_From_Des>SC</Cl_From_Des>
    <Cl>0130</Cl>
    <Cl_Des>WARSZAWA</Cl_Des>
    <Attribute/>
    <StreetName>NOWY ŚWIAT 6/12</StreetName>
    <City>Warszawa</City>
    <OpeningHours>Pn-Pt:06:15-16:00, So:  nieczynne, Nd:
nieczynne</OpeningHours>
    <Location>Automat paczkowy obok sklepu</Location>
</GiveMePackStatusFullHistory>
```

4.7.6 GetGeneratedParcels

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving pre-advised parcel numbers together with their pre-advice dates for the specified date range
Additional information	The response contains parcel numbers pre-advised no earlier than 30 days before the query date. The time returned in the `<Data>` field is in UTC+1 (local Polish time).

4.7.6.1 Request – GetGeneratedParcels

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<dateFrom>`	Start date	**Yes**	YYYY-MM-DDThh:mm:ss
`<dateTo>`	End date	**Yes**	YYYY-MM-DDThh:mm:ss

REQUEST EXAMPLE – GetGeneratedParcels

```
<GetGeneratedParcels>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <dateFrom>2024-07-01T08:10:53.345Z</dateFrom>
    <dateTo>2024-07-15T16:34:31.575Z</dateTo>
</GetGeneratedParcels>
```

4.7.6.2 Response – GetGeneratedParcels

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<Data>	Parcel pre-advice date
<parcel>	Area containing the status of a single parcel

RESPONSE EXAMPLE – GetGeneratedParcels

```
<GetGeneratedParcels>
    <parcels>
        <Err>000</Err>
        <PackCode_RUCH>3335471929138</PackCode_RUCH>
        <Data>2024-07-10T09:37:35.177+02:00</Data>
    </parcels>
    <parcels>
        <Err>000</Err>
        <PackCode_RUCH>3401009533907</PackCode_RUCH>
        <Data>2024-07-23T09:46:44.427+02:00</Data>
    </parcels>
</GetGeneratedParcels>
```

4.7.7 GiveCurrentPack

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving the current parcel number when a return label has been generated for the original parcel
Additional information	-

4.7.7.1 Request – GiveCurrentPack

Field	Description	Required	Characters
<packCode>	Parcel number	**Yes**	13 char

REQUEST EXAMPLE – GiveCurrentPack

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

4.7.7.2 Response – GiveCurrentPack

Field	Description
<GiveCurrentPackResult>	Current parcel number

RESPONSE EXAMPLE – GiveCurrentPack

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

4.7.8 GiveMasterPack

Field	Value
Business process	Parcel Status Tracking
Usage recommendation	Retrieving the original parcel number when a return parcel number has been received.
Additional information	-

4.7.8.1 Request – GiveMasterPack

Field	Description	Required	Characters
<packCode>	Return parcel number	**Yes**	13 char

REQUEST EXAMPLE – GiveMasterPack

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

4.7.8.2 Response – GiveMasterPack

Field	Description
<GiveMasterPackResult>	Original (parent) parcel number

RESPONSE EXAMPLE – GiveMasterPack

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

4.8 Consumer Returns

4.8.1 GenerateCustomerReturn

Field	Value
Business process	Consumer Returns
Usage recommendation	Pre-advising a consumer return linked to the original parcel, intended for shipping with a label. The method does not return a label – after the return is created, the label must be retrieved using a label duplicate method.
Additional information	To enable the consumer return service, contact your ORLEN Paczka account manager. To generate a consumer return, the pre-advice data of the original parcel (declared, for example, in a `GenerateBusinessPackTwo`, `GenerateLabelBusinessPackListTwo`, or `GenerateLabelBusinessPackTwo` request) must include the `<ReturnQuantity>` parameter, and fields starting with "Return" must be left empty. In this case, the recipient data will be replaced with the sender data. `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix A consumer return linked to the original parcel inherits its additional parameters. The response to the request returns a new return parcel number. A label for this number must be generated using a method that returns label duplicates (**section 4.3**).

4.8.1.1 Request – GenerateCustomerReturn

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PackCode>`	Parcel number	**Yes**	13 char
`<SenderPhoneNumber>`	Sender phone number	**Yes**	9 char / 12 char

REQUEST EXAMPLE – GenerateCustomerReturn

```
<GenerateCustomerReturn>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PackCode>2100123123123</PackCode>
    <SenderPhoneNumber>111222333</SenderPhoneNumber>
</GenerateCustomerReturn>
```

4.8.1.2 Response – GenerateCustomerReturn

Field	Description
<Err>	Error code
<ErrDes>	Error description
<packCode_base>	Original Parcel Number
<packCode_return>	Return parcel number
<OBSZAR>	Area
<MIKROREJON>	Microregion
<SORTOWNIA>	Sorting facility
<KURIER_ZWROTY>	Return courier

RESPONSE EXAMPLE – GenerateCustomerReturn

```
<GenerateCustomerReturn>
    <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>
</GenerateCustomerReturn>
```

4.8.2 GenerateCustomerReturnShippingCode

Field	Value
Business process	Consumer Returns
Usage recommendation	Pre-advising a consumer return sent without a label that is linked to the original parcel
Additional information	To enable the consumer return service, contact your ORLEN Paczka account manager. To generate a consumer return, the pre-advice data of the original parcel must include the `<ReturnQuantity>` parameter, and fields starting with “Return” must be left empty. In this case, the recipient data will be replaced with the sender data. `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix A consumer return linked to the original parcel inherits its additional parameters.

4.8.2.1 Request – GenerateCustomerReturnShippingCode

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PackCode>`	Parcel number	**Yes**	13 char
`<SenderPhoneNumber>`	Sender phone number	**Yes**	9 char / 12 char

REQUEST EXAMPLE – GenerateCustomerReturnShippingCode

```
<GenerateCustomerReturnShippingCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PackCode>3401009538988</PackCode>
    <SenderPhoneNumber>123123123</SenderPhoneNumber>
</GenerateCustomerReturnShippingCode>
```

4.8.2.2 Response – GenerateCustomerReturnShippingCode

Field	Description
<Err>	Error code
<ErrDes>	Error description
<packCode_base>	Original Parcel Number
<shippingCode_return>	Return parcel shipping code
<OBSZAR>	Area
<MIKROREJON>	Microregion
<SORTOWNIA>	Sorting facility
<KURIER_ZWROTY>	Return courier

RESPONSE EXAMPLE – GenerateCustomerReturnShippingCode

```
<GenerateCustomerReturnShippingCode>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <packCode_base>3401009538988</packCode_base>
    <shippingCode_return>8925944048</shippingCode_return>
    <OBSZAR />
    <MIKROREJON>WSC3</MIKROREJON>
    <SORTOWNIA />
    <KURIER_ZWROTY>WSC3</KURIER_ZWROTY>
</GenerateCustomerReturnShippingCode>
```

4.8.3 GenerateStandardCustomerReturn

Field	Value
Business process	Consumer Returns
Usage recommendation	Pre-advising a consumer return linked to the original parcel, intended for shipping with a label. The method does not return a label – after the return is created, the label must be retrieved using a label duplicate method.
Additional information	To enable the consumer return service, contact your ORLEN Paczka account manager. The return delivery address is the company registered office address specified in the agreement or another address agreed with an ORLEN Paczka account manager. The return parcel is not linked to the original parcel. All required data and additional parameters must be provided in the current call. `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field. The response to the request returns a new return parcel number. If the `<PrintLabel>` field is set to `T`, the response contains the parcel data and the label in the `<Label>` field. If the field is set to `F`, the label can be retrieved later using a method that returns a label duplicate (**section 4.3**).

4.8.3.1 Request – GenerateStandardCustomerReturn

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PrintLabel>`	Print label	No	T – print; F – do not print
`<Format>`	Label format	**Yes**	PDF; EPL; ZPL; PNG
`<PrintType>`	Label type	**Yes**	1 – full address; 2 – anonymous address; 3 – recipient and sender name only
`<BoxSize>`	Parcel size	**Yes**	1 char
`<SenderFirstName>`	Sender first name	Conditional	30 char
`<SenderLastName>`	Sender last name	Conditional	30 char

Field	Description	Required	Characters
<SenderCompanyName>	Sender company name	Conditional	70 char
<SenderStreetName>	Sender street name	**Yes**	30 char
<SenderBuildingNumber>	Sender building number	**Yes**	10 char
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderEMail>	Sender email address	**Yes**	30 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in section 2.5.5)	No	30 char
<ExternalPackageNumber>	External parcel number (additional information in section 2.5.5)	No	30 char
<PackValue>	**The additional insurance service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<Insurance>			

REQUEST EXAMPLE – GenerateStandardCustomerReturn

```
<GenerateStandardCustomerReturn>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PrintLabel>T</PrintLabel>
    <Format>PDF</Format>
    <PrintType>1</PrintType>
    <BoxSize>S</BoxSize>
    <SenderFirstName>Jan</SenderFirstName>
    <SenderLastName>Nowak</SenderLastName>
    <SenderCompanyName>ORLEN Paczka</SenderCompanyName>
    <SenderStreetName>Annopol</SenderStreetName>
    <SenderBuildingNumber>17a</SenderBuildingNumber>
    <SenderFlatNumber></SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>02-236</SenderPostCode>
    <SenderEMail>return@orlenpaczka.pl</SenderEMail>
    <SenderPhoneNumber>111222333</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ExternalSenderPackageNumber></ExternalSenderPackageNumber>
    <ExternalPackageNumber></ExternalPackageNumber>
</GenerateStandardCustomerReturn>
```

4.8.3.2 Response – GenerateStandardCustomerReturn

Field	Description
<PackCode_RUCH>	Parcel number
<DEST_ODDZIAL>	Number of the logistics center handling return delivery
<KURIER_ZWROTY>	Courier handling return delivery
<OP_ADRES>	Street name for return delivery
<OP_KODPOCZTOWY>	Postal code for return delivery
<OP_MIASTO>	City for return delivery
<OP_NAZWA>	Return delivery name
<OP_KRAJ>	Country for return delivery
<Label>	Base64-encoded label (PrintLabel = T)

RESPONSE EXAMPLE – GenerateStandardCustomerReturn

