



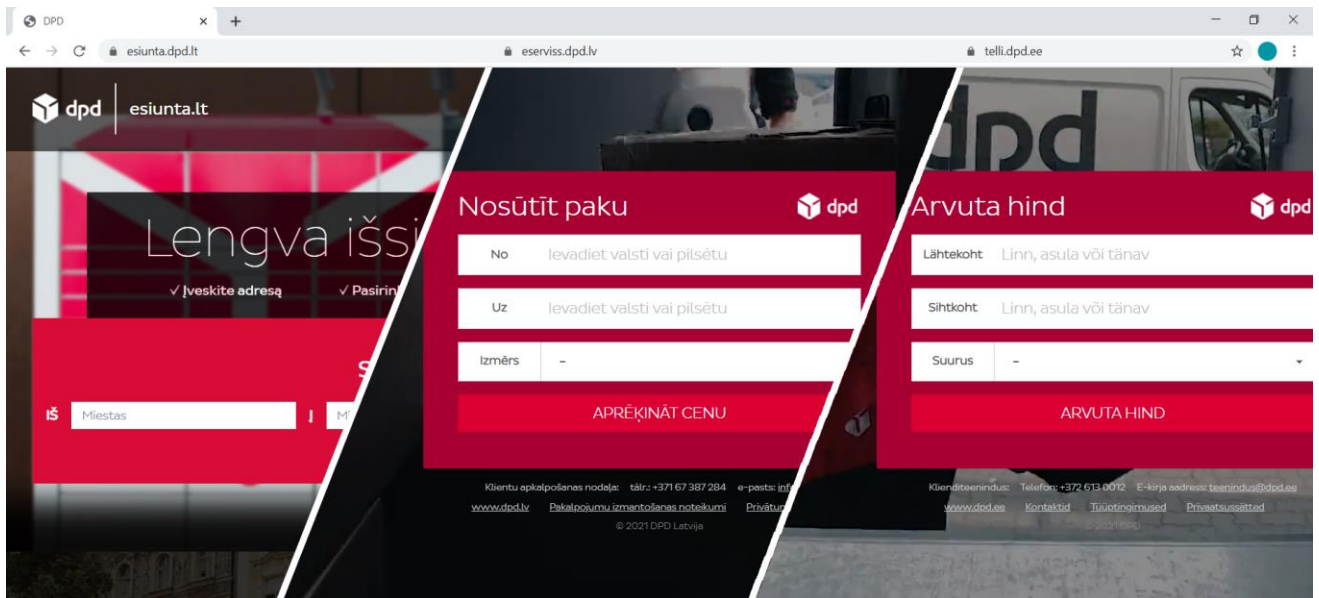
# DPD API documentation

v.1.2.1, March, 2023

1.	Introduction .....	3
2.	Overview of the web service.....	4
3.	DPD services and service restrictions .....	6
3.1.	The main services.....	6
3.2.	Additional services.....	8
4.	Authorization management.....	12
4.1.	Token processing by GUI .....	12
4.2.	Token processing by API request.....	13
4.2.1.	Authorization token creation .....	13
4.2.2.	Authorization token list retrieval.....	14
4.2.3.	Authorization token deletion .....	15
5.	Shipment sending process .....	16
5.1.	Shipment creation.....	16
5.2.	Label creation .....	24
5.3.	Courier request .....	26
6.	Parcel tracking .....	29
6.1.	On demand .....	29
6.1.1.	Details array for basic data (detail: 0).....	30
6.1.2.	Parcel status messages.....	31
6.1.3.	Details array for advanced data (detail: 3).....	31
6.1.4.	Status codes / service codes .....	32
6.1.5.	Additional codes .....	35
6.2.	Call-back request.....	36
7.	Additional functionality .....	37
7.1.	Service list retrieval.....	37
7.2.	Pickup point list retrieval.....	39
7.3.	Payer list retrieval .....	43
7.4.	Shipment list retrieval .....	44
7.5.	Shipment deletion .....	46
7.6.	Courier request time frame retrieval .....	47
8.	Pricing / invoicing.....	49
9.	Frequently asked questions (FAQ).....	49
10.	Response errors .....	51
10.1.	Knowledge base .....	51
11.	Contact information.....	52
12.	Examples .....	53
12.1.	DPD Classic shipment creation request .....	53
12.2.	B2C + COD shipment creation request (incl. label request).....	54
12.3.	B2C + Evening shipment creation request .....	55
12.4.	PDF A4 label creation request.....	56
12.5.	Courier request .....	56

# 1. Introduction

API is a REST-based solution that is a part of DPD labelling system and can be used to automate processes of the DPD web portals ([esiunta.dpd.lt](http://esiunta.dpd.lt), [eserviss.dpd.lv](http://eserviss.dpd.lv), [telli.dpd.ee](http://telli.dpd.ee)):



For easier understanding of the documentation, we have divided the methods into 3 parts:

- Authorization management (Chapter 4) will contain all the methods needed for managing authorization tokens.  
As the knowledge of tokens will be needed further, please read this chapter first.
- Shipment sending (Chapter 5) will contain commonly used methods for creating and sending shipments.  
Please take into account that shipment creation depends on the service type you're planning to use, therefore we have added short service and additional service descriptions in Chapter 3. On the right side of each service, we have listed request blocks that shall be used for a specific service.
- Additional functionality (Chapter 6) – here you will find methods that support the main functionality of the API (such as retrieving Pickup point list). The methods are intended for closer integration into the DPD system.

There are multiple options on how the API methods can be used, hence in the Chapter 9 "Frequently asked questions (FAQ)" we have listed some of the ideas that may be useful to read before starting an integration.

To keep continuity and track changes the documentation has version number assigned. The number is built according to the following principles: x1.x2.x3

- x1 – web service version (new endpoint)
- x2 – web service subversion (new methods, new parameters)
- x3 – documentation description or formatting changes

Here are SWAGGER links available that can also be used:

- LT: <https://esiunta.dpd.lt/api>
- LV: <https://eserviss.dpd.lv/api>
- EE: <https://telli.dpd.ee/api>

## 2. Overview of the web service

The key point of the DPD API:

- API can be used only by DPD contracted clients!
- API uses UTF-8 encoding.
- HTTP headers are meaningful.
- Accept header “application/json” should be used unless listed differently on a specific method.
- Standard HTTP response codes are returned in response, matching the type of response. Response HTTP code contains information about the request status:
  - 200 – Request has been processed correctly
  - 201 – Request has been processed correctly
  - 204 – Request has been processed correctly, there will be no data in response body
  - 206 – Request has been processed partially, response body should be checked
  - 400 – Request has not been processed (bad request), request data should be checked
  - 401 – Request has not been processed (unauthorized), authorization token should be checked!
  - 403 – Request has not been processed (forbidden)
  - 404 – Request has not been processed (not found)
  - 422 – Request has not been processed (unprocessable entity)
  - 429 – Request has not been processed (too many requests)
- All methods (except authorisation token generation method) must contain bearer authentication key that can be requested from the system (chapter **Error! Reference source not found.**).

- HTTP verbs are used to denote the type of operations:
  - search (GET)
  - create (POST)
  - delete (DELETE)
- Parameters must be provided based on HTTP method:
  - GET: Parameters must be provided within URL as standard GET parameter (`http://site.name?parameter=value`)
  - DELETE: Parameters must be provided within URL as part of it (`http://site.name/value`)
  - POST: Parameters must be provided within request body as JSON data entity in a structure and format defined by API. The API uses JSON bodies.
- Responses can contain JSON response in HTTP body.



Environments:

	Live environment	Test environment
Lithuania	<code>https://esiunta.dpd.lt/api/v1</code>	<code>https://sandbox-esiunta.dpd.lt/api/v1</code>
Latvia	<code>https://eserviss.dpd.lv/api/v1</code>	<code>https://sandbox-eserviss.dpd.lv/api/v1</code>
Estonia	<code>https://telli.dpd.ee/api/v1</code>	<code>https://sandbox-telli.dpd.ee/api/v1</code>

How to receive access:

Once your contract is signed, you will need to register in the specific environment. When this is done, please contact DPD support, who will assign the contract to your profile. From the moment the contract is assigned, you will be able to use API according to your agreement. As soon as this is done, you're ready to go!

## 3.DPD services and service restrictions

DPD separates all the services into two groups: main services and additional services. Main services define shipment specifics and principles of the delivery process. Additional services provide additional functionality/service to the specific main service (it can change some parts of the main service process).

There can be some shipment packaging and labelling regulations related to specific main and/or additional services, therefore packaging and labelling guide should be read before shipment creation.

### 3.1. The main services

#### **DPD Classic (B2B) service**

This is standard DPD to-door delivery service for business-to-business deliveries. This service can be enriched by any additional services. Delivery can be provided to all EU countries, Ukraine, Norway and Switzerland.

EE: Delivery to Ukraine is not available.

Shipment creation blocks needed for this service

- Sender block
- Receiver block
- Service block
- Parcels / pallets blocks

Example: chapter 12.1

#### **DPD Private (B2C) service**

B2C service is a DPD to-door delivery for business-to-consumer deliveries. This service includes informing the consignee about the delivery (predict SMS). Depending on your business specifics, you can choose additional services to make delivery as convenient as possible for you and your consignee.

LV: For shipments to Greece, Bulgaria, Norway or Switzerland – DPD Classic should be used instead of DPD Private service. For shipments to Finland and Sweden - Pickup service should be used instead of DPD Private service.

Shipment creation blocks needed for this service

- Sender block
- Receiver block (to-door)
- Service block
- Parcels / pallets blocks

Example: chapter 12.2

EE: For shipments to Greece DPD Classic should be used instead of DPD Private service.

