***A red and white logo

Description automatically generated with medium confidenceA yellow sign with black text

Description automatically generated with medium confidence***

***HP Shipping Service API Technical documentation***

Table of contents

[1. Changelog 4](#_Toc162522639)

[2. Getting started 5](#_Toc162522640)

[3. API URL-s 6](#_Toc162522641)

[4. Ping API availabiliti 7](#_Toc162522642)

[Definition 7](#_Toc162522643)

[Response example 7](#_Toc162522644)

[5. Authorization – DX\_Security\_WebAPI 8](#_Toc162522645)

[Response parameters 8](#_Toc162522646)

[6. Shipment methods 9](#_Toc162522647)

[6.1 CreateShipmentOrders 9](#_Toc162522648)

[Definition 9](#_Toc162522649)

[Request parameters 9](#_Toc162522650)

[Request example 14](#_Toc162522651)

[Response example 15](#_Toc162522652)

[Response parameters 16](#_Toc162522653)

[6.2. CancelShipmentOrders 16](#_Toc162522654)

[Definition 16](#_Toc162522655)

[Request parameters 16](#_Toc162522656)

[Request example 17](#_Toc162522657)

[Response example 17](#_Toc162522658)

[Response parameters 17](#_Toc162522659)

[6.3. FetchShipmentStatus 18](#_Toc162522660)

[Definition 18](#_Toc162522661)

[Request parameters 18](#_Toc162522662)

[Request example 18](#_Toc162522663)

[Response example 18](#_Toc162522664)

[Response parameters 19](#_Toc162522665)

[6.4. FetchShippingLabels 20](#_Toc162522666)

[Definition 20](#_Toc162522667)

[Request parameters 20](#_Toc162522668)

[Request example 20](#_Toc162522669)

[Response example 21](#_Toc162522670)

[Response parameters 21](#_Toc162522671)

[Response example 21](#_Toc162522672)

[Response parameters 21](#_Toc162522673)

[7. Delivery point methods 22](#_Toc162522674)

[7.1. FetchParcelDeliveryPoint 22](#_Toc162522675)

[Definition 22](#_Toc162522676)

[Request parameters 22](#_Toc162522677)

[Request example 23](#_Toc162522678)

[Response example 23](#_Toc162522679)

[Appendix A – SCANS - STATUS AND REASONS FOR PARCEL COLLECTION NOT DONE AND NON-DELIVERY 26](#_Toc162522680)

[STATUS 26](#_Toc162522681)

[REASONS OF PARCEL COLLECTION IS NOT DONE (NPR) 26](#_Toc162522682)

[REASONS OF NON-DELIVERY (IND) 26](#_Toc162522683)

[Appendix B – API validation errors 28](#_Toc162522684)

[Appendix C – List of comments on scan NPA 32](#_Toc162522685)

[Appendix D – EasyReturn 35](#_Toc162522686)

[Return through service 39 35](#_Toc162522687)

[Return through service 40 35](#_Toc162522688)

[Return through additional service 45 36](#_Toc162522689)

[Appendix E – Examples of method calls in C#, JavaScript and PHP 37](#_Toc162522690)

# Changelog

|  |  |  |
| --- | --- | --- |
| Change No. | Since | Description |
| 1 | 11.11.2022. | Initial version |
| 2 | 12.12.2023. | Add „fetch“ methods  New examples |
| 3 | 19.02.2024. | „PRIPAK“ and „SPAK“ statuses were added |
| 4 | 28.03.2024. | Added services: 39 and 40  Addedd additional services: 11, 29, 31, 38, 45, 46  Ping method  Added fields: comment, collect\_by\_date, collect\_by\_time1, collect\_by\_time2 |
| 5 | 03.06.2024. | Added „payerCECODE“ parameter for third party payment client (BOX users usualy) |

# Getting started

This DXWebAPI service enables contract users of Hrvatska pošta d.d. to:

* create shipment orders
* cancel shipment orders
* get shipment status
* get shipping labels
* get parcel delivery point

Before using this DXWebAPI service you must be contract user of Hrvatska pošta d.d..

If you don’t have the required login credentials please contact Hrvatska pošta d.d..

For development purposes, on test environment, following credentials could be used, which are releted with 111111 CECODE (customer ID):

|  |  |
| --- | --- |
| **Username:** | testweb |
| **Password:** | testweb |

DXWebAPI service communication is designed for using via ***HTTPS***.

DXWebAPI service supports approach: ***REST*** (format ***JSON***).