```
<GenerateStandardCustomerReturn>
    <PackCode_RUCH>2100000000230</PackCode_RUCH>
    <DEST_ODDZIAL>0130</DEST_ODDZIAL>
    <KURIER_ZWROTY>03</KURIER_ZWROTY>
    <OP_ADRES>Annopol 17a</OP_ADRES>
    <OP_KODPOCZTOWY>03-236</OP_KODPOCZTOWY>
    <OP_MIASTO>Warszawa</OP_MIASTO>
    <OP_NAZWA>ORLEN Paczka</OP_NAZWA>
    <OP_KRAJ>PL</OP_KRAJ>
</GenerateStandardCustomerReturn>
<Label>Base64-encoded label</Label>
```

4.8.4 GenerateStandardCustomerReturnShippingCode

Field	Value
Business process	Consumer Returns
Usage recommendation	Pre-advising a consumer return sent without a label that is not linked to the original parcel
Additional information	To enable the consumer return service, contact your ORLEN Paczka account manager. The return delivery address is the company registered office address specified in the agreement or another address agreed with an ORLEN Paczka account manager. The return parcel is not linked to the original parcel. All required data and additional parameters must be provided in the current call. `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field.

4.8.4.1 Request – GenerateStandardCustomerReturnShippingCode

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<PrintType>`	Label type	**Yes**	1 – full address; 2 – anonymous address; 3 – recipient and sender name only
`<BoxSize>`	Parcel size	**Yes**	1 char
`<SenderFirstName>`	Sender first name	Conditional	30 char
`<SenderLastName>`	Sender last name	Conditional	30 char
`<SenderCompanyName>`	Sender company name	Conditional	70 char
`<SenderStreetName>`	Sender street name	**Yes**	30 char
`<SenderBuildingNumber>`	Sender building number	**Yes**	10 char
`<SenderFlatNumber>`	Sender unit number	No	10 char
`<SenderCity>`	Sender city	**Yes**	30 char
`<SenderPostCode>`	Sender postal code	**Yes**	6 char
`<SenderEMail>`	Sender email address	**Yes**	30 char
`<SenderPhoneNumber>`	Sender phone number	**Yes**	9 char / 12 char

Field	Description	Required	Characters
<SenderOrders>	Reference data	No	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in section 2.5.5)	No	30 char
<ExternalPackageNumber>	External parcel number (additional information in section 2.5.5)	No	30 char
<PackValue>	**The additional insurance service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<Insurance>			

REQUEST EXAMPLE – GenerateStandardCustomerReturnShippingCode

```
<GenerateStandardCustomerReturnShippingCode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PrintType>1</PrintType>
    <BoxSize>S</BoxSize>
    <SenderFirstName>Jan</SenderFirstName>
    <SenderLastName>Nowak</SenderLastName>
    <SenderCompanyName>ORLEN Paczka</SenderCompanyName>
    <SenderStreetName>Annopol</SenderStreetName>
    <SenderBuildingNumber>17a</SenderBuildingNumber>
    <SenderFlatNumber></SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>02-236</SenderPostCode>
    <SenderEMail>return@orlenpaczka.pl</SenderEMail>
    <SenderPhoneNumber>111222333</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ExternalSenderPackageNumber></ExternalSenderPackageNumber>
    <ExternalPackageNumber></ExternalPackageNumber>
</GenerateStandardCustomerReturnShippingCode>
```

4.8.4.2 Response – GenerateStandardCustomerReturnShippingCode

Field	Description
<ShippingCode>	Label-free shipping code
<DEST_ODDZIAL>	Code of the logistics center handling return delivery
<KURIER_ZWROTY>	Courier handling return delivery
<OP_ADRES>	Street name for return delivery
<OP_KODPOCZTOWY>	Postal code for return delivery
<OP_MIASTO>	City for return delivery
<OP_NAZWA>	Return delivery name

Field	Description
<OP_KRAJ>	Country for return delivery

RESPONSE EXAMPLE – GenerateStandardCustomerReturnShippingCode

```
<GenerateStandardCustomerReturnShippingCode>
    <ShippingCode>1234567891</ShippingCode>
    <DEST_ODDZIAL>0130</DEST_ODDZIAL>
    <KURIER_ZWROTY>03</KURIER_ZWROTY>
    <OP_ADRES>Annopol 17a</OP_ADRES>
    <OP_KODPOCZTOWY>02-236</OP_KODPOCZTOWY>
    <OP_MIASTO>Warszawa</OP_MIASTO>
    <OP_NAZWA>ORLEN Paczka</OP_NAZWA>
    <OP_KRAJ>PL</OP_KRAJ>
</GenerateStandardCustomerReturnShippingCode>
```

4.8.5 GenerateBrokerReturn2Home

Field	Value
Business process	Consumer Returns
Usage recommendation	Pre-advice and label generation for a consumer return – method for brokers
Additional information	To enable the service, contact your ORLEN Paczka account manager. `<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field. The `<FirstName>` and `<LastName>` fields are required if no value is provided in the `<CompanyName>` field.

4.8.5.1 Request – GenerateBrokerReturn2Home

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<OriginalPackageNumber>`	Original parcel number linked to the return	No	13 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<EMail>`	Recipient email address	**Yes**	30 char
`<FirstName>`	Recipient first name	Conditional	30 char
`<LastName>`	Recipient last name	Conditional	30 char
`<CompanyName>`	Recipient company name	Conditional	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	**Yes**	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	**Yes**	30 char
`<PostCode>`	Recipient postal code	**Yes**	6 char
`<SenderEMail>`	Return sender email address	**Yes**	30 char
`<SenderFirstName>`	Return sender first name	Conditional	30 char
`<SenderLastName>`	Return sender last name	Conditional	30 char
`<SenderCompanyName>`	Return sender company name	Conditional	70 char

Field	Description	Required	Characters
<SenderStreetName>	Return sender street name	No	30 char
<SenderBuildingNumber>	Return sender building number	**Yes**	10 char
<SenderFlatNumber>	Return sender unit number	No	10 char
<SenderCity>	Return sender city	**Yes**	30 char
<SenderPostCode>	Return sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Return sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in **section 2.5.5**)	No	30 char
<ExternalPackageNumber>	External parcel number (additional information in **section 2.5.5**)	No	30 char
<Format>	Label format	**Yes**	PDF; PDF10; EPL; ZPL
<Serwis>	Technical field	No	null
<ItemsInPackage>	Parcel contents	No	30 char
<PackValue>	**The additional insurance service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<Insurance>			

REQUEST EXAMPLE – GenerateBrokerReturn2Home

```
<GenerateBrokerReturn2Home>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <BoxSize>S</BoxSize>
    <OriginalPackageNumber></OriginalPackageNumber>
    <PhoneNumber>111222333</PhoneNumber>
    <EMail>return@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>Test</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Adam</SenderFirstName>
    <SenderLastName>Tester</SenderLastName>
    <SenderCompanyName>ORLEN Paczka</SenderCompanyName>
```

REQUEST EXAMPLE – GenerateBrokerReturn2Home

```
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
    <SenderFlatNumber>2</SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ExternalSenderPackageNumber></ExternalSenderPackageNumber>
    <ExternalPackageNumber></ExternalPackageNumber>
    <Serwis></Serwis>
    <ItemsInPackage></ItemsInPackage>
    <Format>PDF</Format>
</GenerateBrokerReturn2Home>
```

4.8.5.2 Response – GenerateBrokerReturn2Home

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true / false)
<LabelData>	Base64-encoded label

RESPONSE EXAMPLE – GenerateBrokerReturn2Home

```
<GenerateBrokerReturn2Home>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009518508</PackCode_RUCH>
    <DestinationCode>0130</DestinationCode>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
</GenerateBrokerReturn2Home>

<LabelData>Base64-encoded label</LabelData>
```

4.8.6 GenerateBrokerShippingCodeReturn2Home

Field	Value
Business process	Consumer Returns
Usage recommendation	Pre-advice and generation of a label-free shipping code for a consumer return – method for brokers
Additional information	To enable the service, contact your ORLEN Paczka account manager. `<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field. The `<FirstName>` and `<LastName>` fields are required if no value is provided in the `<CompanyName>` field.

4.8.6.1 Request – GenerateBrokerShippingCodeReturn2Home

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<OriginalPackageNumber>`	Original parcel number linked to the return	No	13 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<EMail>`	Recipient email address	**Yes**	30 char
`<FirstName>`	Recipient first name	Conditional	30 char
`<LastName>`	Recipient last name	Conditional	30 char
`<CompanyName>`	Recipient company name	Conditional	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	**Yes**	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	**Yes**	30 char
`<PostCode>`	Recipient postal code	**Yes**	6 char
`<SenderEMail>`	Return sender email address	**Yes**	30 char
`<SenderFirstName>`	Return sender first name	Conditional	30 char
`<SenderLastName>`	Return sender last name	Conditional	30 char
`<SenderCompanyName>`	Return sender company name	Conditional	70 char

Field	Description	Required	Characters
<SenderStreetName>	Return sender street name	No	30 char
<SenderBuildingNumber>	Return sender building number	**Yes**	10 char
<SenderFlatNumber>	Return sender unit number	No	10 char
<SenderCity>	Return sender city	**Yes**	30 char
<SenderPostCode>	Return sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Return sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ExternalSenderPackageNumber>	External sender number (additional information in **section 2.5.5**)	No	30 char
<ExternalPackageNumber>	External parcel number (additional information in **section 2.5.5**)	No	30 char
<Serwis>	Technical field	No	null
<ItemsInPackage>	Parcel contents	No	30 char
<PackValue>	**The additional insurance service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<Insurance>			

REQUEST EXAMPLE – GenerateBrokerShippingCodeReturn2Home

```
<GenerateBrokerShippingCodeReturn2Home>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <BoxSize>S</BoxSize>
    <OriginalPackageNumber></OriginalPackageNumber>
    <PhoneNumber>111222333</PhoneNumber>
    <EMail>return@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Adam</SenderFirstName>
    <SenderLastName>Tester</SenderLastName>
    <SenderCompanyName>Test</SenderCompanyName>
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
```

REQUEST EXAMPLE – GenerateBrokerShippingCodeReturn2Home