LT: For shipments to Greece and Bulgaria DPD Classic should be used instead of DPD Private service.

### **DPD Pickup service**

This is a delivery to a Pickup point (parcel locker or parcelshop). There are size and weight restrictions for this service. A multiparcel shipments will be processed as separate shipments!

To get more detailed information regarding these limitations please check information on the local DPD website or contact the local DPD sales department.

### **DPD Pickup return service**

The service provides a return shipment label that can be used for returning a shipment back to sender. The list of countries where the service is available is limited. For more details, please check local DPD website.

The returning shipment must be dropped to any DPD Pickup point and it will be delivered with courier to return address.

This service can be used for returning shipments within the same country as well. Returns with DPD Classic and DPD Private services are also possible, please contact the local DPD sales department for price evaluation.

### **Collection request**

This service is used to collect parcels and pallets abroad and delivered to your local country.

Shipment creation blocks needed for this service

- Sender block
- Receiver block (to Pickup point)
- Service block
- Parcels / pallets blocks

Shipment creation blocks needed for this service

- Sender block
- Receiver block (to-door)
- Service block
- Parcels / pallets blocks

Shipment creation blocks needed for this service

- Sender block
- Receiver block (to-door)
- Service block
- Parcels / pallets blocks
- Pickup block

Collection from Greece is not provided. Pallets can only be collected from Latvia, Lithuania, Estonia, Poland, Denmark, Finland and Sweden.

### **Saturday delivery**

This service provides shipment delivery on Saturday.

Shipment creation blocks needed for this service

- Sender block
- Receiver block (to-door)
- Service block
- Parcels / pallets blocks

### **Tyre service**

This service must be used for sending tyres.

Shipment creation blocks needed for this service

- Sender block
- Receiver block (to-door)
- Service block
- Parcels / pallets blocks

## 3.2. Additional services

### **Cash on delivery (COD)**

This additional service allows recipient to pay with a payment card for a post-paid shipment delivered by courier (DPD Classic/DPD Private) or through DPD Pickup network in Latvia, Lithuania and Estonia.

In addition to main service blocks, these additional blocks are needed:

- Additional service block

Example: chapter 12.2

### **Return of documents (ROD)**

This additional service covers the signing of a document attached to a shipment handed over for delivery on behalf of the

In addition to main service blocks, these additional blocks are needed:



recipient and the return of the relevant copies of a document (delivery note, contract, passport copy) to the sender according to instructions. The content must be agreed with DPD in advance.

In case of this service shipment/parcel reference is overwritten by ROD reference. For LV/LT – shipment's 2<sup>nd</sup> reference is overwritten, for EE – parcel's 3<sup>rd</sup> reference is overwritten. Please do not use these references for other purpose!

### **SWAP**

This additional service is used to exchange parcels. Courier will hand out a parcel(s) if the same number of parcel(s) is given back to a courier.

### **ID check**

This additional service is used to identify the recipient on delivery. The parcel can only be delivered to the recipient stated on the parcel label.

### **Complete delivery**

This additional service ensures that all the parcels and/or pallets of one shipment are delivered at the same time.

### **Loading service**

A.k.a. "two-man delivery service", "4hands"

This additional service is for large cargo delivery which requires additional person to ensure unloading of parcels or pallets over 31.5kg (one unit cannot exceed the weight of 80kg).

- Additional service block

Shipment creation blocks that are needed for this service

- Return block
- Additional service block

In addition to main service blocks, these additional blocks are needed:

- Additional service block

In addition to main service blocks, these additional blocks are needed:

- Additional service block

In addition to main service blocks, these additional blocks are needed:

- Additional service block

To get more information please check your local DPD website or contact the local DPD sales department.

EE: The highest floor for parcel to be delivered in case if there is no elevator is the fifth floor.

### **Evening delivery**

This additional service ensures that parcels are delivered on evening within the timeframe that is defined.

To receive information about defined timeframes in specific country, please contact DPD. Timeframe must be provided within format: hh:mm-hh:mm

### **Department delivery service**

The service should be used if total weight of a shipment is more than 31.5 kg and it needs to be delivered to a certain floor. The maximum total weight of shipment is 700 kg. The highest floor for parcel to be delivered in case if there is no elevator is the fifth floor.

LV: Not available in Latvia.

### **Courier brings label**

This additional service can be used when the sender customer does not have a possibility to print a label. The label will be brought by a courier upon parcel pick-up. Only next day courier pick-up available when using this service!

In addition to main service blocks, these additional blocks are needed:

- Additional service block

Example: chapter 12.3

In addition to main service blocks, these additional blocks are needed:

- Additional service block

In addition to main service blocks, these additional blocks are needed:

- Additional service block
- Pickup block

### **DPD Pickup return**

Identical to DPD Pickup return main service. In case if this additional service will be required instead of the main service, there will be return labels generated together with the main service labels.

In addition to main service blocks, these additional blocks are needed:

- Return block
- Additional service block

## 4. Authorization management

Every API request will need to have a bearer token in the Header for user authorization. An unauthorized request won't be processed.

Please keep your tokens secure. Any action done by a specific token will be treated as an action by that specific user. In case of created and sent shipments, these shipments will be invoiced to that specific customer.

Here you will find information about the main functions that will allow you to manage your tokens.

Each user can have up to 100 tokens! Inactive tokens must be deleted!



### 4.1. Token processing by GUI

There is a full list of active tokens for specific user, that can be retrieved by clicking on "Token list" hyperlink under user profile Customer data block. It's possible to create a new token there, delete the token or to get information about the token: When was the token created? By whom was it created? When was it last used? how many times it was used? What is the deadline for it?

In case if new token is created here, token name and validity period (in seconds) will be requested. Validity field is optional – token will be unlimited in case if this will be left unfilled. Any parameters for the token won't be editable after it is created. In case if any changes will be needed, a new token must be created.

Please note that for security purposes the token will be displayed only once, therefore it must be copied and saved securely. In case if token is lost, a new token must be created.

CUSTOMER DATA

CUSTOMER COMPANY

CUSTOMER COMPANY CONTACT PHONE

API KEY (FOR PARCEL STATUS WEB SERVICE)  
19071\_vlQgDTzvNwNDglwzLyrYVb [Regenerate api key](#)

[Token list](#)

USERS  
 View all shipments

APPLICATIONS  
[Automatic import application](#)  
[Direct print application](#)

TOKEN NAME  
Test token

TOKEN TIME-TO-LIVE IN SECONDS (OPTIONAL)  
3000

Valid until: 2022-07-14 10:01:25

[CREATE](#)

Token created successfully

Please save this token as it will not be display anymore

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1Ni99.eyJzdXNOb2Z1ct9pZCI6MTkwnzEslmFibWluX2lkjpdWks

[COPY](#)

## 4.2. Token processing by API request

### 4.2.1. Authorization token creation

This method generates an authorization token that can be used for any of the web service methods.

Method: **/auth/tokens**

Type: POST

Headers Authorization: Basic auth (by providing DPD system username and password)

#### Request:

---

Name	Type	Length	Req.	Description
name	string	256	M	Your assigned token name that can be used for token identification purposes.
ttl	integer	11	O	Token validity period in seconds. Must be max 99999999999. In case of <i>null</i> , there will be lifetime validity granted (more than 3000 years).

#### Response:

---

Name	Type	Length	Description
secretId	string	36	Unique ID number for the token.
validUntil	datetime	19	Returns the date and time of the token expiration (YYYY-MM-DD HH:mm)
token	string	1000	Authorization token.

## 4.2.2. Authorization token list retrieval

This method provides a list of all active authorization tokens. To retrieve the list, there must be a valid token provided in the header bearer authentication.

If a token has been forgotten, the token value cannot be retrieved. The new token must be generated instead and the previous existing token must be deleted.

Method: **/auth/token-secrets**

Type: GET

---

**Request:**

No parameter is needed

---

**Response:**

Each token block:

Name	Type	Length	Description
secretId	string	36	Unique ID number for the token.
name	string	256	Token name that has been assigned during a token creation.
createdAt	datetime	19	Date and time of the token creation (YYYY-MM-DD HH:mm)
uses	integer	10	Count of requests performed using a specific token.
lastUse	datetime	19	Date and time of the last usage of a token (YYYY-MM-DD HH:mm)
validUntil	datetime	19	Date and time of the token validity (YYYY-MM-DD HH:mm)

## 4.2.3. Authorization token deletion

This method deletes specific authorization tokens.

To retrieve the list, there must be a valid token provided in the header bearer authentication. Please be aware that there is an option to delete a specific token by using the same token. In such case, it will be deleted and there won't be an option to use it again.

