

DPD webservice interfaces

This document describes the prototype of web service interfaces in DPD Weblabel software.

1. Pickup service query interface

With this interface, external software can query the pickup package takeaway general information:

- possibility of same day pickup
- pickup starting time, ending time
- Minimum interval for pickup time

Web service URL: https://weblabel.dpd.hu/dpd_wow/pickup_info.php

Parameter submit method: post/get

Input parameters:

Name	Data
pickup_pcode	Pickup postal code, maximum length:9
pickup_date	Date of pickup
secret	pre-defined secret code: FcJyN7vU7WKPtUh7m1bx

RESULT:

The result will be a JSON text with the following parameters:

Name	Data	Format
sameday	Is same day allowed	True or false, with text: true , false
pickup_from_start	Time of first pickup from range	HH24:Mi 08:00
pickup_from_ends	Time of last pickup from range	HH24:Mi 16:00
minimum_interval	Minimum interval size in hours	4

```
{  
  "sameday" : "false",  
  "pickup_from_start" : "08:00",  
  "pickup_from_ends" : "16:00",  
  "minimum_interval" : "4"  
}
```

This means that the same day is not allowed, and the pickup from goes from 08:00-16:00, pickup till goes from 12:00-20:00, and minimum 4 hour difference must be between the pickup and delivery time.

SANDBOX USER DETAILS:

Name	Value
user	demo
password	o2Ijwe2!

2. Parcel generation interface:

With this interface, DPD WOW can save new parcels and get back the parcel number(s)

Web service URL: https://weblabel.dpd.hu/dpd_wow/parcel_import.php

Parameter submit method: post

Parameters submit information:

- method:post
- character-encoding: UTF-8
- content-type: "application/x-www-form-urlencoded"

Input parameters:

Name	Max length	Required	Data
username	20	yes	client's weblabel username
password	20	yes	client's weblabel password.
name1	40	Yes	Recipient name
name2	40	no	Recipient name2 (if needed.)
street	40	yes	Recipient street
city	40	yes	Recipient city
country	2	yes	Country code in iso2 character code format (Hungary is HU)
pcode	9	yes	Recipient postal code
email	100	no	Recipient email
phone	50	no	Recipient phone
idm_sms_number	30	no	SMS number for the IDM module
remark	100	no	Delivery instructions for courier (It won't appear all character (only 42) on the printed label)
weight	4	yes	parcel's weight
num_of_parcel	2	yes	number of parcel labels to be generated
order_number	20	yes	customer's order number
parcel_type	10	yes	Parcel type string: DPD Classic: D, DPD Classic COD: D-COD. D-PREDICT DPD Classic with predict service. Complete type list in DPD Weblabel manual.
parcel_cod_type	10	no	This field is made for the COD amount splitting.

			<p>The following values are accepted:</p> <ul style="list-style-type: none"> • avg (the amount of each parcel will be the average amount of the given cod_amount) • all (all parcel have the same amount which is in the cod_amount field) • firstly (only the first parcel will have the amount and the other will be simple DPD Classic parcel with the additional options given in the parcel_type field)" helyett "firstly (only the first parcel will have the COD amount and the others will have 0 COD amount)
cod_amount	10	no	Required for COD, in destination country currency.
cod_purpose	50	no	COD Purpose
cop_amount	10	no	Required for COP, in HUF
hins_amount	10	no	High insurance amount
hins_currency	3	no	The currency of the parcel if high insurance is active default value is "HUF"
hins_content	25	no	The parcel content
delta_service	1	no	Required for Collection Request (value:
who_sends	1	no	Required if Collection Request is chosen
who_prints	1	no	Required if Collection Request is chosen
same_day_pickup	1	no	Only required if Collection Request is chosen and the date is actual day
pickup_date	8	yes(delta)	Can be empty if same_day_pickup is set
sender_name	80	yes(delta)	Required for Collection Request
sender_street	80	yes(delta)	Required for Collection Request
sender_city	50	yes(delta)	Required for Collection Request
sender_pcode	7	yes(delta)	Required for Collection Request
sender_country	50	yes(delta)	Required for Collection Request
sender_phone	50	yes(delta)	Required for Collection Request
sender_email	50	yes(delta)	Required for Collection Request
sender_contact	50	yes(delta)	Required for Collection Request
sender_note	100	no	Required for Collection Request
parcelshop_id	4	no	Identifier of the Parcelshop
ekaer_number	20	no	Required for EKAER needed parcels
ekaer_price	10	no	Required for EKAER needed parcels
ekaer_risky	1	no	Required for EKAER needed parcels

EXAMPLE:

username=demo&password=o2Ijwe2!&sender_name=Test+Sender&sender_city=Budapest&sender_street=Importantt+street+13&sender_pcode=1112&name1=Janos+Szentesi&street=Kesmark+u+4&city=Budapest&country=HU&pcode=1158&weight=1&num_of_parcel=1&parcel_type=D