```
    <SenderFlatNumber></SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ExternalSenderPackageNumber></ExternalSenderPackageNumber>
    <ExternalPackageNumber></ExternalPackageNumber>
    <Serwis></Serwis>
    <ItemsInPackage></ItemsInPackage>
</GenerateBrokerShippingCodeReturn2Home>
```

4.8.6.2 Response – GenerateBrokerShippingCodeReturn2Home

Field	Description
<Err>	Error number
<ErrDes>	Error description
<ShippingCode>	Label-free shipping code
<DestinationCode>	Pickup point code
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true / false)
<ShippingCodeExpirationDate>	Shipping code expiration date

RESPONSE EXAMPLE – GenerateBrokerShippingCodeReturn2Home

```
<GenerateBrokerShippingCodeReturn2Home>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <ShippingCode>1679364162</ShippingCode>
    <DestinationCode>0130</DestinationCode>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <ShippingCodeExpirationDate>2026-07-
31T22:00:00+02:00</ShippingCodeExpirationDate>
</GenerateBrokerShippingCodeReturn2Home>
```

4.9 Archive

The section below lists methods that remain active and can still be used but are not recommended for new integrations. A newer version of each method is available and recommended by ORLEN Paczka.

Archived method	Recommended method
GenerateLabelBusinessPack	GenerateLabelBusinessPackListTwo
GenerateLabelBusinessPackTwo	GenerateLabelBusinessPackListTwo
GenerateLabelBusinessPackList	GenerateLabelBusinessPackListTwo
LabelPrintDuplicate	LabelPrintDuplicateListTwo
LabelPrintDuplicateTwo	LabelPrintDuplicateListTwo
LabelPrintDuplicateList	LabelPrintDuplicateListTwo
CallPickup	CallPickupNew
GiveMeAllLocationWithAllData	GiveMeAllLocationWithAllDataWithZipCode
GiveMeAllApmLocation	GiveMeAllLocationWithAllDataWithZipCode
GiveMeAllRUCHLocation	GiveMeAllLocationWithAllDataWithZipCode
GiveMeAllRUCHZipcode	GiveMeAllLocationWithAllDataWithZipCode

The `GiveMeAllPSDLocation`, `GiveMePSDProvince`, `GiveMePSDZipcode`, and `GiveMeCityPSDLocation` methods are being phased out due to the discontinuation of kiosk support. These methods do not return pickup point data and **should not be used**.

4.9.1 GenerateLabelBusinessPack

Field	Value
Recommended method	`GenerateLabelBusinessPackListTwo`
Business process	Pre-advice and Label Generation
Usage recommendation	Shipment pre-advice and PDF label creation in one step
Additional information	`<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<FirstName>` and `<LastName>` fields are required if no value is provided in the `<CompanyName>` field. The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field.

4.9.1.1 Request – GenerateLabelBusinessPack

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	Conditional	30 char
`<LastName>`	Recipient last name	Conditional	30 char
`<CompanyName>`	Recipient company name	Conditional	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<SenderEMail>`	Sender email address	**Yes**	60 char
`<SenderFirstName>`	Sender first name	Conditional	30 char
`<SenderLastName>`	Sender last name	Conditional	30 char
`<SenderCompanyName>`	Sender company name	Conditional	70 char
`<SenderStreetName>`	Sender street name	**Yes**	30 char
`<SenderBuildingNumber>`	Sender building number	**Yes**	10 char

Field	Description	Required	Characters
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Building number in the return address	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<ReturnPack>	Return parcel	No	T = return parcel
<TransferDescription>	Reference data	No	70 char
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<PrintType>	Label type	**Yes**	1 – full address; 2 – anonymous address; 3 – recipient and sender name only
<CashOnDelivery>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<AmountCashOnDelivery>			
<PackValue>			
<Insurance>			

REQUEST EXAMPLE – GenerateLabelBusinessPack

```
<GenerateLabelBusinessPack>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <AlternativeDestinationCode></AlternativeDestinationCode>
    <BoxSize>S</BoxSize>
    <EMail>receiver@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName>ORLEN Paczka</CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <PhoneNumber>111222333</PhoneNumber>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Adam</SenderFirstName>
    <SenderLastName>Tester</SenderLastName>
    <SenderCompanyName>Test</SenderCompanyName>
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
    <SenderFlatNumber></SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ReturnDestinationCode></ReturnDestinationCode>
    <ReturnEMail></ReturnEMail>
    <ReturnFirstName></ReturnFirstName>
    <ReturnLastName></ReturnLastName>
    <ReturnCompanyName></ReturnCompanyName>
    <ReturnStreetName></ReturnStreetName>
    <ReturnBuildingNumber></ReturnBuildingNumber>
    <ReturnFlatNumber></ReturnFlatNumber>
    <ReturnCity></ReturnCity>
    <ReturnPostCode></ReturnPostCode>
    <ReturnPhoneNumber></ReturnPhoneNumber>
    <ReturnPack></ReturnPack>
    <TransferDescription></TransferDescription>
    <PrintAdress>1</PrintAdress>
    <ReturnAvailable></ReturnAvailable>
    <ReturnQuantity></ReturnQuantity>
    <PrintType>1</PrintType>
</GenerateLabelBusinessPack>
```

4.9.1.2 Response – GenerateLabelBusinessPack

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code (after verification)
<DestinationId>	Pickup point identifier

Field	Description
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)
<NameCL>	ORLEN Paczka logistics center name
<NrCL>	ORLEN Paczka logistics center number
<LabelData>	Base64-encoded label

RESPONSE EXAMPLE – GenerateLabelBusinessPack

```
<GenerateLabelBusinessPack>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009509636</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <NameCL>NOWY SĄCZ</NameCL>
    <NrCL>4902</NrCL>
</GenerateLabelBusinessPack>

<LabelData>Base64-encoded label</LabelData>
```

4.9.2 GenerateLabelBusinessPackTwo

Field	Value
Recommended method	`GenerateLabelBusinessPackListTwo`
Business process	Pre-advice and Label Generation
Usage recommendation	Shipment pre-advice and PDF label creation in one step
Additional information	`<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field.

4.9.2.1 Request – GenerateLabelBusinessPackTwo

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	No	30 char
`<LastName>`	Recipient last name	No	30 char
`<CompanyName>`	Recipient company name	No	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<SenderEMail>`	Sender email address	Conditional	60 char
`<SenderFirstName>`	Sender first name	Conditional	30 char
`<SenderLastName>`	Sender last name	Conditional	30 char
`<SenderCompanyName>`	Sender company name	No	70 char
`<SenderStreetName>`	Sender street name	**Yes**	30 char
`<SenderBuildingNumber>`	Sender building number	**Yes**	10 char
`<SenderFlatNumber>`	Sender unit number	No	10 char
`<SenderCity>`	Sender city	**Yes**	30 char

Field	Description	Required	Characters
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Building number in the return address	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<ReturnPack>	Return parcel	No	T = return parcel
<TransferDescription>	Reference data	No	70 char
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<PrintType>	Label type	**Yes**	1 – full address; 2 – anonymous address; 3 – recipient and sender name only
<AutoDestinationChange>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.** **Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<ConfirmTermsOfUse>			
<ConfirmMarketing>			
<CofirmMailing>			
<CashOnDelivery>			
<AmountCashOnDelivery>			
<PackValue>			
<Insurance>			

REQUEST EXAMPLE – GenerateLabelBusinessPackTwo

```
<GenerateLabelBusinessPackTwo>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <AlternativeDestinationCode></AlternativeDestinationCode>
    <BoxSize>S</BoxSize>
    <EMail>receiver@orlenpaczka.pl</EMail>
    <FirstName>Jan</FirstName>
    <LastName>Nowak</LastName>
    <CompanyName></CompanyName>
    <StreetName>Annopol</StreetName>
    <BuildingNumber>17a</BuildingNumber>
    <FlatNumber></FlatNumber>
    <City>Warszawa</City>
    <PostCode>03-236</PostCode>
    <PhoneNumber>111222333</PhoneNumber>
    <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
    <SenderFirstName>Adam</SenderFirstName>
    <SenderLastName>Tester</SenderLastName>
    <SenderCompanyName>Test</SenderCompanyName>
    <SenderStreetName>Testowa</SenderStreetName>
    <SenderBuildingNumber>1a</SenderBuildingNumber>
    <SenderFlatNumber>2</SenderFlatNumber>
    <SenderCity>Warszawa</SenderCity>
    <SenderPostCode>00-001</SenderPostCode>
    <SenderPhoneNumber>444555666</SenderPhoneNumber>
    <SenderOrders></SenderOrders>
    <ReturnDestinationCode></ReturnDestinationCode>
    <ReturnEMail></ReturnEMail>
    <ReturnFirstName></ReturnFirstName>
    <ReturnLastName></ReturnLastName>
    <ReturnCompanyName></ReturnCompanyName>
    <ReturnStreetName></ReturnStreetName>
    <ReturnBuildingNumber></ReturnBuildingNumber>
    <ReturnFlatNumber></ReturnFlatNumber>
    <ReturnCity></ReturnCity>
    <ReturnPostCode></ReturnPostCode>
    <ReturnPhoneNumber></ReturnPhoneNumber>
    <ReturnPack></ReturnPack>
    <TransferDescription></TransferDescription>
    <PrintAdress>1</PrintAdress>
    <ReturnAvailable></ReturnAvailable>
    <ReturnQuantity></ReturnQuantity>
    <PrintType>1</PrintType>
</GenerateLabelBusinessPackTwo>
```

4.9.2.2 Response – GenerateLabelBusinessPackTwo

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code (after verification)
<DestinationId>	Pickup point identifier

Field	Description
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)
<NameCL>	ORLEN Paczka logistics center name
<NrCL>	ORLEN Paczka logistics center number
<OriginDestinationCode>	Pickup point code (from the request)
<AutoChangeDestinationConfirm>	Technical field
<LabelData>	Base64-encoded label

RESPONSE EXAMPLE – GenerateLabelBusinessPackTwo

```
<GenerateLabelBusinessPackTwo>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009510304</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <NameCL>NOWY SĄCZ</NameCL>
    <NrCL>4902</NrCL>
    <OriginDestinationCode>NS-358394-N4-35</OriginDestinationCode>
    <AutoChangeDestinationConfirm>0</AutoChangeDestinationConfirm>
</ GenerateLabelBusinessPackTwo>