Method: **/auth/token-secrets**

Type: DELETE

### Request:

---

Name	Type	Length	Req.	Description
secretId	string	36	M	secretId of the token that must be deleted. This parameter must be provided within link: {endpoint}/auth/token-secrets/01188c2a-88a6-4063-be2a-fd61becc09bc

### Response:

---

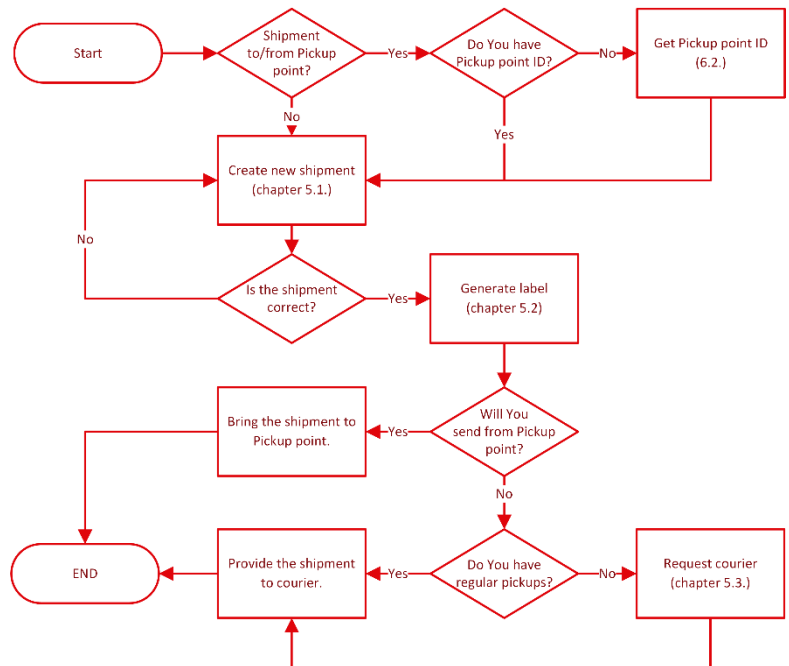
HTTP 204 status code

# 5. Shipment sending process

This is a standard process that describes how a shipment can be sent. This chapter contains information of how to create a shipment, generate shipment labels and request a courier if needed.

As soon as parcel IDs are assigned to a shipment, the receiver information is assigned to the specific parcel for 6 months and no other shipments will have the same IDs. Therefore, there is no shipment editing functionality.

You can find a few tips in chapter 9 that will allow you to use these methods more efficiently.



## 5.1. Shipment creation

This method creates a shipment that can contain one or multiple parcels. The data that is needed for creating shipments will depend on the DPD service that is requested.

Method: **/shipments**

Type: POST

Examples: chapter 12.1, chapter 12.2, chapter 12.3

### Request

---

Every request can contain up to 50 shipment blocks, every shipment block must contain data blocks according to the mandatory blocks of the required service (chapter 3). Additional data block can be used according to specifics of the exact block.



## Request - Payer code

---

This parameter is optional.

It allows a user to create a shipment on behalf of another user (only if the correct permissions have been granted)

Name	Type	Length	Req.	Description
payerCode	integer	7	O	DPD client ID that will be invoiced. Specific permissions must be granted before using this functionality (chapter 9)

## Request - Sender block

---

This block is mandatory.

Parameter name: *senderAddress*

Based on the sender's address, there are 2 options how this block should be filled in (in case if both options will be used together – *puold* parameter will be submitted, request will be treated as shipment from Pickup point):

### Request - Sender block - From address

---

Use if sender's address is available or if a parcel cannot be returned to a Pickup point in case of failed delivery (for example – because of the parcel size):

Name	Type	Length	Req.	Description
name	string	35	M	Sender's name and surname or company's name.
email	string	100	O	Sender's email address Only one email address on this parameter.
phone	string	30	M	Sender's phone number that will be displayed on a label. Only one phone number on this parameter. No other information should be provided here! There must be an international country code provided, e.g. "+372555555", "+37065123456" If there is no country code, it will be added automatically based on the <i>country</i> parameter.
street	string	35	M	Sender's address. In case it is not possible to separate, this can contain street name + property number or street name + property number + flat number.
streetNo	string	8	O	Sender's property number. In case it is not possible to separate, this can contain property number + flat number.
flatNo	string		O	Sender's flat number. If both parameters ( <i>streetNo</i> and <i>flatNo</i> ) are provided, max length (both parameter character sum) is reduced to 7
city	string	35	M	Sender's city.

postalCode	string	7	M	Sender's postal code. Without the country code and spaces.
country	string	3	M	Sender's country. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.

### Request - Sender block - From Pickup point

Use in case there is no sender address available and shipment must be returned to a Pickup point in case of a failed delivery:

Name	Type	Length	Req.	Description
name	string	35	M	Sender's name and surname or company's name.
email	string	100	O	Sender's email address Only one email address on this parameter.
phone	string	30	M	Sender's phone number that will be displayed on a label. Only one phone number on this parameter. No other information should be provided here! There must be an international country code provided, e.g. "+372555555", "+37065123456". If there is no country code, it will be added automatically based on the <i>country</i> parameter.
pudold	string	20	M	Pickup point's ID from which a shipment will be sent. This can be obtained by Pickup point search method – chapter 7.2.

### Request – Receiver block

This block is mandatory.

Parameter name: *receiverAddress*

There are 2 options how this block should be filled in based on the service selected - how the parcel will be delivered (in case if both options will be used together – *pudold* parameter will be submitted, request will be treated as shipment to Pickup point):

### Request – Receiver block – To-door delivery

Name	Type	Length	Req.	Description
name	string	35	M	Recipient's name and surname or company's name.
contactInfo	string	35	O	Additional information that will be displayed on the label.
email	string	100	O / M	Recipient's email address Mandatory for DPD Latvia. Even though this parameter is optional at the moment for DPD Lithuania and DPD Estonia, we strongly

				recommend providing it as it can increase delivery's quality. Only one email address on this parameter.
phone	string	30	M	Recipient's phone number that will be displayed on a label. Only one phone number on this parameter. No other information should be provided here! There must be an international country code provided, e.g. "+372555555", "+37065123456". If there is no country code, it will be added automatically based on the <i>country</i> parameter.
street	string	35	M	Recipient's address. In case it is not possible to separate, this can contain street name + property number or street name + property number + flat number.
streetNo	string	8	O	Recipient's property number. In case it is not possible to separate, this can contain property number + flat number.
flatNo	string		O	Recipient's flat number. If both parameters ( <i>streetNo</i> and <i>flatNo</i> ) are provided, max length (both parameter character sum) is reduced to 7
city	string	35	M	Recipient's city.
postalCode	string	7	M	Recipient's postcode. Without the country code and spaces.
country	string	3	M	Recipient's country. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.

#### Request - Receiver block – To Pickup point

Name	Type	Length	Req.	Description
name	string	35	M	Recipient's name and surname or company's name.
contactInfo	string	35	O	Additional information that will be displayed on the label.
email	string	100	O / M	Recipient's email address Mandatory for DPD Latvia. Even though this parameter is optional at the moment for DPD Lithuania and DPD Estonia, we strongly recommend providing it as it can increase delivery's quality. Only one email address on this parameter.
phone	string	30	M	Recipient's phone number that will be displayed on label. Only one phone number on this parameter. No other information should be provided here! There must be an international country code provided, e.g. "+372555555", "+37065123456" If there is no country code, it will be added automatically based on the <i>country</i> parameter.
pudold	string	20	M	Pickup point's ID where the shipment should be sent.

				This can be obtained by Pickup point search method - Chapter 7.2.
--	--	--	--	---

### Request – Return block

---

This block is mandatory only for additional services SWAP and DPD Pickup return.

Parameter name: *returnAddress*

In case of these additional services 2 labels will be generated:

1. Shipment from sender address to receiver address
2. Shipment from receiver address to return address

If return address is identical to sender address, please use sender block data to fill return block.

For technical specification Request - Receiver block specification must be used.

### Request – Service block

---

This block is mandatory!

Parameter name: *service*

Parameters within this block must be filled in according to the services available for a specific user (chapter 7.1)

Name	Type	Length	Req.	Description
serviceAlias	string	120	M	DPD assigned name alias of the required service. Parameter <i>serviceAlias</i> value from service list response (chapter 7.1) must be used here.

### Request – Additional service block

---

This block is optional!

Parameter name: *additionalServices*

Parameters within this block must be filled in according to the services available for a specific user (chapter 7.1)

Name	Type	Length	Req.	Description
serviceAlias	string	120	M	DPD assigned name alias of the required additional service. Parameter <i>serviceAlias</i> value from service list response (chapter 7.1) must be used here.

fields	array			Accordingly, to service list response (chapter 7.1), specific additional service parameters must be used here. Parameter name that should be used in this array is defined in <i>specialFields name</i> value, information what should be listed within specific parameter – <i>specialFields description</i> . Note that other <i>specialFields</i> parameter ( <i>mandatory, type, possibleValues, validationType, validationRules</i> ) can contain valuable information how specific parameter should be provided as well as how it will be validated by DPD.
--------	-------	--	--	---

**Request – Parcels / pallets blocks**

This block is mandatory (one of these blocks must be provided).

Parameter name: *parcels*

There can be up to 50 blocks, every block must contain parameters for each parcel:

Name	Type	Length	Req.	Description
weight	float	(8,3)	M	Parcel weight in kilograms. Note! Currently, in case of multiple parcels, the response will change this value to an average weight.
mpsReferences	array	4	O	Parcel references. Up to 4 (four) references (string, 35).

Parameter name: *pallets*

There can be up to 50 blocks, every block must contain parameters for each pallet:

Name	Type	Length	Req.	Description
weight	float	(7,2)	M	Pallet weight in kilograms
type	string	3	M	Pallet type. Values: <ul style="list-style-type: none"> <li>• EUR</li> <li>• FIN</li> </ul>
mpsReferences	array	4	O	Parcel (pallet) references. Up to 4 (four) references (string, 35).

**Request – Shipment reference**

This parameter is optional.

It allows to set shipment-based references that can be used for specific reports. If parcel/pallet-based reference is needed, *parcels/pallets* block *mpsReference* should be used instead.

Name	Type	Length	Req.	Description
shipmentReferences	array	4	O	Up to 4 (four) references (string, 35).

### Request – Pickup block

This block is mandatory only for main service Collection request or additional service Courier brings label.

Parameter name: *pickup*

It will provide information to DPD when courier should arrive to pick up parcels at sender's address.