COLLECTION REQUEST EXAMPLE:

username=demo&password=o2Ijwe2!&name1=Test2+Kft&name2=&street=Kossuth+u.+1&city=Budapest&country=HU&pcode=1158&email=teszt%40teszt.hu&phone=&remark=&weight=11&num_of_parcel=1&parcel_type=D&delta_service=I&pickup_date=20150430&sender_name=teszter&sender_street=sdfsdf%20utca%203&sender_city=Budapest&sender_pcode=1078&sender_country=HU&sender_phone=061111111&sender_email=teszt@teszter.hu&sender_contact=Jozsi

PREDICT EXAMPLE:

username=demo&password=o2Ijwe2!&name1= Teszt+222&city=Budapest&street= Teszt+street+20&country=HU&phone=06308972133&pcode=1044&email=teszt@teszt.hu&idm_sms_number=06307069899&num_of_parcel=1&parcel_type=D-PREDICT

RESULT:

The result will be a JSON text with the following parameters:

Name	Data	Format
status	status of the parcel import	'ok': no problem occurred, 'err': some problem happened, please display errlog
errlog	Text message about the error	Display this message for the user
pl_number	DPD array of parcel number for the inserted parcels. filled only if no error happened.	16401234567890

```
{
  "status" : "ok",
  "errlog" : "",
  "pl_number" : ["16401234567890", "16401234567891"]
}
```

This means that the parcels are inserted, with these parcel number.

ERROR CODES:

The result will be a JSON text with the following parameters:

errorcode	meaning
Parcel import error:. row , xxxx xxxx, xx: xxxx;	Different errors in imported data which described in errorcode
'IV106', ' IV107'	Geocode errorcode. Contact the support

3. Parcel label interface

With this interface, external software can query the parcel label in PDF, or in the set label printer format.

Web service URL: https://weblabel.dpd.hu/dpd_wow/parcel_print.php

Parameters submit information:

- method:post
- character-encoding: UTF-8
- content-type: "application/x-www-form-urlencoded"

Input parameters:

Name	Max length	Required	Data
username	20	yes	client's weblabel username
password	20	yes	client's weblabel password
parcels	1500	Yes	Parcel number list, separated with ' '. Example: 16401234567890 16401234567891 16401122334455

Returns a parcel label stream with content type: application/octet-stream (Normally a PDF file, begins with %PDF).

On error it returns a json error message, with content type: application/json.

ERROR RESULT:

```
{"status" : "err", "errlog" : "parcel not found"}
```


4. Data transfer

With this interface, external software can instruct DPD weblabel to transfer parcel data to internal database. All the printed data will be transferred. This function should be called after all the parcels are printed, every day. Can be called more than once per day.

Web service [URL: https://weblabel.dpd.hu/dpdwow/parceldatasend.php](https://weblabel.dpd.hu/dpdwow/parceldatasend.php)

Parameters submit information:

- method:post
- character-encoding: UTF-8
- content-type: "application/x-www-form-urlencoded"

Input parameters:

Name	Max length	Required	Data
username	20	yes	client's weblabel username
password	20	yes	client's weblabel password

Return a json with an OK state, and the number affected of parcels in the parcels field, or error message on

RESULT:

```
{ "status": "ok", "errlog": "", "parcels": "13" }
```

5. Parcel status

This interface gives back the status of the chosen parcel in JSON format.

Web service URL: https://weblabel.dpd.hu/dpd_wow/parcel_status.php

Parameters submit information:

- method: post
- character-encoding: UTF-8
- content-type: "application/x-www-form-urlencoded"

Input parameters:

Name	Max length	Required	Data
secret	30	yes	The value must be the following: FcJyN7vU7WKPtUh7m1bx
parcel_number	14	yes	Number of the chosen parcel.

EXAMPLE REQUEST:

```
https://weblabel.dpd.hu/dpd_wow/parcel_status.php?secret=FcJyN7vU7WKPtUh7m1bx&parcel_number=16407425917162
```

Output parameters:

Name	Required	Data
status	yes	Indication if the status check is successful or not.
parcel_status	yes	Scan type and additional code in the following format: scantype/addcode For the list of possible scan type and additional code values, see the following sections.
event_date	Yes	Scan date and time in the following format: YYYY-MM-DD hh:mm:ss

EXAMPLE RESPONSE:

```
{"status":"ok","parcel_status":"Pickup scan","event_date":"2021-06-01 09:47:21"}
```

In case of bad input parameters, the following response is returned:

```
{"status":"err_400", "errmsg":"<error_message>"}
```

Where <error_message> indicates a specific error message.