<LabelData>Base64-encoded label</LabelData>
```

4.9.3 GenerateLabelBusinessPackList

Field	Value
Recommended method	`GenerateLabelBusinessPackListTwo`
Business process	Pre-advice and Label Generation
Usage recommendation	Pre-advice of individual parcels or a list of shipments and creation of a PDF, PDF10, or EPL label.
Additional information	`<PhoneNumber>` and `<SenderPhoneNumber>` – number of characters: 9 char – number without the +48 prefix / 12 char – number with the +48 prefix The `<SenderFirstName>` and `<SenderLastName>` fields are required if no value is provided in the `<SenderCompanyName>` field. The maximum number of labels returned at one time is 50.

4.9.3.1 Request – GenerateLabelBusinessPackList

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<Format>`	Label format	**Yes**	PDF; PDF10; EPL
`<DestinationCode>`	Pickup point code	**Yes**	15 char
`<AlternativeDestinationCode>`	Alternative pickup point code	No	15 char
`<ReturnDestinationCode>`	Return pickup point code	No	15 char
`<BoxSize>`	Parcel size	**Yes**	1 char
`<EMail>`	Recipient email address	No	60 char
`<FirstName>`	Recipient first name	No	30 char
`<LastName>`	Recipient last name	No	30 char
`<CompanyName>`	Recipient company name	No	70 char
`<StreetName>`	Recipient street name	No	30 char
`<BuildingNumber>`	Recipient building number	No	10 char
`<FlatNumber>`	Recipient unit number	No	10 char
`<City>`	Recipient city	No	30 char
`<PostCode>`	Recipient postal code	No	6 char
`<PhoneNumber>`	Recipient phone number	**Yes**	9 char / 12 char
`<SenderEMail>`	Sender email address	**Yes**	60 char
`<SenderFirstName>`	Sender first name	Conditional	30 char

Field	Description	Required	Characters
<SenderLastName>	Sender last name	Conditional	30 char
<SenderCompanyName>	Sender company name	Conditional	70 char
<SenderStreetName>	Sender street name	**Yes**	30 char
<SenderBuildingNumber>	Sender building number	**Yes**	10 char
<SenderFlatNumber>	Sender unit number	No	10 char
<SenderCity>	Sender city	**Yes**	30 char
<SenderPostCode>	Sender postal code	**Yes**	6 char
<SenderPhoneNumber>	Sender phone number	**Yes**	9 char / 12 char
<SenderOrders>	Reference data	No	30 char
<ReturnEMail>	Return email address	No	60 char
<ReturnFirstName>	Return contact first name	No	30 char
<ReturnLastName>	Return contact last name	No	30 char
<ReturnCompanyName>	Return company name	No	70 char
<ReturnStreetName>	Return street name	No	30 char
<ReturnBuildingNumber>	Building number in the return address	No	10 char
<ReturnFlatNumber>	Return unit number	No	10 char
<ReturnCity>	Return city	No	30 char
<ReturnPostCode>	Return postal code	No	6 char
<ReturnPhoneNumber>	Return phone number	No	9 char / 12 char
<PrintAdress>	Select the address to print for a return	**Yes**	1 – sender address; 2 – return address
<ReturnPack>	Return parcel	No	T = return parcel
<TransferDescription>	Reference data	No	70 char
<ReturnAvailable>	Option to send a return parcel	No	T = Yes
<ReturnQuantity>	Number of return parcels	No	1 char
<PrintType>	Label type	**Yes**	1 – full address; 2 – anonymous address; 3 – recipient and sender name only
<ConfirmTermsOfUse>	**Cash on delivery (COD) and additional insurance services have been discontinued in ORLEN Paczka.**		
<ConfirmMarketing>			
<CofirmMailing>			

Field	Description	Required	Characters
<CashOnDelivery>	**Fields and messages related to these services remain in the API contract only for backward compatibility. New integrations should not use these fields.**		
<AmountCashOnDelivery>			
<PackValue>			
<Insurance>			

REQUEST EXAMPLE – GenerateLabelBusinessPackList

```
<GenerateLabelBusinessPackList>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <Format>PDF</Format>
    <BusinessPackList>
        <BusinessPack>
            <DestinationCode>NS-358394-N4-35</DestinationCode>
            <AlternativeDestinationCode></AlternativeDestinationCode>
            <BoxSize>S</BoxSize>
            <EMail>receiver@orlenpaczka.pl</EMail>
            <FirstName>Jan</FirstName>
            <LastName>Nowak</LastName>
            <CompanyName></CompanyName>
            <StreetName>Annopol</StreetName>
            <BuildingNumber>17a</BuildingNumber>
            <FlatNumber></FlatNumber>
            <City>Warszawa</City>
            <PostCode>03-236</PostCode>
            <PhoneNumber>111222333</PhoneNumber>
            <SenderEMail>sender@orlenpaczka.pl</SenderEMail>
            <SenderFirstName>Adam</SenderFirstName>
            <SenderLastName>Tester</SenderLastName>
            <SenderCompanyName>Test</SenderCompanyName>
            <SenderStreetName>Testowa</SenderStreetName>
            <SenderBuildingNumber>1a</SenderBuildingNumber>
            <SenderFlatNumber>2</SenderFlatNumber>
            <SenderCity>Warszawa</SenderCity>
            <SenderPostCode>00-001</SenderPostCode>
            <SenderPhoneNumber>444555666</SenderPhoneNumber>
            <SenderOrders></SenderOrders>
            <ReturnDestinationCode></ReturnDestinationCode>
            <ReturnEMail></ReturnEMail>
            <ReturnFirstName></ReturnFirstName>
            <ReturnLastName></ReturnLastName>
            <ReturnCompanyName></ReturnCompanyName>
            <ReturnStreetName></ReturnStreetName>
            <ReturnBuildingNumber></ReturnBuildingNumber>
            <ReturnFlatNumber></ReturnFlatNumber>
            <ReturnCity></ReturnCity>
            <ReturnPostCode></ReturnPostCode>
            <ReturnPhoneNumber></ReturnPhoneNumber>
            <ReturnPack></ReturnPack>
            <TransferDescription></TransferDescription>
            <PrintAdress>1</PrintAdress>
            <ReturnAvailable></ReturnAvailable>
            <ReturnQuantity></ReturnQuantity>
            <PrintType>1</PrintType>
            <ConfirmTermsOfUse></ConfirmTermsOfUse>
            <ConfirmMarketing></ConfirmMarketing>
```

REQUEST EXAMPLE – GenerateLabelBusinessPackList

```
            <CofirmMailing></CofirmMailing>
        </BusinessPack>
    </BusinessPackList>
</GenerateLabelBusinessPackList>
```

4.9.3.2 Response – GenerateLabelBusinessPackList

Field	Description
<Err>	Error number
<ErrDes>	Error description
<PackCode_RUCH>	Parcel number
<DestinationCode>	Pickup point code (after verification)
<DestinationId>	Pickup point identifier
<PackPrice>	Estimated price per parcel (in grosz)
<PackPaid>	Shipment paid status (true/false)
<NameCL>	ORLEN Paczka logistics center name
<NrCL>	ORLEN Paczka logistics center number
<LabelData>	Base64-encoded label

RESPONSE EXAMPLE – GenerateLabelBusinessPackList

```
<GenerateLabelBusinessPackList>
    <Err>000</Err>
    <ErrDes>saved</ErrDes>
    <PackCode_RUCH>3401009509735</PackCode_RUCH>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <DestinationId>358394</DestinationId>
    <PackPrice>999</PackPrice>
    <PackPaid>true</PackPaid>
    <NameCL>NOWY SĄCZ</NameCL>
    <NrCL>4902</NrCL>
</GenerateLabelBusinessPackList>

<LabelData>Base64-encoded label</LabelData>
```

4.9.4 LabelPrintDuplicate

Field	Value
Recommended method	`LabelPrintDuplicateListTwo`
Business process	Pre-advice and Label Generation
Usage recommendation	Retrieve the label again for a pre-advised parcel in PDF format
Additional information	-

4.9.4.1 Request – LabelPrintDuplicate

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char
<PackCode>	Parcel number	**Yes**	13 char

REQUEST EXAMPLE – LabelPrintDuplicate

```
<LabelPrintDuplicate>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <PackCode>2100123123123</PackCode>
</LabelPrintDuplicate>
```

4.9.4.2 Response – LabelPrintDuplicate

Field	Description
<Err>	Error number
<ErrDes>	Error description
<Label>	Base64-encoded label

RESPONSE EXAMPLE – LabelPrintDuplicate

```
<LabelPrintDuplicate>
    <Err>0</Err>
    <ErrDes>ok</ErrDes>
    <Label>Base64-encoded label</Label>
</LabelPrintDuplicate>
```

4.9.5 LabelPrintDuplicateTwo

Field	Value
Recommended method	`LabelPrintDuplicateListTwo`
Business process	Pre-advice and Label Generation
Usage recommendation	Retrieve the label again for a pre-advised parcel in PDF, PDF10, EPL, or ZPL format
Additional information	-

4.9.5.1 Request – LabelPrintDuplicateTwo

Field	Description	Required	Characters
<PartnerID>	API identifier	**Yes**	10 char
<PartnerKey>	API password	**Yes**	10 char
<Format>	Label format	**Yes**	PDF; PDF10; EPL; ZPL; ZPL_300DPI
<PackCode>	Parcel number	**Yes**	13 char

REQUEST EXAMPLE – LabelPrintDuplicateTwo

```
<LabelPrintDuplicateTwo>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <Format>ZPL</Format>
    <PackCode>2100123123123</PackCode>
</LabelPrintDuplicateTwo>
```

4.9.5.2 Response – LabelPrintDuplicateTwo

Field	Description
<Err>	Error number
<ErrDes>	Error description
<Label>	Base64-encoded label

RESPONSE EXAMPLE – LabelPrintDuplicateTwo