### Request – Pickup block – Collection request

Name	Type	Length	Req.	Description
pickupDate	date	10	M	Desired pickup date (YYYY-MM-DD) Starting from next working day. There can be local limitations what is the final time to request pickup for next day. If this limit is breached DPD has permissions to change pickup date to the next working day (+1 day).
messageToCourier	string	250	O	Additional information for pickup. For example: door code, floor, etc.

### Request – Pickup block – Courier brings label

Name	Type	Length	Req.	Description
pickupDate	date	10	M	Desired pickup date (YYYY-MM-DD) There can be limitations until what time the same day pickup can be requested. If this period is missed, please choose another business day.
pickupTimeFrom	time	5	O	Desired pickup time interval – starting time (HH:mm) Minutes should be either 00 or 30. There can be interval restrictions that can be affected by country and ZIP code. Request should be submitted at least 15 minutes before pickup time.
pickupTimeTo	time	5	O	Desired pickup time interval – final time (HH:mm) Minutes should be either 00 or 30. There can be interval restrictions that can be affected by country and ZIP code.
messageToCourier	string	250	O	Additional information for pickup. For example: door code, floor, etc.

## Request – Additional functionality block

---

This block is optional and should be used in one of these cases:

- if you want to save addresses into DPD system address book
- if you want to generate digital label (PIN and QR code) that can be used for printing the label at the parcel locker.

Parameter name: *shipmentFlags*

This block must contain the following parameters:

Name	Type	Length	Req.	Description
savesSenderAddress	boolean	true/false	O	In case of “true”, sender address will be saved in address book in DPD system.
savesReceiverAddress	boolean	true/false	O	In case of “true”, receiver address will be saved in address book in DPD system.
generatesDplPin	boolean	true/false	O	In case of “true”, response will contain additional parameters – digital label PIN code and QR code (as PNG binary encoded file).

## Response

---

Name	Type	Length	Description
id	string	36	Shipment identifier this identifier will be needed for any other action with the shipment.
parcelNumbers	array	100	A list of parcel identifiers (string, 14) Empty by default. Parcel identifiers will be provided if the parcel label is requested in combination with the shipment creation (Chapter 9).
payer	object	1	Payer information

*payer* object:

Name	Type	Length	Description
code	integer	7	DPD customer ID that will be invoiced.

If the *generatesDplPin* is set to “true”, there will be *dplPin* parameter in the response that will contain a set of arrays for each parcel:

Name	Type	Length	Description
parcelNumber	integer	14	Parcel identifier that will have a PIN code and QR code (digital label) within the array.
dpl	blob		Binary encoded PNG file that contains QR code (digital label) that can be scanned at the parcel locker to print the parcel label at the locker.
pin	integer	6	PIN code that can be used instead of QR code to print the parcel label at the locker.

DPL service is only available in Baltics. DPL and related PIN code won't work in locker of any other country!



There will also be a full list of request parameters added to the response. This can be used for response data validation regarding the requested data.

Parcel/pallet block can contain an additional parameter *parcelNumber* (integer, 14), that will indicate which references and weight data is assigned to the specific parcel identifier. This parameter will be provided in case the label or invoice is generated.

## 5.2. Label creation

This method generates labels for shipments/parcels that were created either by a user or a user that is granted permissions to access specific shipments/parcels.

Method: **/shipments/labels**

Type: POST

Example: chapter 12.4

### Request

Name	Type	Length	Req.	Description
shipmentIds	array	50	M	Shipment identifier (string, 36) for which label should be generated. The labels will be printed for all of the parcels that are within the shipment will be printed. Only one of these parameters ( <i>shipmentIds</i> , <i>parcelNumbers</i> ) can be used at a time.
parcelNumbers	array	50		Parcel identifier (integer, 14) for which label should be generated. In case only one of the shipment parcels is requested, just the specific parcel will be processed. Only one of these parameters ( <i>shipmentIds</i> , <i>parcelNumbers</i> ) can be used at a time.



offsetPosition	integer	1	O	Starting positions of the first DPD label on an A4 page. Values: <ul style="list-style-type: none"> <li>• 0 – starts on top left corner</li> <li>• 1 – starts on top right corner</li> <li>• 2 – starts on bottom left corner</li> <li>• 3 – starts on bottom right corner</li> </ul> For A6 page size, this value can only be 0.
downloadLabel	boolean	true/false	M	True/false parameter that indicates if the response should contain a label file In case of false, 204 header response will be returned.
emailLabel	boolean	true/false	O	True/false parameter that indicates if DPD should send a label to a sender's email address. If shipment creation request parameter <i>generatesDplPin</i> was set as "true", email will contain digital label PIN code and additional attachment of digital label QR code as PNG file. Default value: false
labelFormat	string	15	O	Requested file format. Values: <ul style="list-style-type: none"> <li>• application/pdf (default, PDF file)</li> <li>• image/png (PNG file)</li> </ul>
paperSize	string	2	O	Paper size. Values: <ul style="list-style-type: none"> <li>• A4 (default)</li> <li>• A6</li> </ul>

## Response

Name	Type	Length	Description
shipmentIds	array	50	List of shipment identifiers (string, 36) that was processed.
parcelNumbers	array	-	List of parcel numbers (string, 14) that were processed.
labelFormat	string	15	Requested label format. As set in request <i>labelFormat</i> parameter.
pages	block	100	Binary encoded parcel label files. Each block consists of a parameter "binaryData" (blob), that contains binary encoded file. In case of a PDF file, there will only be one file, in case of a PNG file – each page will be provide as a separate block.

In case of the A4 page document, labels' sequence on the page is as follows: top left, top right, bottom left, bottom right.

## 5.3. Courier request

This method submits a courier request to inform DPD that a courier pickup (arrival) is needed from a specific address during a specific time frame. Please keep in mind that there can be specific courier request conditions for each country, as well as different conditions within one country (based on the geographical location – postal code). Parameters like:

- the pickup date (until which time it's possible to request a courier for the same day arrival),
- pickup starting time (from what time is the shipment ready for pickup),
- pickup final time (the last time until which the courier can arrive),
- minimal interval between starting time and final time,
- interval from request submission time until the starting time

These parameters are described in DPD service use terms and conditions and are subject to change (changes can be affected by regular processes like peak periods, as well as by unpredictable processes like pandemics).

Method: **/pickups**

Type: POST

Example: chapter 12.5

### Request

Name	Type	Length	Req.	Description
pickupDate	date	10	M	Desired pickup date (YYYY-MM-DD) There can be limitations until what time the same day pickup can be requested. If this period is missed, please choose another business day.
pickupTimeFrom	time	5	M	Desired pickup time interval – starting time (HH:mm) Minutes should be either 00 or 30. There can be interval restrictions that can be affected by country and ZIP code. Request should be submitted at least 15 minutes before pickup time (precise information about <i>cutoff</i> time can be found in pickup timeframe list – chapter 7.6)
pickupTimeTo	time	5	M	Desired pickup time interval – final time (HH:mm) Minutes should be either 00 or 30. There can be interval restrictions that can be affected by country and ZIP code.
address	block	1	M	Pickup address - where courier must arrive (see below).
messageToCourier	string	250	O	Additional information for pickup. For example: door code, floor, etc.

shipmentUids	array	-	M	Shipment identifier (string, 36) about shipments that must be picked up. Either <i>shipmentUids</i> or <i>parcel</i> and/ or <i>pallets</i> parameters can be used at a time.
parcel	block	1		Information about parcels that must be picked up. Either <i>shipmentUids</i> or <i>parcel</i> and/ or <i>pallets</i> parameters can be used at a time.
pallets	block	50		Information about pallets that must be picked up. Either <i>shipmentUids</i> or <i>parcel</i> and/ or <i>pallets</i> parameters can be used at a time.

address block:

Name	Type	Length	Req.	Description
name	string	35	M	Name and surname or company's name, where the pickup must be made
contactName	string	35	M	Contact person's name, who could be contacted regarding the pickup. If <i>name</i> already contains this information, this parameter must contain the same information.
email	string	100	O	Pickup contact person's email address Only one email address on this parameter.
phone	string	30	M	Pickup contact person's phone number. Only one phone number on this parameter. No other information should be provided here! There must be an international country code provided. e.g. "+372555555", "+37065123456"
street	string	35	M	Pickup address. In case it is not possible to separate, this can contain street name + property number or street name + property number + flat number.
streetNo	string	8	O	Pickup property number. In case it is not possible to separate, this can contain property number + flat number.
flatNo	string	8	O	Pickup flat number.
city	string	35	M	Pickup city.
postalCode	string	9	M	Pickup postcode. Without the country code and spaces.
country	string	2	M	Pickup country. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.

parcel block:

Name	Type	Length	Req.	Description
count	integer	5	M	Parcel count that must be picked up Each parcel must not exceed 31.5 kg. If a parcel is heavier than 31.5 kg, it must be submitted as a pallet.
weight	float	(7,2)	M	Parcel weight in kilograms. In case multiple parcels have to be picked up, an average parcel weight can be provided. Max 31.5 kg.

*pallets* – block must contain array of parameters for each pallet:

Name	Type	Length	Req.	Description
weight	float	(7,2)	M	Pallet weight in kilograms In case of multiple pallets, the sum of all pallet weight must be provided. Max 32767 kg
type	string	3	M	Pallet type. Values: <ul style="list-style-type: none"> <li>• EUR</li> <li>• FIN</li> </ul>

## Response

Name	Type	Length	Description
address	block	1	Address data as in request.
shipmentUids	array	-	Information about shipments according to the request.
parcel	block	1	Information about parcels according to the request.
pallets	blocks	50	Information about pallets according to the request.
messageToCourier	string	250	Additional information for pickup according to the request
pickupDateFrom	datetime	19	Desired pickup time interval – start time (YYYY-MM-DD HH:mm).
pickupDateTo	datetime	19	Requested pickup time interval – end time (YYYY-MM-DD HH:mm).
payerCode	integer	7	DPD client ID, requesting the pickup.