5.1 Possible Scan Type Values

The following values can be returned for scantype (scan type):

- Info-scan
- Delivered to PUDO
- Collection request
- Delivered scan
- Drivers return
- Drivers pickup
- Infoscan
- Pickup scan
- Warehouse
- System return
- Loading to consolidated
- HUB-scan
- Loading
- Outbound
- Inbound exception
- Loading to storage unit
- Inbound scan
- Driver's return
- Out for delivery
- Consolidation

5.2 Possible Additional Code Values

The following values can be returned for addcode (additional code):

- misloaded - return into system
- misroute
- postcode error - return into system
- sent to NI-Lager - unidentified
- wrong address
- refused by consignee - damaged
- refused by consignee - not ordered
- refused by consignee
- refused by consignee - contents checked, found to be damaged
- return (pick-up) after wrong delivery
- consignee not present - informed first time
- self-collecting consignee
- broken in delivery drivers custody
- booked in/fixed date by consignor

- contents checked
- closed (e.g.department store) on inbound day
- lack of space in delivery car
- other parcel service provider
- consignment incomplete
- consignee on holiday
- non system traffic
- not delivered due to lack of time
- no export documents
- consignee not present, informed, on holiday
- 3 times consignee not present
- sorting error in recipient depot
- delivery to a third party with notification of delivery
- cash on delivery not paid - 2nd delivery
- Return third party
- customers returns
- Digicode required or access not possible
- P.O.Box address (for Sweden)
- private address
- Post Office
- no delivery due to unknown reason - system return
- ID check failed
- Parcel not ready for collection
- Parcel shipped / used by other means
- Customer had fewer parcels than contained in return order
- Date change in delivery
- regional holiday
- Address change
- COD data were missing
- no forwarding on inbound day
- Parcelshop delivery after 1st delivery attempt

6. Parcel delete

This interface deletes the chosen parcels, and returns status in JSON format.

Web service URL: <https://weblabel.dpd.hu/dpdwow/parceldelete.php>

Parameters submit information:

- method:post
- character-encoding: UTF-8
- content-type: "application/x-www-form-urlencoded"

Input parameters:

Name	Max length	Required	Data
username	20	yes	client's weblabel username
password	20	yes	client's weblabel password
parcels	200	Yes	Parcel number list, separated with ' '. Example: 16401234567890 16401234567891 16401122334455

ERROR RESULT:

```
{"status": "err", "errlog": "No parcel was selected"}
```

7. Parcelshop info

This interface returns parcelshop data in JSON format.

Web service URL: https://weblabel.dpd.hu/dpd_wow/parcelshop_info.php

Parameters submit information:

- method: post
- character-encoding: UTF-8
- content-type: "application/x-www-form-urlencoded"

Input parameters:

Name	Max length	Required	Data
username	20	yes	Client Weblabel username
password	20	yes	Client Weblabel password
id	20	no	Parcelshop ID
company	40	no	Parcelshop company
street	40	no	Parcelshop street address
city	40	no	Parcelshop city address
pcode	5	no	Parcelshop postal code
phone	30	no	Parcelshop phone number
email	30	no	Parcelshop email address

EXAMPLE REQUEST:

```
https://weblabel.dpd.hu/dpd_wow/parcelshop_info.php?username=demo&password=o2Ijwe2!&pcode=1118
```

Output parameters:

Name	Max length	Data
id	20	Parcelshop ID
company	40	Parcelshop company
street	40	Parcelshop street address
city	40	Parcelshop city address
pcode	5	Parcelshop postal code
phone	30	parcelshop phone number
email	30	Parcelshop email address
open_hours	N/A	Parcelshop open hours
gpslat	N/A	Latitude of the parcelshop position
gpslong	N/A	Longitude of the parcelshop position

EXAMPLE RESPONSE:

```
{"status":"ok", "errlog":"","parcelshops":
  [{"parcelshop_id":"HU20122",
    "company":"FotóBolt - Csepel Pláza",
    "city":"Budapest",
    "pcode":"1211",
    "street":"II. Rákóczi Ferenc út 154-170",
    "email":"","
    "phone":"","
    "open_hours":"10:00-12:00-12:00-20:00-10:00-12:00-
12:00-20:00-10:00-12:00-12:00-20:00-10:00-12:00-
12:00-20:00-10:00-12:00-12:00-20:00-10:00-12:00-
12:00-20:00-10:00-12:00-12:00-18:00"
    "gpslat":"47.423896"
    "gpslong":"19.068048"}]}
```

8. Service info

This interface gives back the available services in a postal code area in JSON format.

Web service URL: https://weblabel.dpd.hu/dpd_wow/service_info.php

Parameters submit information:

- method: post
- character-encoding: UTF-8
- content-type: "application/x-www-form-urlencoded"

Input parameters:

Name	Required	Data
username	yes	Client Weblabel username
password	yes	Client Weblabel password
dcountry	yes	Destination country code in ISO-2 alpha character code format (Hungary is HU).
dpostcode	yes	Destination postal code.

EXAMPLE REQUEST:

```
https://weblabel.dpd.hu/dpd_wow/service_info.php?username=demo
&password=o2Ijwe2!&dcountry=US&dpostcode=84000
```

Output parameters:

Name	Required	Data
soCode	yes	Numerical representation of a service code.
asCode	yes	Numerical representation of an additional service code.

EXAMPLE RESPONSE:

```
[
  { "soCode": "136", "asCode": null },
  { "soCode": "300", "asCode": null },
  { "soCode": "303", "asCode": null }
]
```

In case of bad input parameters, "HTTP 400 bad request" is returned.