```
<LabelPrintDuplicateTwo>
    <Err>0</Err>
    <ErrDes>ok</ErrDes>
    <Label>Base64-encoded label</Label>
</LabelPrintDuplicateTwo>
```

4.9.6 LabelPrintDuplicateList

Field	Value
Recommended method	`LabelPrintDuplicateListTwo`
Business process	Pre-advice and Label Generation
Usage recommendation	Retrieve labels again for multiple parcels in PDF, PDF10, or EPL format
Additional information	Parcel numbers should be provided in `<string>` elements of the `<PackCodeList>` collection. The maximum number of parcels in the `<PackCodeList>` collection is 50.

4.9.6.1 Request – LabelPrintDuplicateList

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<Format>`	Label format	**Yes**	PDF; PDF10; EPL
`<PackCodeList>`	Parcel number	**Yes**	13 char

REQUEST EXAMPLE – LabelPrintDuplicateList

```
<LabelPrintDuplicateList>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <Format>PDF</Format>
    <PackCodeList>
        <string>3401009545290</string>
        <string>3401009545320</string>
    </PackCodeList>
</LabelPrintDuplicateList>
```

4.9.6.2 Response – LabelPrintDuplicateList

Field	Description
`<Err>`	Error number
`<ErrDes>`	Error description
`<LabelData>`	Base64-encoded label

RESPONSE EXAMPLE – LabelPrintDuplicateList

```
<LabelPrintDuplicateList>
    <PackCode>3401009545290</PackCode>
    <Err>0</Err>
    <ErrDes>saved</ErrDes>
</LabelPrintDuplicateList>
<LabelPrintDuplicateList>
    <PackCode>3401009545320</PackCode>
    <Err>0</Err>
    <ErrDes>saved</ErrDes>
</LabelPrintDuplicateList>
```

4.9.7 CallPickup

Field	Value
Recommended method	`CallPickupNew`
Business process	Ordering a Courier Pickup
Usage recommendation	Ordering a courier pickup from the address assigned to the agreement
Additional information	The pickup address is retrieved automatically from the data assigned to the commercial agreement in the ORLEN Paczka system. The request must include the `<readyDate>` and `<pickupDate>` parameters defining the courier pickup time slot. Available courier arrival time slots are returned in the response from the `GetAvailablePickups` method (**section 4.5.1**). Parcel numbers to be collected should be provided in `<string>` elements of the `<packList>` collection. The time returned in the `<ReadyDate>` and `<PickupDate>` fields is in UTC+1 (local Polish time). A courier pickup can be canceled by contacting an ORLEN Paczka account manager or the ORLEN Paczka helpline and providing the courier pickup number.

4.9.7.1 Request – CallPickup

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char
`<packList>`	Parcel numbers to be collected	**Yes**	13 char
`<readyDate>`	Start date and time when the parcel is ready for collection	**Yes**	YYYY-MM-DDThh:mm:ss
`<pickupDate>`	End date and time when the parcel is ready for collection	**Yes**	YYYY-MM-DDThh:mm:ss

REQUEST EXAMPLE – CallPickup

```
<CallPickup>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
    <packList>
        <string>2100123123123</string>
        <string>2100123123124</string>
    </packList>
    <readyDate>2025-11-26T11:00:00</readyDate>
    <pickupDate>2025-11-26T13:00:00</pickupDate>
</CallPickup>
```

4.9.7.2 Response – CallPickup

Field	Description
<Err>	Error number
<ErrDes>	Error description
<Data>	Courier pickup request number

RESPONSE EXAMPLE – CallPickup

```
<CallPickupResult>
    <Err>0</Err>
    <ErrDes>saved</ErrDes>
    <Data>14682606</Data>
</CallPickupResult>
```

4.9.8 GiveMeAllLocationWithAllData

Field	Value
Recommended method	`GiveMeAllLocationWithAllDataWithZipCode`
Business process	Pickup point search
Usage recommendation	Search for extended data about pickup points in the ORLEN Paczka network (APM, PPP, PKN, and PPK)
Additional information	-

4.9.8.1 Request – GiveMeAllLocationWithAllData

Field	Description	Required	Characters
-	-	-	-

REQUEST EXAMPLE – GiveMeAllLocationWithAllData

```
<tns:GiveMeAllLocationWithAllData
xmlns:tns="https://91.242.220.103/WebServicePwR"/>
```

4.9.8.2 Response – GiveMeAllLocationWithAllData

Field	Description
`<DestinationCode>`	Pickup point code (after verification)
`<StreetName>`	Street name
`<BuildingNumber>`	Building and unit number
`<City>`	City
`<District>`	District
`<Longitude>`	Longitude
`<Latitude>`	Latitude
`<Province>`	Province
`<OpeningHours>`	Pickup point opening hours
`<Location>`	Pickup point location description
`<PSD>`	Unique 6-digit pickup point ID
`<Available>`	Point availability in the service: `T` or `N`
`<Obszar>`	Pickup point area
`<Mikrorejon>`	Pickup point microregion
`<Skrotnrpok>`	Abbreviated pickup point ID number
`<Sortownia>`	Sorting facility assigned to the pickup point
`<PointType>`	Point type (section 2.4.2)
`<Presort>`	Pickup point presort
`<Czas>`	Pickup point availability time parameter
`<NearestPoint>`	Addresses and distances in meters to the three nearest pickup points, determined relative to the original base point

Field	Description
<CashOnDelivery>	**The cash on delivery (COD) service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**

RESPONSE EXAMPLE – GiveMeAllLocationWithAllData

```
<LocationWithAllData>
    <DestinationCode>KL-895926-J2-55</DestinationCode>
    <StreetName>CHROBREGO</StreetName>
    <BuildingNumber>6</BuildingNumber>
    <City>Jędrzejów</City>
    <District>Jędrzejów</District>
    <Longitude>20.2861596</Longitude>
    <Latitude>50.6426837</Latitude>
    <Province>świętokrzyskie</Province>
    <CashOnDelivery>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:00:00-24:00, So:00:00-24:00, Nd:00:00-
24:00</OpeningHours>
    <Location>Automat na terenie biuro ubezpieczeniowe</Location>
    <PSD>895926</PSD>
    <Available>T</Available>
    <Mikrorejon>KLJ2</Mikrorejon>
    <Skrotnrpok>J2-55</Skrotnrpok>
    <Presort>11</Presort>
    <Czas>JJ</Czas>
    <PointType>APM</PointType>
    <NearestPoints>
        <NearestPoint>
            <Name>Jędrzejów Przypkowskiego 41A/1</Name>
            <Distance>142</Distance>
        </NearestPoint>
        <NearestPoint>
            <Name>Jędrzejów Armii Krajowej 1A</Name>
            <Distance>506</Distance>
        </NearestPoint>
        <NearestPoint>
            <Name>Jędrzejów Reymonta 12A</Name>
            <Distance>542</Distance>
        </NearestPoint>
    </NearestPoints>
</LocationWithAllData>
```

4.9.9 GiveMeAllApmLocation

Field	Value
Recommended method	`GiveMeAllLocationWithAllDataWithZipCode`
Business process	Pickup point search
Usage recommendation	Search only for parcel lockers (APM) with an extended location description
Additional information	-

4.9.9.1 Request – GiveMeAllApmLocation

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMeAllApmLocation

```
<GiveMeAllApmLocation>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMeAllApmLocation>
```

4.9.9.2 Response – GiveMeAllApmLocation

Field	Description
`<DestinationCode>`	Pickup point code (after verification)
`<StreetName>`	Street name
`<BuildingNumber>`	Building and unit number
`<City>`	City
`<District>`	District
`<Zipcode>`	Postal code
`<Longitude>`	Longitude
`<Latitude>`	Latitude
`<Province>`	Province
`<OpeningHours>`	Pickup point opening hours
`<Location>`	Pickup point location description
`<PSD>`	Unique 6-digit pickup point ID
`<Available>`	Point availability in the service: `T` or `N`
`<CashOnDelivery>`	**The cash on delivery (COD) service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**

RESPONSE EXAMPLE – GiveMeAllApmLocation

```
<AllApmLocation>
    <DestinationCode>KA-573428-E3-20</DestinationCode>
    <StreetName>PIŁSUDSKIEGO</StreetName>
    <BuildingNumber>8</BuildingNumber>
    <City>Konin</City>
    <District>Konin</District>
    <Zipcode>62-504</Zipcode>
    <Longitude>18.2371520</Longitude>
    <Latitude>52.1986380</Latitude>
    <Province>wielkopolskie</Province>
    <CashOnDelivery>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:00:00-24:00, So:00:00-24:00, Nd:00:00-
24:00</OpeningHours>
    <Location>Automat paczkowy z prawej strony bloku nr 8 ulicy
Piłsudskiego</Location>
    <PSD>573428</PSD>
    <Available>T</Available>
</AllApmLocation>
```

4.9.10 GiveMeAllRUCHLocation

Field	Value
Recommended method	`GiveMeAllLocationWithAllDataWithZipCode`
Business process	Pickup point search
Usage recommendation	Search for extended data about pickup points in the ORLEN Paczka network (APM, PPP, PKN, and PPK)
Additional information	-

4.9.10.1 Request – GiveMeAllRUCHLocation

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMeAllRUCHLocation

```
<GiveMeAllRUCHLocation>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMeAllRUCHLocation>
```

4.9.10.2 Response – GiveMeAllRUCHLocation

Field	Description
`<DestinationCode>`	Pickup point code (after verification)
`<StreetName>`	Street name
`<BuildingNumber>`	Building and unit number
`<City>`	City
`<District>`	District
`<Longitude>`	Longitude
`<Latitude>`	Latitude
`<Province>`	Province
`<OpeningHours>`	Pickup point opening hours
`<Location>`	Pickup point location description
`<PSD>`	Unique 6-digit pickup point ID
`<Available>`	Point availability in the service: `T` or `N`
`<PointType>`	Point type (section 2.4.2)
`<CashOnDelivery>`	**The cash on delivery (COD) service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**

RESPONSE EXAMPLE – GiveMeAllRUCHLocation