## 6.Parcel tracking

There are 2 options to retrieve parcel delivery statuses. It is possible to retrieve statuses for specific parcels on demand or you can use call-back (DPD system would send data to specific endpoint in case of any new parcel delivery status).

Please note that:

- Only statuses for shipments with Baltic parcel numbers created by specific user are available.
- Data is available for the period of last two months.
- When multiple parcel numbers are returned, results are returned in order passed.
- Statuses for each parcel are returned in descending order (newest first).

### 6.1. On demand

This method provides information regarding parcel statuses. You can have up to 30 parcels checked per request. In case if more parcels need to be checked, there should be multiple requests made where none of those contains more than 30 parcels.

Additionally – there is no need to request all events (*show\_all=1*) every time. Scheduled task (*CROM*) can request only the latest event (*show\_all=0*), but in case if You want to use the data for internal reports, you can request all events only once at the end of the life cycle (when parcel is delivered).

Method: **/status/tracking**

Type: GET

#### Request

Name	Type	Length	Req.	Description
pknr	String	2	M	Parcel numbers. In case if multiple parcels are provided those have to be separated with " ". Maximum 30 parcels in one request are allowed. Note! Parcel number consists of 14 numeric characters.
detail	Char	1	O	Response detail level. Values: <ul style="list-style-type: none"><li>• 0 – basic (default, chapter 6.1.1)</li><li>• 3 – advanced (chapter 6.1.3)</li></ul> Note! detail value "1" and "2" is legacy value that is not supported anymore.

show_all	Char	1	0	Number of statuses for each parcel. Values: <ul style="list-style-type: none"> <li>• 0 – return only latest parcel status (default)</li> <li>• 1 – return all parcel statuses</li> </ul>
lang	Varchar	2	0	Language for status in case if detail value "0". Values: <ul style="list-style-type: none"> <li>• en – English (default)</li> <li>• lt – Lithuanian</li> <li>• lv – Latvian</li> <li>• ee – Estonian</li> </ul>

## Response

Information about every parcel is returned as separate array, that contains:

Name	Type	Length	Description
parcelNumber	varchar	14	Parcel number
details	array		Array of statuses (numerical values) in case of successful request.
error	array		Array of error information

*error array:*

Name	Type	Length	Description
code	integer	3	Error identification code
message	string	50	Error message

### 6.1.1. Details array for basic data (detail: 0)

Name	Type	Length	Description
status	string	62	Parcel status message (chapter 6.1.2)
dateTime	dateTime	19	Event date and time (YYYY-MM-DD HH:mm:ss)

## 6.1.2. Parcel status messages

English	Latvian	Lithuanian	Estonian
Dropped in Pickup Point	Paka nodota Pickup punktā	Atnešta į siuntų tašką/terminalą	Viidud Pickup punkti
Picked up by Courier	Kurjers paņēmis paku	Kurjeris paėmė siuntą	Kulleri poolt peale korjatud
En route	Paka ir ceļā	Pakeliui	Teel
Delivered to Consignee	Paka piegādāta klientam	Pristatyta gavėjui	Saajale kohale toimetatud
Delivered to Pickup Point	Paka piegādāta Pickup punktā	Pristatyta į siuntų tašką/terminalą	Toimetatud Pickup punkti
Picked up by Consignee from Pickup point	Klients saņēmis paku Pickup punktā	Gavėjas atsiėmė siuntą iš siuntų taško/terminalo	Saaja poolt Pickup punktist välja võetud
Returning to Sender	Paka tiks atgriezta nosūtītājam	Grąžinama siuntėjui	Tagastamisel saatjale
Returned to Sender	Paka ir atgriezta nosūtītājam	Grąžinta siuntėjui	Tagastatud saatjale

## 6.1.3. Details array for advanced data (detail: 3)

Name	Type	Length	Description
serviceCode	var	3	Parcel service identifier, that can be used for tracking parcel (chapter 6.1.4). Service code can change in case of wrong service code or if parcel is returned to sender.
statusCode	var	2	Parcel status identifier, that can be used for tracking parcel (chapter 6.1.4).
dateTime	datetime	19	Event date and time (YYYY-MM-DD HH:mm:ss)
Tour	varchar	3	DPD tour identifier
GpsLat	float	(8,5)	GPS Latitude of the place where event was made. Value might be provided in case it event was made by courier.
GpsLon	float	(8,5)	GPS Longitude of the place where event was made. Value might be provided in case it event was made by courier.

TimeFrame	string	9	Aproximate delivery time (HHmm-HHmm) Value might be provided for events where <i>statusCode</i> is 03 and it indicates approximate delivery time. Example: 1127-1257
AddCode	string	Up to 11	Additional information about the event (chapter 0) Multiple additional codes are separated by comma.
Weight	float	(6,2)	Parcel weight fixed by DPD (kilograms). Value might be provided for events where <i>statusCode</i> is 05 or 10.
Depot	varchar	4	DPD identifier of depot where scan was made.
City	string	30	City where DPD depot is located.
CountryCode	varchar	3	ISO-3166 code of country where scan was made. Examples: 440, 428, 233
CountryIsoName	varchar	2	ISO-3166-2 name of country where scan was made. Examples: LT, LV, EE
prevStatusCode	varchar	2	Previous event's status code or "Multiple" if there are more than one event with same time.

## 6.1.4. Status codes / service codes

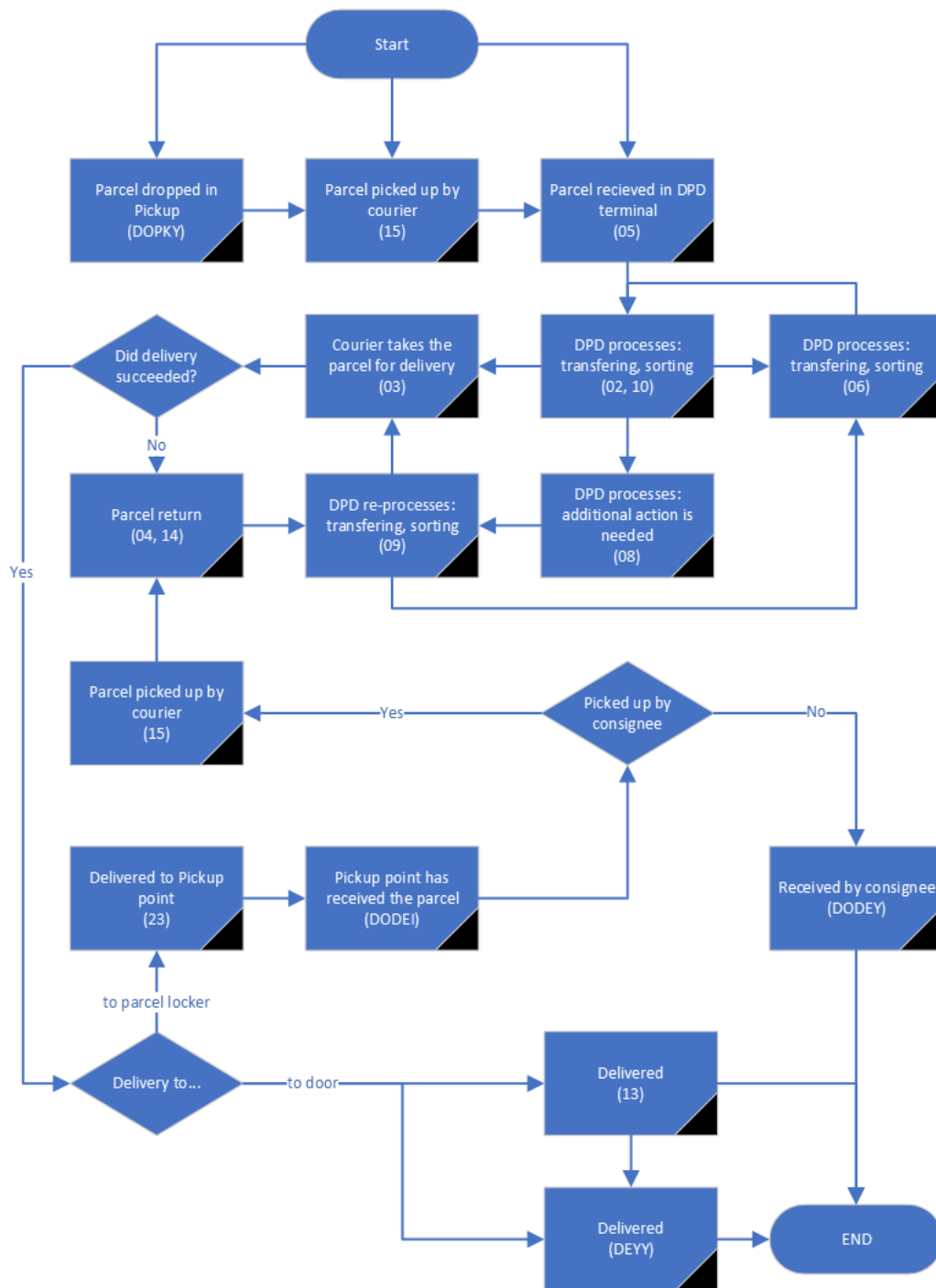
StatusCode is used for internal DPD processes to identify parcel life cycle status. As from parcel life cycle perspective delivery event finalizes parcel life cycle, there is identical status code for cases when parcel was delivered to consignee or when it was delivered back to sender. Therefore, to understand the correct status of the parcel, there is a need to use multiple parameters - combinations of *statusCode* and *serviceCode* allows You to get correct information:

<i>statusCode</i>	<i>serviceCode</i> (at least one)	<i>prevStat usCode</i>	Parcel location	Description
01			In terminal	Parcel is processed (consolidated) in DPD terminal.
02			In terminal	Parcel was accepted in terminal.
03			At the courier	Parcel was scanned by courier before going out of terminal for delivery to consignee.
03	298, 299, 300, 301, 332		At the courier	Parcel was scanned by courier before going out of terminal for delivery to sender
04			In terminal	Delivery failed. Parcel was returned to terminal.
05			In terminal	Parcel was picked up in DPD terminal This will be primary event in case if parcel won't be scanned by courier on pickup.
06			In terminal	Parcel is processed in DPD terminal. Return to sender or redirection to other address.



