

# DPD webservice interfaces

---

This document describes the prototype of web service interfaces between DPD Weblabel system and external (ERP, Warehousing) softwares.

## 1. Pickup postal code 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](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             |
| secret       | pre-defined secret code:<br>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                                      |

Example:

```
{ "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.

## 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](https://weblabel.dpd.hu/dpd_wow/parcel_import.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.  |
| 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  |
| remark          | 100        | no       | Delivery instructions for courier  |
| 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. Complete type list in DPD Weblabel manual. |
| parcel_cod_type | 10         | no       | This field is made for the COD   |

|                    |     |    |  |
|--------------------|-----|----|--|
|                    |     |    | amount splitting. The following values are accepted: 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) |
| 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   |
| delta_service      | 1   | no | Required for Delta service   |
| who_sends          | 1   | no | Required if Delta service is chosen  |
| who_prints         | 1   | no | Required if Delta service is chosen  |
| same_day_pickup    | 1   | no | Only required if Delta service is chosen and the date is actual day  |
| pickup_date        | 8   | no | Can be empty if same_day_pickup is set   |
| sender_name        | 80  | no | Required for Delta service   |
| sender_street      | 80  | no | Required for Delta service   |
| sender_city        | 50  | no | Required for Delta service   |
| sender_pcode       | 7   | no | Required for Delta service   |
| sender_countrycode | 50  | no | Required for Delta service   |
| sender_phone       | 50  | no | Required for Delta service   |
| sender_email       | 50  | no | Required for Delta service   |
| sender_contact     | 50  | no | Required for Delta service   |
| sender_note        | 100 | no | Required for Delta service   |

example: post this data to this url:

username=demo&password=o2ljwe2&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

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,<br>'err': some problem happened,<br>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 |
|-----------|--|----------------|

example:

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

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

### 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](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  | 200        | Yes      | Parcel number list, separated with ' '. Example:<br>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. Format:

```
{ "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/dpd\\_wow/parcel\\_datasend.php](https://weblabel.dpd.hu/dpd_wow/parcel_datasend.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 error, Format:

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

## 5. Parcel status

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

Web service URL: [https://weblabel.dpd.hu/dpd\\_wow/parcel\\_status.php](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:<br>FcJyN7vU7WKPtUh7m1bx |
| parcel_number | 14         | yes      | The chosen parcel's number.                              |

Return JSON message format:

```
{"parcel_status": "Shipped."}
```

## 6. Parcel delete

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

Web service URL: [https://weblabel.dpd.hu/dpd\\_wow/parcel\\_delete.php](https://weblabel.dpd.hu/dpd_wow/parcel_delete.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:<br>16401234567890 16401234567891 16401122334455 |

Return JSON message format:

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