```
<AllRUCHLocation>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <StreetName>NAŚCISZOWSKA</StreetName>
    <BuildingNumber>39</BuildingNumber>
    <City>Nowy Sącz</City>
    <District>Nowy Sącz</District>
    <Longitude>20.7060310</Longitude>
    <Latitude>49.6280470</Latitude>
    <Province>małopolskie</Province>
    <CashOnDelivery>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:09:00-18:00, So:  nieczynne,
Nd:  nieczynne</OpeningHours>
    <Location>Furgonetka Regionalna Telewizja Kablowa naprzeciwko składu
materiałów budowlanych</Location>
    <PSD>358394</PSD>
    <Available>T</Available>
    <PointType>PPP</PointType>
</AllRUCHLocation>
```

4.9.11 GiveMeAllRUCHZipcode

Field	Value
Recommended method	`GiveMeAllLocationWithAllDataWithZipCode`
Business process	Pickup point search
Usage recommendation	Search for extended data about ORLEN Paczka network pickup points (APM, PPP, PKN, and PPK), including their postal codes
Additional information	-

4.9.11.1 Request – GiveMeAllRUCHZipcode

Field	Description	Required	Characters
`<PartnerID>`	API identifier	**Yes**	10 char
`<PartnerKey>`	API password	**Yes**	10 char

REQUEST EXAMPLE – GiveMeAllRUCHZipcode

```
<GiveMeAllRUCHZipcode>
    <PartnerID>username</PartnerID>
    <PartnerKey>password</PartnerKey>
</GiveMeAllRUCHZipcode>
```

4.9.11.2 Response – GiveMeAllRUCHZipcode

Field	Description
`<DestinationCode>`	Pickup point code (after verification)
`<StreetName>`	Street name
`<BuildingNumber>`	Building and unit number
`<City>`	City
`<District>`	District
`<Longitude>`	Longitude
`<Latitude>`	Latitude
`<Province>`	Province
`<OpeningHours>`	Pickup point opening hours
`<Location>`	Pickup point location description
`<PSD>`	Unique 6-digit pickup point ID
`<Available>`	Point availability in the service: `T` or `N`
`<PointType>`	Point type (section 2.4.2)
`<Zipcode>`	Postal code
`<CashOnDelivery>`	**The cash on delivery (COD) service has been discontinued in ORLEN Paczka.** **Fields and messages related to the service remain in the API contract only for backward compatibility. New integrations should not use these fields.**

RESPONSE EXAMPLE – GiveMeAllRUCHZipcode