08			In terminal	Parcel was stopped in terminal. Additional action/information is needed.
09			In terminal	Parcel was processed for re-delivery, returning to sender or transferring to another terminal.
10			In terminal	Parcel is in sorting process for delivery to next DPD terminal.
13			<b>Delivered</b> At the consignee	Parcel is delivered to consignee.
13	298, 299, 300, 301, 332		<b>Delivered</b> At the sender	Parcel is returned to sender.
14			At the courier	Parcel was not delivered, and it was scanned by courier before returning to terminal.
15			At the courier	Parcel was picked up from consignee and it was scanned by courier on pickup. This can be primary event. Based on delivery specifics this event can be missed out.
20			In line-haul	Parcel has been loaded in truck on the way to the next DPD terminal.
23			In pickup point	Parcel was delivered by courier to pickup point.
DODEI			In pickup point	Parcel was inserted in Parcel locker or parcel was collected from courier by Parcel shop.
DODEY			<b>Delivered</b> At the consignee	Parcel was picked up by consignee from Pickup point.
DOPKY			In pickup point	Parcel was inserted in parcel locker by sender This can be primary event. It can be followed by 05 or 15 status codes.
DEYY		13, 03	<b>Delivered</b> At the consignee	Parcel is delivered to consignee.
DEYY		04	In terminal	Delivery failed. Parcel was returned to terminal.
DEYY			-	Info event, internal status

This is how approximately delivery process looks like:



## 6.1.5. Additional codes

Additional codes can be used to get more information about the specific event – for example – reason why parcel was not delivered to consignee. We have tried to describe some of additional codes and:

<i>AddCode</i> (at least one)	Description
12, 16, 22	Parcel damages were discovered
14, 15, 16	Parcel was refused by consignee
80	Delivery date/time was changed by consignee.
11, 12, 14, 15, 16, 22, 24, 25, 29, 30, 32, 33, 37, 41, 42, 46, 47, 49, 50, 61, 62, 66, 72, 73, 84, 85, 94, 95, 96	Additional information is needed from sender to proceed with delivery.

There can be combinations of these events – for example: in case of value 12, DPD could contact sender to recheck if the damaged parcel should be delivered to consignee.

The rest of codes are used for DPD internal processes and there is no need to process those.

## 6.2. Call-back request

This method allows to retrieve information to specific endpoint as soon as any changes happens to specific parcel with 2 month period. After 2 months specific subscription is automatically suspended. Unsubscription is not required and can be used only in specific cases.

Method (subscribe): **/status/events/subscribetoparcel**

Method (unsubscribe): **/status/events/unsubscribetoparcel**

Type: GET

### Request

---

Name	Type	Length	Req.	Description
parcelnumber	string	14	M	Parcel number. Note! Parcel number consists of 14 numeric characters.
callbackurl	string	255	M	URL where data must be submitted whenever there is any new status for the parcel. Note.URL needs to be URL-encoded before passing to this request. For example, "http://somesite.com" must be provided as "http%3A%2F%2Fsomesite.com"  In case of unsubscribing this parameter is optional, but both parameters need to match to the ones that were submitted on subscription.

### Response

---

HTTP 200 status code in case of:

- correct subscription.
- correct unsubscription
- no subscription to unsubscribe from

HTTP 400 status code in case if there is an existing subscription on specific combination.

### Call-back request

---

Type: POST

Once the call-back URL is invoked a POST request is sent to specific URL every time parcel status changes. Information about the parcel and the new status will be provided within BODY of the request as JSON accordingly to tracking (chapter 6.1 and chapter 6.1.1) request where *pknr* is the parcel ID that was registered, *detail=0*, *show\_all=0* and *lang=en* (these values are not configurable).

## 7. Additional functionality

Additional functions list consists of methods that either are needed in certain situations, or will help you to manage information within the DPD system.

### 7.1. Service list retrieval

This method will return a list of DPD services that can be used by a specific user.

Method: **/services**

Type: GET

#### Request

Name	Type	Length	Req.	Description
countryFrom	String	2	M	Sender's country. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.
postalCodeFrom	String	7	O	Sender's postcode. Without the country code and spaces.
countryTo	String	2	M	Recipient's country. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.
postalCodeTo	String	7	O	Recipient's postcode. Without the country code and spaces.
serviceType	Enum	1	O	An additional DPD service classifier that identifies the receiver type. Can be one of these options: <ul style="list-style-type: none"><li>• Business</li><li>• Private</li><li>• Pudo</li><li>• Collection Request</li><li>• Return through Pudo</li></ul>
mainServiceName	String	120	O	Name of the DPD main service that is searched
mainServiceAlias	String	50	O	The main service alias used within the shipment creation requests that is searched.
payerCode	integer	7	O	DPD client ID that will be invoiced. Specific permissions must be granted before using this functionality (chapter 9)

## Response

Name	Type	Length	Description
serviceName	string	255	Name of the service used.
serviceAlias	string	255	The value that must be used on the shipment creation method.
serviceType	array	5	An additional DPD service classifier that identifies the receiver type. Can contain one or multiple of these options (string, 50): <ul style="list-style-type: none"> <li>• Business</li> <li>• Private</li> <li>• Pudo</li> <li>• Collection Request</li> <li>• Return through Pudo</li> </ul>
specialFields	-	-	This parameter can be ignored for main services (it will always be empty). <i>Only the additional services will contain special fields.</i>
price	float	(5,2)	This parameter by default will be 0. It is used only for specific cases when suggested by DPD. For pricing calculation purposes please read chapter 9.
message	blob		Additional information about the service <i>Line breaks and backslashes are escaped.</i>
additionalServices	block	-	Information on additional services available for the specific main service (see below)
additionalRestrictions Apply	boolean	true/false	True/false parameter that indicates if there are any geographical restrictions for the specific service. For example – it will be true if the request is done for a domestic shipment, while the service is only available for a delivery to a specific zip code.

*additionalServices* array:

Name	Type	Length	Description
serviceName	string	120	Name of the service used.
serviceAlias	string	50	Value that must be used on the shipment creation method.
serviceType	array	5	An additional DPD service classifier that identifies the receiver type. Can contain one or multiple of these options (string, 50): <ul style="list-style-type: none"> <li>• Business</li> <li>• Private</li> <li>• Pudo</li> <li>• Collection Request</li> <li>• Return through Pudo</li> </ul>
specialFields	block	-	Information about special parameters that must be provided to request the additional service (see below)

price	float	(5,2)	This parameter by default will be 0. It is used only for specific cases when suggested by DPD. For pricing calculation purposes please read chapter 9.
message	blob	-	Additional information about the service Line breaks and backslashes are escaped.
additionalServices	-	-	This parameter can be ignored for additional services (it will be always empty). Only main services will contain special fields.
additionalRestrictions Apply	boolean	true/false	True/false parameter that indicates if there are geographical restrictions for a specific service. For example – it will be true if the request is done for a domestic shipment, while the service is only available for a delivery to a specific zip code.

*specialFields* array:

Name	Type	Length	Description
name	string	100	Parameter name for the special field parameter that might be needed in case of an additional service request.
description	string	1000	Information about the purpose of the specific parameter.
mandatory	boolean	true/false	True/false parameter that indicates if a specific parameter is mandatory for shipment creation request.
type	string	20	Parameter type. Example: "integer", "float", "enum" etc.
possibleValues	array	-	If <i>type</i> has a value "enum", this parameter will contain all possible values for the specific parameter. Each parameter – string (string, 250)
validationType	string	50	Information on validation that must be passed to submit the parameter.
validationRules	array	-	List of all validations that will be done on the specific parameter. Each parameter – string (string, 250)

## 7.2. Pickup point list retrieval

This method returns a list of DPD Pickup points that are needed for creating shipments from/to specific pickup points.

In case of multiple parameters used in one request, only those Pickup points that consist of all the parameters will be provided. There will be blank response if no Pickup point matches all the parameters.