```
<AllRUCHZipcode>
    <DestinationCode>NS-358394-N4-35</DestinationCode>
    <StreetName>NAŚCISZOWSKA</StreetName>
    <BuildingNumber>39</BuildingNumber>
    <City>Nowy Sącz</City>
    <District>Nowy Sącz</District>
    <Longitude>20.7060310</Longitude>
    <Latitude>49.6280470</Latitude>
    <Province>małopolskie</Province>
    <CashOnDelivery>false</CashOnDelivery>
    <OpeningHours>Pn-Pt:09:00-18:00, So:  nieczynne, Nd:
nieczynne</OpeningHours>
    <Location>Furgonetka Regionalna Telewizja Kablowa naprzeciwko składu
materiałów budowlanych</Location>
    <PSD>358394</PSD>
    <Available>T</Available>
    <PointType>PPP</PointType>
    <Zipcode>33-300</Zipcode>
</AllRUCHZipcode>
```

5 Error Messages

The section below lists error messages returned by the ORLEN Paczka API. For each code, it explains what the error means, which area it relates to, and what action must be taken before retrying the request. The table helps distinguish errors related to, among other things, request format, data, authentication, permissions, configuration, and API operation.

Error code	What does it mean?	What should you do?
100	**Request:** Required `PartnerID` field is missing.	Provide the API identifier in the `PartnerID` field.
101	**Request:** Required `PartnerKey` field is missing.	Provide the API password in the `PartnerKey` field.
103	**Request:** Required recipient phone number is missing.	Complete `PhoneNumber` according to the requirements of the method being called.
104	**Request:** Pickup point code `DestinationCode` is missing.	Provide the current identifier returned by the method used to retrieve pickup points.
105	**Request:** Neither the recipient first and last name nor the company name was provided.	Complete `FirstName` and `LastName`, or `CompanyName`.
106	**Request: Parcel number** `PackCode` is missing.	Provide the parcel number required by the method being called.
107	**Request:** `PrintAdress` field specifying the address printed for a return is missing.	Set `PrintAdress` to `1` for the sender address or `2` for the return address.
108	**Request:** Required email address is missing.	Complete the Email field indicated by the method being called.
111	**Request:** Sender email address is missing.	Complete `SenderEMail`.
112	**Request:** Sender phone number is missing.	Complete `SenderPhoneNumber`.
113	**Request:** Sender city is missing.	Complete `SenderCity`.
114	**Request:** Sender street name is missing.	Complete `SenderStreetName`.
115	**Request:** Sender building number is missing.	Complete `SenderBuildingNumber`.
116	**Request:** Sender postal code is missing.	Complete `SenderPostCode`.
117	**Request:** Neither the sender first and last name nor the company name was provided.	Complete `SenderFirstName` and `SenderLastName`, or `SenderCompanyName`.

Error code	What does it mean?	What should you do?
120	**Discontinued service:** `AmountCashOnDelivery` amount is missing. The code remains for backward compatibility.	Remove COD-related fields from new calls. The COD service has been disabled.
121	**Request:** `TransferDescription` field required in this scenario is missing.	Complete the field according to the method conditions or remove the parameters that enable this scenario.
123	**Request:** City is missing from the return address.	Complete `ReturnCity` if the request contains return data.
124	**Request:** Street name is missing from the return address.	Complete `ReturnStreetName` if the request contains return data.
125	**Request:** Building number is missing from the return address.	Complete `ReturnBuildingNumber` if the request contains return data.
126	**Request:** Postal code is missing from the return address.	Complete `ReturnPostCode` if the request contains return data.
127	**Request:** Company name is missing from the return data.	Complete `ReturnCompanyName` according to the requirements of the scenario.
128	**Request:** `ReturnAvailable` field is missing.	Set `ReturnAvailable` to T if required by the scenario.
129	**Request:** Number of return parcels `ReturnQuantity` is missing.	Set `ReturnQuantity` if a return should be available for the shipment.
130	**Discontinued service:** `AmountCashOnDelivery` contains invalid characters. The code remains for backward compatibility.	Remove COD-related fields from the request.
131	**Data:** `TransferDescription` contains invalid characters.	Remove unsupported characters and do not exceed the 70-character limit.
132	**Request format:** At least one value has an invalid format.	Compare the field format and order with the WSDL and check `ErrDes`.
133	**Data:** `PhoneNumber` has an invalid format.	Provide 9 digits without the +48 prefix or 12 characters with the +48 prefix.
134	**Discontinued service:** `PackValue` has an invalid value. The code relates to additional insurance.	Remove Insurance and `PackValue`, or set Insurance to F and provide `PackValue` in grosz, e.g. PLN 32,99 = 3299
135	**Discontinued service:** `PackValue` exceeds the historical insurance limit.	Remove the additional insurance fields from the request or provide the value in grosz, e.g. PLN 32,99 = 3299

Error code	What does it mean?	What should you do?
136	**Data:** `TransferDescription` has an invalid value or length.	Check the field value and limit it to 70 characters.
137	**Data:** `ReturnQuantity` has an invalid value.	Provide a value that fits in one character and is valid for the supported return scenario.
138	**Data:** Postal code has an invalid format.	Provide the six-character `NN-NNN` format, e.g. 03-236.
139	**Request:** `PrintType` field is missing.	Set `PrintType` to `1`, `2`, or `3` according to the required data scope on the label.
140	**Discontinued service:** `AmountCashOnDelivery` amount is zero.	Remove COD-related fields from the request.
141	**Data:** `BoxSize` contains an unsupported parcel size.	Set `BoxSize` to S, M, or L.
142	**Data:** `SenderPhoneNumber` has an invalid format.	Provide 9 digits without the +48 prefix or 12 characters with the +48 prefix.
143	**Data:** An unsupported label format was selected.	Use a format supported by the method: `PDF`, `PDF10`, `EPL`, `ZPL`, or `ZPL_300DPI`.
144	**Data:** Parcel number has an invalid format.	Provide the 13-character parcel number returned by the API.
150	**Limit:** The list exceeds the maximum number of parcels supported in a single label generation call.	Split the list into requests containing no more than 50 parcels.
161	**Request:** Required City is missing.	Complete City according to the requirements of the method being called.
162	**Request:** NIP number is missing.	Complete Nip if it is required in the supported process.
163	**Request:** Neither `PackCode` nor `PhoneNumber` was provided.	Provide at least one of these fields.
164	**Request:** Number of parcel numbers to reserve is missing.	Set `NumberOfWaybills` to a value from 1 to 10 000.
191	**Configuration:** The specified postal code is not supported in this process.	Check the postal code and service availability, in particular using `GetAvailablePickups`.
201	**Resource state:** The parcel has already been canceled.	Do not retry the cancellation. Retrieve the current parcel status.
202	**Resource state:** The current parcel state no longer allows cancellation.	Retrieve the parcel status and continue processing according to its current state.
205	**Data:** The API does not recognize the provided `PackCode`.	Check the parcel number and the environment in which it was created.

Error code	What does it mean?	What should you do?
206	**Data:** The API does not recognize `DestinationCode`.	Retrieve the current point list and provide the point identifier.
207	**Data:** The API does not recognize `ReturnDestinationCode`.	Retrieve the current return point identifier and correct the request.
208	**Resource state:** A customer with the provided data already exists.	Do not create the customer again. Use the existing customer identifier.
209 / 243	**Data:** `PackCode` is invalid.	Check the format, source, and value of the parcel number.
211	**Configuration:** Routing data required to prepare the label is missing.	Check the address data and destination point. If they are correct, report the missing configuration to your ORLEN Paczka account manager.
212	**Permissions:** The account does not have access to the specified label number.	Use a number created for this `PartnerID` or verify the account permissions.
213	**Resource state:** The parcel with this number has already been submitted to the ORLEN Paczka system.	Do not submit the pre-advice data again. Retrieve the current parcel status.
214 / 223	**Permissions:** The parcel does not belong to the authenticated partner.	Check whether the parcel number was created under the same `PartnerID`.
222	**Data:** The parcel was not found in the ORLEN Paczka system.	Check the parcel number, environment, and stage at which the parcel was submitted to the system.
240	**Limit:** The allowed number of returns has already been used for the parcel.	Do not generate another return for this parcel. Check the existing returns and `ReturnQuantity`.
250	**Request:** `PSD` point identifier is missing.	Complete `PSD` if you are using a method that requires this identifier.
251	**Data:** The API does not recognize the `PSD` identifier.	Retrieve the current point data and correct the identifier.
252	**Data:** The API does not recognize `ReturnPSD`.	Retrieve the current return point data and correct the identifier.
253	**Data:** The API does not recognize `AlternativePSD`.	Retrieve the current point data and correct the alternative identifier.
254	**Data:** The API does not recognize `AlternativeDestinationCode`.	Retrieve the current point list and provide the returned identifier.
300	**API system:** The data could not be saved.	Check `ErrDes` and the operation result before retrying the request, if necessary.

Error code	What does it mean?	What should you do?
308	**Permissions:** The account does not have the permission required by the called operation.	Check with your account manager whether the service is active for the given `PartnerID`.
309	**Permissions:** The account cannot create customers.	Obtain approval to use the `CreateCustomer` method or do not call this operation.
310	**Discontinued service:** The account does not have COD permission.	Remove COD-related fields; the service has been disabled.
311	**Discontinued service:** The account does not have permission for additional insurance.	Remove Insurance and `PackValue`, or set Insurance to F and declare `PackValue` in grosz, e.g. PLN 32,99 = 3299
312	**Permissions:** The account does not have the returns service enabled.	Before using return methods, have the service enabled for the account.
314	**Permissions:** The account is not allowed to create label-free shipments.	Do not use label-free shipping without the appropriate configuration with your ORLEN Paczka account manager.
318	**Discontinued service:** The point specified in `DestinationCode` does not support COD.	Remove COD-related fields; the service has been disabled.
330 / 336	**API system:** The label could not be generated.	Check `ErrDes`, the label format, and the shipment data.
350	**Resource state:** The shipping code does not exist, is inactive, has been used, or is outside its active period.	Check the code status. Generate a new shipping code if necessary.
399	**Data:** The specified parcel was not found.	Check the number and environment; tracking methods do not return statuses older than 90 days.
400	**Limit:** The requested pool of parcel numbers exceeds the method limit.	Set `NumberOfWaybills` to no more than 10 000 parcels.
401	**Authentication:** `PartnerID` or `PartnerKey` is invalid.	Check the credentials and environment. Take into account any password change in WebTrucker.
801	**Request:** No parcels were provided for the shipping protocol.	Add at least one parcel number to the parcels collection.
802	**Limit:** The number of parcels in the protocol exceeds 571.	Split the parcels across several protocols.
901	**API system:** An unspecified error occurred.	Check the call context and, if you cannot identify the error, report it to ORLEN Paczka.

Error code	What does it mean?	What should you do?
999	**API system:** The API is temporarily unable to confirm the account permissions.	Retry the request after a short time.
1038	**Request:** `PartnerName` is empty or `NULL`.	Provide the sender company name in `PartnerName`.
1039	**Request:** Street is empty or `NULL`.	Provide the street name in `Street`.
1040	**Request:** City is empty or `NULL`.	Provide the city in `City`.
1041	**Request:** `PostCode` is empty or NULL.	Provide the postal code in `PostCode`.
1043	**Request:** `EMail` is empty or `NULL`.	Provide the email address in `Email`.
1052	**Request:** The end of the pickup time slot is not set in `PickupDate`.	Set `PickupDate` to a value from an available pickup time slot.
1053	**Request:** The start of the pickup time slot is not set in `ReadyDate`.	Set `ReadyDate` to a value from an available pickup time slot.
1054	**Data:** `PickupDate` falls on a Sunday.	Select another day returned by `GetAvailablePickups`.
1055	**Data:** The end of the pickup time slot is earlier than its start.	Set the end time later than `ReadyDate`.
1067	**Data:** The pickup time slot does not meet the timing requirements for the selected day.	Use the times returned by `GetAvailablePickups` and keep the `MinimumInterval`.
1077	**Data:** The pickup end time exceeds the latest available time on the selected day.	Set `PickupDate` no later than `MaxPickupDate` returned by `GetAvailablePickups`.
1084	**Availability:** Courier pickup is not available on the selected day.	Retrieve current dates using `GetAvailablePickups` and select an available day.

6 Warning Messages Indicating Successful Pre-advice Generation

Messages `006`, `007`, or `008` in the `<Err>` field indicate that the point code was changed automatically, for example when a universal pickup point code is provided.

However, these are not errors that prevent label generation. Even if they occur, the label is generated correctly and the `<DestinationCode>` field is populated with the current value identifying the pickup point.

Message code	Message description
000	Saved
006	Saved, but DestinationCode was changed
007	Saved, but ReturnDestinationCode was changed
008	Saved, but DestinationCode and ReturnDestinationCode were changed

7 Parcel Statuses

The section below contains shipment statuses returned by the ORLEN Paczka API tracking methods. The status code is returned in the `<Trans>` field and its description in the `<Trans_Des>` field. In methods that return full history, the `<Attribute>` field provides additional status context, such as the standard shipment flow, return to the sending point, or consumer return. The same code may occur with different attributes, so both fields must be considered when interpreting the history.

7.1 Status Response Fields

Methods used to track parcel statuses return the status code, its description, and the date and time when the event was recorded. Depending on the selected method, the response may also include an attribute specifying the status context and additional information about the shipment and pickup point. The table below explains the most important status response fields.

Field	Meaning
`Trans`	Numeric parcel status code
`Trans_Des`	Description of the status corresponding to the `<Trans>` value
`Attribute`	Additional status context, e.g. return type or point change. The field may be empty.
`Data`	Date and time when the parcel received the status.

7.2 Status Attributes

The `<Attribute>` field adds information about the process the parcel is currently in, such as the standard flow, return to the sending point, or consumer return. The attribute is not a separate status and must be interpreted together with the code returned in the `<Trans>` field. The field may be empty if the status does not require additional context.

Attribute value	Meaning
`brak` / `NULL`	The status relates to the standard parcel flow or does not require additional context.
`POWROT`	Parcel return to the sending point (the recipient did not collect the parcel).
`2_POWROT`	Parcel return to the sending warehouse (the recipient did not collect the parcel).
`ZWROT`	Return parcel sent to the sender warehouse (consumer return).
`MOBILNY`	The status relates to a mobile pickup point.
`ALTERNATYWNY`	The status relates to an alternative pickup point.
`ZMIENIONY`	The status relates to a changed pickup point.

7.3 Parcel Status Table

The table below contains the codes returned in the `<Trans>` field, their descriptions, and their business meaning. When interpreting a status, also consider the value of the

`<Attribute>` field because the same code may occur in the standard parcel flow and in return processes.

Status code	Status	What it means	Attributes
100	At regional sorting facility	The parcel was accepted at the branch.	brak, 2_POWROT, POWROT, ZWROT
110	In transit from the branch to SC	The parcel is being transported from the branch to the central sorting facility.	brak, 2_POWROT, POWROT
120	In transit from the branch to DP	The parcel is being transported from the branch to the pickup point.	brak, 2_POWROT, POWROT
193	Redirected to APM	The parcel was redirected to a parcel locker.	brak
195	Redirected to pickup point	The parcel was redirected to a staffed pickup point.	brak
200	Pre-advised to PwR	The parcel data has been registered in the system, but this status does not yet confirm that ORLEN Paczka has physically accepted the parcel.	brak, 2_POWROT, POWROT, ZWROT
201	Pre-advice canceled	The parcel pre-advice was canceled.	brak, 2_POWROT, POWROT, ZWROT
210	Sent at pickup point	The parcel was sent at a staffed pickup point or parcel locker.	brak, ZWROT
230	In transit from pickup point to branch	The parcel was prepared for collection by the courier.	brak, ZWROT
240	In transit with courier to branch	The parcel was collected by the courier.	brak, 2_POWROT, POWROT, ZWROT
241	In transit after storage	The parcel was forwarded for further transport after storage.	brak, POWROT
300	At central sorting facility	The parcel is at the central sorting facility.	brak, 2_POWROT, POWROT, ZWROT
400	At central sorting facility	The parcel is at the central sorting facility.	brak, 2_POWROT, POWROT, ZWROT

Status code	Status	What it means	Attributes
401	At central sorting facility	The parcel is at the central sorting facility.	brak, 2_POWROT, POWROT, ZWROT
450	In transit from SC to branch	The parcel is being transported from the central sorting facility to the branch.	brak, 2_POWROT, POWROT
610	Handed over to DP courier for delivery	The parcel was handed over to the courier for delivery.	brak, 2_POWROT, POWROT, ZWROT
620	Ready for pickup	The parcel was accepted at the pickup point and is waiting to be collected by the recipient.	brak, 2_POWROT, POWROT
653	At branch	The parcel was accepted at the branch.	brak, 2_POWROT, POWROT, ZWROT
660	In transit with courier to pickup point	The parcel is being transported by the courier to the pickup point.	brak, 2_POWROT, POWROT, ZWROT
665	Handed over to pickup point	The parcel was handed over to a staffed pickup point.	brak, 2_POWROT, POWROT
670	Handed over to DP	The parcel was handed over to a staffed pickup point.	brak, 2_POWROT, POWROT, ZWROT
677	Pickup point closed	The parcel could not be handed over to a staffed pickup point or parcel locker because the point was closed.	brak, 2_POWROT
679	Not handed over to courier	The parcel was not handed over to the courier for delivery.	brak, 2_POWROT, POWROT
680	In transit to pickup point	The parcel was handed over to the courier and is being transported to a staffed pickup point or parcel locker.	brak, 2_POWROT, POWROT, ZWROT
681	Sent for storage	The parcel was sent for storage.	brak, POWROT
685	In transit to pickup point (DP)	The parcel is being transported by the	brak, 2_POWROT, POWROT

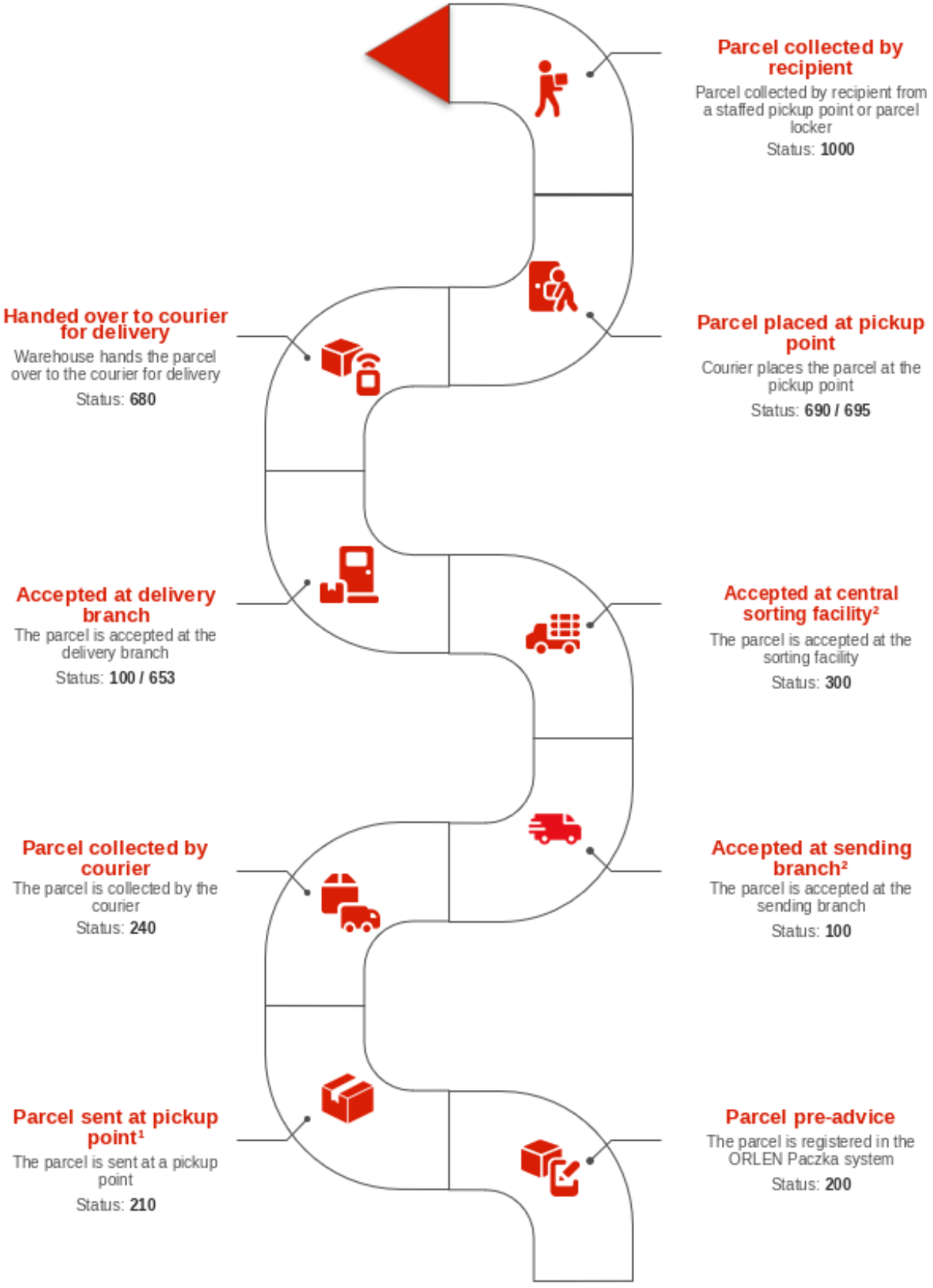
Status code	Status	What it means	Attributes
		courier to the pickup point.	
690	At pickup point	The parcel was accepted at a staffed pickup point or parcel locker and is waiting to be collected by the recipient.	brak, 2_POWROT, POWROT, ZWROT
691	Stored at pickup point	The parcel is being stored at the pickup point.	brak, POWROT
695	At pickup point – SMS sent	The parcel was accepted at a staffed pickup point or parcel locker and is waiting to be collected by the recipient.	brak, 2_POWROT, POWROT
696	Stored at pickup point – SMS sent	The shipment is being stored at the pickup point.	brak, POWROT
700	At branch for transport to SC	The parcel was sent back to the sender.	brak, 2_POWROT, POWROT
708	Stored at pickup point – not collected	The parcel was not collected while it was stored at the pickup point.	brak, POWROT
709	Return – not collected on time	The parcel will be returned to the sender because it was not collected from the pickup point on time.	brak, POWROT
710	Lost / being searched for	The parcel was marked as being searched for.	brak
711	Parcel under verification	The parcel was sent for verification.	brak
712	On hold	Parcel delivery was put on hold.	brak
714	POK closed	The parcel could not be handed over to the pickup point because the point was closed.	brak, 2_POWROT, POWROT
729	Return – invalid pickup point	The parcel will be returned because the pickup point is invalid.	brak, 2_POWROT, POWROT

Status code	Status	What it means	Attributes
739	Not handed over to pickup point	The parcel was not handed over to a staffed pickup point or parcel locker.	brak, 2_POWROT, POWROT
749	Complaint	A complaint was registered for the parcel.	brak
750	Return to sender	The parcel was sent back to the sender.	brak, 2_POWROT, POWROT
780	Delivery to DP not possible	The parcel cannot be placed at the pickup point.	brak
790	Return to branch	The parcel return was sent to the branch.	brak, 2_POWROT, POWROT
800	Return to sorting facility	The parcel return was sent to the sorting facility.	brak
888	Archived	The parcel data was moved to the system archive.	brak, 2_POWROT, POWROT, ZWROT
900	Return to sender	The parcel was sent back to the sender.	brak, 2_POWROT, POWROT, ZWROT
999	Damaged – lost	The parcel was marked as damaged or lost.	brak, 2_POWROT, POWROT, ZWROT
1000	Collected by customer	The parcel was collected by the recipient from a staffed pickup point or parcel locker.	brak, 2_POWROT, POWROT
1100	Collected	The parcel was collected during the return process.	POWROT
1200	Collected – return	The return parcel was collected.	ZWROT
1220	Return to sender	The parcel return was collected by the sender.	2_POWROT, ZWROT
2000	Disposal	The parcel was marked as damaged or lost.	brak

7.4 Standard Parcel Lifecycle – Status Visualization

The diagram below shows the standard successful path of an ORLEN Paczka parcel – from sending to collection by the recipient.

The process may vary depending on the sending location or parcel route, and other scenarios not shown in the diagram may also occur, for example failure to collect the parcel from the pickup point on time. This means that each parcel may have a different history.



[1] This stage occurs only when the parcel is collected from the sender's address, not from a pickup point.
[2] Statuses may occur depending on the logistics process and the location of the sending and delivery branches.

8 Change History

The section below lists changes made to this documentation together with their publication dates. This makes it possible to track what changes were introduced and when a given document version was in effect.

Data	Version	Change description
10.2026	v_1_26_007_EN	Addition of the <PackCode> field to the response parameters of the GenerateShippingCode method
09.2026	v_1_26_006_EN	Update to the description of the CreateBrokerBranch method
08.2026	v_1_26_005_EN	Updated the document template and aligned the descriptions with the current API.
08.2026	v_1_26_004_EN	Added the CreateBrokerBranch method
07.2026	v_1_26_003_EN	Updated information about generating a label in ZPL_300DPI format using the GenerateLabelBusinessPackListTwo method
05.2026	v_1_26_002_EN	Added the GenerateBrokerReturn2Home and GenerateBrokerShippingCodeReturn2Home methods
03.2026	v_1_26_001_EN	Added the ORLENPaczkaWeekendOpenPUDOPoints method
11.2025	v_1_25_008_EN	Minor text corrections.
10.2025	v_1_25_007_EN	Updated sections 3.8 and 3.14 (insurance service).
09.2025	v_1_25_006_EN	Updated method descriptions for consumer return shipments.
03.2025	v_1_25_005_EN	Updated descriptions of the GiveMePackStatus, GiveMePackStatusList, GiveMePackStatusFullHistory, and GiveMePackStatusFullHistoryList methods.
03.2025	v_1_25_004_EN	Added statuses used during temporary storage (241, 681, 691, 696, 708).
02.2025	v_1_25_003_EN	Added a description of the <BuildingNumber> tag in methods that return ORLEN Paczka pickup point data. This tag contains the building number and unit number (if applicable) of the pickup point.
01.2025	v_1_25_002_EN	Added the ORLEN Paczka Track&Trace link specification.
01.2025	v_1_25_001_EN	Added parcel size specifications for the ORLEN Paczka service (S, M, L).
11.2024	v_1_24_002_EN	Added information about disabling the COD (Cash On Delivery) service.
10.2024 v_1_24_001_EN		Updated the “Introduction” section. Updated method descriptions and their parameters. Added the GiveMeAllLocationWithAllDataWithZipcode method, which returns a list of points with extended descriptions and postal codes. Added the GenerateCustomerReturn method.

Data	Version	Change description
		Added the GenerateCustomerReturnShippingCode method. Added the GenerateStandardCustomerReturn method. Added the GenerateStandardCustomerReturn method.
31.08.2023	v_1_23_003_EN	Added the GiveMeAllLocationWithAllData method, which returns a list of points with extended descriptions.
01.07.2023	v_1_23_002_EN	Added the GetAvailablePickups method, which returns a list of ordered pickups.
12.04.2023	v_1_23_001_EN	Added the GiveMeShippingCodeStatusFullHistory and PutCustomerShippingCodeCanceled methods.
16.01.2023	v_1_22_003_EN	Added the CallPickup pickup method for ordering a courier pickup.
14.07.2022	v_1_22_002_EN	Added a new pickup point type <PointType> PPK that accepts S, M, and L parcel sizes. PPK points do not support sending parcels. Information that they support collection only is returned in the Location Description <Location>.
25.05.2022 v_1_22_001_EN		Added the new GiveMeAllRUCHWithFilledD1 method, which retrieves a list of points that meet D+1 delivery criteria. Added the new Ping method, which checks the connection status with the API web service. Updated the “Introduction” section with a description of the return label generation procedure.
13.10.2021 v_1_16_007_A		Added the new GiveMeAllAPMLocation method, which retrieves a list of all parcel lockers available in the ORLEN Paczka service. Added new point types to the GiveMeAllRUCHLocation and GiveMeAllRUCHWithFilled methods in <pointtype>: APM (parcel lockers) and PKN (ORLEN fuel stations). Added new S, M, and L parcel sizes to the <BoxSize> parameter for the new APM and PKN point types.