Method: **/lockers**


Type: GET

Headers accept application/json+fulldata

### Request

Name	Type	Length	Req.	Description
countryCode	string	2	M	Country code. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.
id	string	7	O	Pickup point id to search by.
name	string	35	O	Pickup point name. This parameter can contain partial match to specific value.
lockerType	enum	1	O	Pickup point type. Values: <ul style="list-style-type: none"><li>• PickupStation</li><li>• ParcelShop</li></ul>
street	string	35	O	Street name. This parameter can contain partial match to specific value.
postalCode	string	9	O	Postal code. Without the country code and spaces.
city	string	25	O	City name. This parameter can contain partial match to specific value.
startPointLatitude	float	(8,5)	O	Search starting point latitude in case of searching nearest Pickup point by GPS location. <i>startPointLongitude</i> and <i>radius</i> must be provided.
startPointLongitude	float	(8,5)	O	Search starting point longitude in case of searching nearest Pickup point by GPS location. <i>startPointLatitude</i> and <i>radius</i> must be provided.
radius	integer	6	O	Radius (in meters) for searching for the nearest Pickup points to position, provided in combination with: <ul style="list-style-type: none"><li>• Address values: <i>city</i> and/or <i>postalCode</i> (<i>street</i> can be provided to have more accurate results)</li><li>• GPS location: <i>startPointLatitude</i> and <i>startPointLongitude</i> parameters.</li></ul> In case if address and GPS location will be provided within one request, only those nearest to GPS location pickup points will be returned that has corresponding address values
distanceType	enum	1	O	This parameter needs to be used either in combination with <i>radius</i> parameter and one of these two - GPS coordinates ( <i>startPointLatitude</i> , <i>startPointLongitude</i> ) or address data ( <i>street</i> , <i>postalCode</i> , <i>city</i> ). In case if GPS coordinates and address will be provided coordin Values:



				<ul style="list-style-type: none"> <li>• air</li> <li>• walking</li> <li>• driving</li> </ul> <p>If this combination used, response will contain list of pickup points ordered ascending by distance from GPS coordinates or address to specific pickup point. In case if GPS coordinates and address will be provided, only address will be used. Without <i>radius</i> this option won't provide nearest pickup points.</p> <p>This is a specific service that needs to be enabled before using it. Please contact DPD support in case if this service is required!</p> 
order	enum	1	O	<p>Parameter by which pickup points will be ordered in ascending order.</p> <p>Values:</p> <ul style="list-style-type: none"> <li>• id</li> <li>• name</li> <li>• city</li> </ul> <p>In combination with <i>radius</i> this parameter will be ignored.</p>
lockerFeatures	array	5	O	<p>Information about the services available in the Pickup point.</p> <p>Values:</p> <ul style="list-style-type: none"> <li>• consigneePickupAllowed Parcel can be delivered to consignee.</li> <li>• returnAllowed Parcel can be dropped off in Pickup point</li> <li>• codAllowed COD service available.</li> <li>• codPaymentType_cash COD amount can be collected in cash.</li> <li>• codPaymentType_cheque COD amount can be collected by cheque.</li> <li>• codPaymentType_card COD amount can be collected by credit card.</li> </ul>

## Response

Name	Type	Length	Description
id	string	7	<p>Pickup point ID. Example: LT90008, LV10193, EE91017 etc. For Baltics this parameter contains:</p> <ul style="list-style-type: none"> <li>• country code – 1<sup>st</sup> and 2<sup>nd</sup> symbols (EE/LV/LT)</li> <li>• type – 3<sup>rd</sup> and 4<sup>th</sup> symbols: <ul style="list-style-type: none"> <li>○ 10 – parcelshop</li> <li>○ 90 – parcel locker</li> </ul> </li> <li>• id – 5<sup>th</sup>, 6<sup>th</sup> and 7<sup>th</sup> symbols</li> </ul>
name	string	35	<p>Pickup point name May include the name of the parcelshop service provider.</p>
lockerType	string	20	<p>Pickup point type Possible values:</p>

			<ul style="list-style-type: none"> <li>PickupStation</li> <li>ParcelShop</li> </ul>
address	block	1	Pickup point address (see below)
hours	block	7	Pickup point working hours (see below)
supportedServices	array	6	Pickup point supported services Values (each – string, 50): <ul style="list-style-type: none"> <li>consigneePickupAllowed Parcel can be delivered to consignee.</li> <li>returnAllowed Parcel can be dropped off in Pickup point</li> <li>codAllowed COD service available.</li> <li>codPaymentType_cash COD amount can be collected in cash.</li> <li>codPaymentType_cheque COD amount can be collected by cheque.</li> <li>codPaymentType_card COD amount can be collected by credit card.</li> </ul>
distance	integer	8	The distance in meters to the origin of the search

*address* block:

Name	Type	Length	Description
street	string	35	Pickup point street.
city	string	35	Pickup point city.
postalCode	string	7	Pickup point zip code.
country	string	2	Pickup point country name. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.
latLong	array	2	Array of 2 float (8,5) that indicates pickup point geographical location.

*hours* block:

Name	Type	Length	Description
Monday	block	2	Array of working hours on Monday.
Tuesday	block	2	Array of working hours on Tuesday.
Wednesday	block	2	Array of working hours on Wednesday.

Thursday	block	2	Array of working hours on Thursday.
Friday	block	2	Array of working hours on Friday.
Saturday	block	2	Array of working hours on Saturday.
Sunday	block	2	Array of working hours on Sunday.

*hours* (each day) block:

Name	Type	Length	Description
morning	array	2	Time values (from/to, HH:mm) that indicate the morning opening hours.
afternoon	array	2	Time values (from/to, HH:mm) that indicate the afternoon opening hours In cases where there is no break between morning and afternoon hours, there will be no time gap between both values. For example: 00:00-12:00 and 12:00-23:59

## 7.3. Payer list retrieval

This method provides a list of payers available for a specific user.

Method: **/customers/payers**

Type: GET

**Request:**

---

No parameter is needed

**Response:**

---

Each block:

Name	Type	Length	Description
id	integer	10	DPD internal identifier that won't be needed for any other purpose.
name	string	35	DPD customer name that will be invoiced.

code	integer	7	DPD customer ID that will be invoiced.
isDefault	boolean	true/false	True/false parameter that indicates if a specific payer is the default payer for the the specific user.

## 7.4. Shipment list retrieval

This method provides a list of all shipments created or accessible (if user has permissions to access other user shipments) by specific user.

Method: **/shipments**

Type: GET

### Request:

Name	Type	Length	Req.	Description
ids	array	-	O	List of shipment identifiers (string, 36).
status	array	-	O	Shipments in specific statuses. Values: <ul style="list-style-type: none"> <li>pending – shipment created</li> <li>not_printed – label not printed</li> <li>not_booked – carrier not ordered</li> <li>in_route – shipment in delivery</li> <li>delivered – shipment delivered</li> <li>returned – shipment returned to sender</li> <li>rdl_in_route – return digital label service shipments that are on the way to be returned to sender</li> <li>rdl_delivered – return digital label service shipments that are delivered back to sender</li> </ul> Default value: pending At the beginning of 2022 default value will be changed. If shipment consist of multiple parcels, search will be performed for latest parcel life cycle status, therefore these statuses are recommended to use in case of one parcel per shipment.
mainServiceAlias	array	-	O	List of main services (string, 50). <i>serviceAlias</i> must be used here (chapter 7.1)
additionalServiceAlias	array	-	O	List of additional services (string, 50). <i>serviceAlias</i> must be used here (chapter 7.1)

referenceNumber	string	35	O	One of shipment references.
parcelNumber	string	14	O	Parcel identifier (string, 14) to be searched for.
senderName	string	35	O	Sender's name and surname or company's name.
receiverName	string	35	O	Recipient's name and surname or company's name.
payerCode	integer	7	O	DPD client ID that will be invoiced.
user	string	255	O	Username for user that created shipments.
direction	string	30	O	Shipment type. Values: <ul style="list-style-type: none"> <li>• domestic</li> <li>• international</li> </ul>
creationDateFrom	date	10	O	Shipments created since date (YYYY-MM-DD). Default value: 30 days before request date
creationDateTo	date	10	O	Shipments created till date (YYYY-MM-DD).
limit	integer	10	O	Count of shipments that must be provided in response "page". Min value: 1 Max/default value: 100
page	integer	10	O	Current "page" of shipments that must be provided in response. Used only in cases if <i>limit</i> contains smaller value as total shipment count.

### Response:

Name	Type	Length	Description
items	block	-	Shipment data accordingly to structure of shipment creation (chapter 5.1). There can be additional parameters (listed below).
total	integer	10	Count of shipments that fits required criterias.
currentPage	integer	10	Current "page" of shipments. At the beginning of 2022 this parameter will be renamed to "page".
pageSize	integer	<u>10</u>	Count of shipments that must be provided in response "page". At the beginning of 2022 this parameter will be renamed to "limit"

*items* block (additional parameters to shipment creation response):

Name	Type	Length	Description
shipmentLabels	block	-	Label data according to response from chapter 5.2. Will be provided in case if label generation parameter <i>downloadLabel</i> is set as "true".

status	string	20	<p>Shipment's status. Values:</p> <ul style="list-style-type: none"> <li>• pending – shipment created</li> <li>• not_printed – label not printed</li> <li>• not_booked – carrier not ordered</li> <li>• in_route – shipment in delivery</li> <li>• delivered – shipment delivered</li> <li>• returned – shipment returned to sender</li> <li>• rdl_in_route – return digital label service shipments that are on the way to be returned to sender</li> <li>• rdl_delivered – return digital label service shipments that are delivered back to sender</li> </ul> <p>If shipment consist of multiple parcels, there will be latest parcel life cycle status, therefore these statuses are recommended to use in case of one parcel per shipment.</p>
--------	--------	----	---

## 7.5. Shipment deletion

This method removes shipment from user's shipments list.

Please note that as soon as DPD parcel ID is assigned to shipment, this creates a possibility for a customer to send the parcel even if it was previously removed from a shipments list, therefore shipment will not be deleted (it will be just removed from list to ease up customer processes), but data will be processed by DPD according to privacy policy as for any regular shipment.

If a shipment consists of multiple parcels, all parcels will be deleted together when shipment is deleted or - when one of shipment's parcels is deleted.

Method: **/shipments**

Type: DELETE

### Request:

Name	Type	Length	Req.	Description
ids	string	36	M	An identifier of a parcel or shipment which needs to be deleted.

### Response:

HTTP 204 status code

## 7.6. Courier request time frame retrieval

This method will provide possible time frames for courier request (chapter 0). There is no need to use this method every time before requesting courier. If there is a need for irregular courier requests this method can be requested once per day to save request time frames for following days.

Method: **/pickup-timeframes**

Type: GET

### Request:

Name	Type	Length	Req.	Description
dateTo	date	10	O	Limit for dates to return to return pickup timeframes (YYYY-MM-DD). Max/default value: 30 days from request date.
country	string	2	M	Pickup country. ISO 3166-1 alpha-2 country codes format, e.g. LT, LV, EE.
zip	string	7	M	Pickup postal code. Without the country code and spaces. At the beginning of 2022 this parameter will be renamed to "postalCode"
additionalServices	array	1	O	Additional service alias in case if additional service can affect courier arrival time frames. At the moment there is only Courier brings label additional service that can affect pickup timeframes.

### Response:

Name	Type	Length	Description
timeFrames	block	-	List of available pickup dates within the requested range.

*timeFrames* block:

Name	Type	Length	Description
date	date	10	Pickup date (YYYY-MM-DD)
timeFramesDTO	block	4	Information about pickup timeframes on specific date. At the beginning of 2022 this parameter will be renamed.

*timeFramesDTO* block:

Name	Type	Length	Description
timeFrameFrom	array	-	List of possible starting times (HH:mm)
timeFrameTo	array	-	List of possible final times (HH:mm)
minimalInterval	float	(3,1)	Interval (hours) between pickup starting time and final time that must be used for pickup request.
cutoff	integer	4	Interval (minutes) between request time and pickup starting time that must be used for pickup request.



## 8. Pricing / invoicing

Pricing for contracted customers is done according to an agreement conditions for the parcels that were shipped. If a shipment was created, label was printed, but parcels were not provided to DPD, shipment won't be invoiced.

As there can be multiple agreement conditions that can affect pricing, DPD do not provide web service that could provide exact price for specific parcel. If there is a need to reinvoice 3<sup>rd</sup> party or require pre-payment from 3<sup>rd</sup> party for delivery services, a shipping price list can be designed according to specifics of a customer business model.

## 9. Frequently asked questions (FAQ)

### **Can I create a shipment and print a label within one request?**

---

There is an option to request a parcel label within the shipment creation request. To do that an object *labelOptions* must be provided on shipment creation request (chapter 5.1). This array must contain the label creation parameters (chapter 5.2).

This will create a new block in shipment creation response named *shipmentLabels* that will contain the label data (chapter 5.2)

Example: chapter 12.2

### **How can I get the payerId value?**

---

There are methods that require *payerId* value. How to obtain it? Every DPD client has a unique identifier assigned. This identifier is listed in the agreement and can also be obtained via API (chapter 7.3) or by contacting DPD support.

Permissions can be managed in the DPD system under User management menu by the user who has permissions of an account's admin.

### **How often should I renew the Pickup points list?**

---

As DPD does not change the Pickup list very often, there's no need to load the Pickup list more than once per day (chapter 7.2)

## **How can I help identify the root cause for any issue?**

---

Please log all your API requests and responses. This information will be useful in case of any data exchange issues. When contacting DPD support regarding the web service issues, this information would help to identify the issue faster. Therefore, please make sure to include the following in the email:

- Full request that you sent (with URL and all parameters)
- Error message you've received in response

## 10. Response errors

Name	Type	Length	Description
type	string	250	URL that can be used as a GET request endpoint for accessing additional information about the issue (chapter 10.1) Request must contain the authorization token. No other parameters are needed.
title	string	250	Information about the cause of the issue.
detail	object	-	Detailed information about the cause of the issue. This will contain information on which block/parameter contains incorrect data. It can contain multiple entries (string, 250).
instance	string	50	Error message identifier, that could be used for support purposes.

### 10.1. Knowledge base

This method provides more detailed information on the specific problem from the DPD system's knowledge base.

Method: Endpoint must be taken from the error response *type* parameter.

Type: GET

#### Request:

No parameter is needed

#### Response:

Name	Type	Length	Description
problemTypeid	string	250	DPD internal identifier for specific issue, that will be included into endpoint URL.
title	string	250	Information about the cause. In case of a wrong request data, this will provide information on which block/parameter contains the incorrect data.
description	string	500	Detailed information on what can cause a specific issue and what can be done to solve it.

# 11. Contact information

If You have any technical issues, contact us at:

Estonia	Latvia	Lithuania
e-mail: <a href="mailto:ic@dpd.ee">ic@dpd.ee</a>	phone: +371 67387285 e-mail: <a href="mailto:support@dpd.lv">support@dpd.lv</a>	e-mail: <a href="mailto:support@dpd.lt">support@dpd.lt</a>

## 12. Examples

### 12.1. DPD Classic shipment creation request

```
[
  {
    "senderAddress": {
      "name": "Test Sender",
      "email": "example@example.com",
      "phone": "+37112345678",
      "street": "Uriekstes",
      "streetNo": "8a",
      "flatNo": null,
      "city": "Rīga",
      "postalCode": "1005",
      "country": "LV"
    },
    "receiverAddress": {
      "name": "Test Receiver",
      "email": "example@example.com",
      "phone": "+37112345678",
      "street": "Uriekstes",
      "streetNo": "8a",
      "flatNo": null,
      "city": "Rīga",
      "postalCode": "1005",
      "country": "LV"
    },
    "service": {
      "serviceAlias": "DPD CLASSIC"
    },
    "parcels": [
      {
        "weight": 1.2,
        "mpsReferences": ["Parcel reference 1", "Parcel reference 2", "Parcel reference 3", "Parcel reference 4"]
      },
      {
        "weight": 2.3
      }
    ],
    "pallets": [
      {
        "weight": 150,
        "type": "EUR",
        "mpsReferences": ["Pallet reference 1", "Parcel reference 2"]
      },
      {
        "weight": 250,
        "type": "FIN",
        "mpsReferences": ["Pallet reference 1", "Pallet reference 2", "Pallet reference 3", "Pallet reference 4"]
      }
    ],
    "shipmentReferences": ["Shipment reference 1", "Shipment reference 2", "Shipment reference 3", "Shipment reference 4"]
  }
]
```

## 12.2. B2C + COD shipment creation request (incl. label request)

```
[
  {
    "senderAddress": {
      "name": "Test Sender",
      "email": "example@example.com",
      "phone": "+37112345678",
      "street": "Uriekstes",
      "streetNo": "8a",
      "flatNo": null,
      "city": "Rīga",
      "postalCode": "1005",
      "country": "LV"
    },
    "receiverAddress": {
      "name": "Test Receiver",
      "email": "example@example.com",
      "phone": "+37112345678",
      "street": "Uriekstes",
      "streetNo": "8a",
      "flatNo": null,
      "city": "Rīga",
      "postalCode": "1005",
      "country": "LV"
    },
    "service": {
      "serviceAlias": "DPD B2C"
    },
    "additionalServices": [
      {
        "serviceAlias": "COD",
        "fields": {
          "cst_cod_value": "10",
          "cst_cod_currency": "EUR",
          "cst_cod_reference": "COD reference"
        }
      }
    ],
    "parcels": [
      {
        "weight": 10,
        "mpsReferences": ["Parcel reference 1"]
      }
    ],
    "shipmentReferences": ["Shipment reference 1"],
    "labelOptions": {
      "shipmentIds": [],
      "offsetPosition": 0,
      "downloadLabel": true,
      "emailLabel": false,
      "labelFormat": "image/png",
      "paperSize": "A6"
    }
  }
]
```

## 12.3. B2C + Evening shipment creation request

```
[
  {
    "senderAddress": {
      "name": "Test Sender",
      "email": "example@example.com",
      "phone": "+37112345678",
      "street": "Uriekstes",
      "streetNo": "8a",
      "flatNo": null,
      "city": "Rīga",
      "postalCode": "1005",
      "country": "LV"
    },
    "receiverAddress": {
      "name": "Test Receiver",
      "email": "example@example.com",
      "phone": "+37112345678",
      "street": "Uriekstes",
      "streetNo": "8a",
      "flatNo": null,
      "city": "Rīga",
      "postalCode": "1005",
      "country": "LV"
    },
    "service": {
      "serviceAlias": "DPD B2C"
    },
    "additionalServices": [
      {
        "serviceAlias": "Evening",
        "fields": {
          "cst_timeframe_service_timeframe_select": "18:00-22:00"
        }
      }
    ],
    "parcels": [
      {
        "weight": 10,
        "mpsReferences": ["Parcel reference 1"]
      }
    ]
  }
]
```

## 12.4. PDF A4 label creation request

```
{
  "shipmentIds": ["e32496bd-7303-4751-954e-2f886a44bbe8"],
  "parcelNumbers": [],
  "offsetPosition": 0,
  "downloadLabel": true,
  "emailLabel": false,
  "labelFormat": "application/pdf",
  "paperSize": "A4"
}
```

## 12.5. Courier request

```
{
  "pickupDate": "2021-03-23",
  "pickupTimeFrom": "12:00",
  "pickupTimeTo": "17:00",
  "address": {
    "name": "Company",
    "contactName": "John Doe",
    "email": "example@example.com",
    "phone": "+37122222222",
    "street": "Uriekstes",
    "streetNo": "8a",
    "flatNo": null,
    "city": "Riga",
    "postalCode": "1005",
    "country": "LV"
  },
  "messageToCourier": "Office entrance must be used",
  "parcel": {
    "count": 2,
    "weight": 2
  },
  "pallets": [
    {
      "weight": 300,
      "type": "EUR",
      "count": 2
    }
  ]
}
```