Used HTTP response codes: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/

# API URL-s

|  |  |
| --- | --- |
| OKRUŽENJE | URL |
| TEST - DX\_Security\_WebAPI | https://dxwebapi**t**.posta.hr:9000/api/authentication/client\_auth |
| PRODUCTION - DX\_Security\_WebAPI | https://dxwebapi.posta.hr:9000/api/authentication/client\_auth |
| TEST – DXWebAPI | https://dxwebapi**t**.posta.hr:9020/api/{format}/{methodName} |
| PRODUCTION – DXWebAPI | https://dxwebapi.posta.hr:9020/api/{format}/{methodName} |

# Ping API availabiliti

### Definition

This method test API availability.

|  |  |
| --- | --- |
| Method | URL example - TEST |
| GET | https://dxwebapit.posta.hr:9020/api/ping |

### Response example

DxWeb API is alive 25.03.2024 14: 30: 27

# Authorization – DX\_Security\_WebAPI

To access the DXWebAPI service, it is necessary to make an authorization process by calling service:

* **DX\_Security\_WebAPI**

DX\_Security\_WebAPI gives a **bearer token** as a response which must then **be inserted into the *header of the DXWebAPI service call*.**

Bearer token duration is 4 hours.

For calling DX\_Security\_WebAPI service you will need:

* username
* password

provided by Hrvatska pošta d.d..

|  |  |
| --- | --- |
| Method | URL example - TEST |
| POST | https://dxwebapit.posta.hr:9000/api/authentication/client\_auth |

Request example:

{

    "username": "string",

    "password": "string"

}

Response example:

{

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiI2NzU5ODQiLCJuYmYiOjE2NjkyOTU1ODgsImV4cCI6MTY2OTMwOTk4OCwiaWF0IjoxNjY5Mjk1NTg4fQ.Gq0jwJNDfMH0KOSXpkYGBeMpNILtksVYXtTx4OVwcLw",

"expiresIn": "14400"

}

### Response parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| accessToken | String | Bearer token |
| expiresIn | Integer | Bearer token duration in **seconds** |

# Shipment methods

## CreateShipmentOrders

### Definition

This method is used for creating shipment orders.

|  |  |
| --- | --- |
| Method | URL example - TEST |
| POST | https://dxwebapit.posta.hr:9020/api/shipment/create\_shipment\_orders |

### Request parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| parcels | List<[ShipmentOrderData](#ShipmentOrderData)> | A list of shipment orders to create | YES |
| return\_address\_label | Bool | Flag for shipping labels in a response for created shipment orders in PDF format  Possible values:  **false** – NO  **true** – YES |  |

**Class ShipmentOrderData**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | | Description | REQUIRED |
| client\_reference\_number | | String(100) | Unique client ID of shipment order | YES |
| service | | Integer | Type of service  Possible values:  **24** – Pallet shipment  **26** – Package 24 D+1  **29** – Package 24 D+2  **32** – Package 24 D+3  **38** – Package 24 D+4  **39** – EasyReturn D+3 (1st option)  **40** – EasyReturn D+3 (2nd option) | YES |
| payed\_by | | Integer | Shipment payer  Possible values:  **1** – Sender  **2** – Recipient  **3** – Third party defined by „payerCECODE“ parameter  If NULL Sender is default. |  |
| delivery\_type | | Integer | Shipment delivery type  Possible values:  **1** – Address  **2** – Post office  **3** – Parcel locker | YES |
| payerCECODE | | String(6) | Enter if shipment has to be charged to other contract, not connected with authentication username | YES if payed\_by is set to 3 (third party) |
| value | | Decimal | Insured shipment value  Number of decimal places: 2 |  |
| payment\_value | | Decimal | Cash-on-delivery amount  Number of decimal places: 2 | YES if additional service 9 is chosen (Cash-on-delivery shipment) |
| collect\_by\_date | | System.Date | Collect by - date. It is sent if collection at the agreed time is required. |  |
| collect\_by\_time1 | | List<EnumTime> | Collect by - start time. It is sent if collection at the agreed time is required. |  |
| collect\_by\_time2 | | List<EnumTime> | Collect by - final time. It is sent if collection at the agreed time is required. |  |
| pickup\_type | | Integer | Shipment pickup type  Possible values:  **1** – Address  **2** – Post office  **3** – Parcel locker |  |
| pickup\_description | | String(500) | Additional description about pickup. |  |
| parcel\_size | | String | Parcel size  Possible values:  **X** – Parcel dimension XS  **S** – Parcel dimension S  **M** – Parcel dimension M  **L** – Parcel dimension L | YES if delivery\_type 3  is chosen  (Shipment is delivered to a Parcel locker) |
| comment | | String(500) | Optional customer comment on order. |  |
| reference\_field\_B | | String(100) | Order reference field B |  |
| reference\_field\_C | | String(100) | Order reference field C |  |
| reference\_field\_D | | String(100) | Order reference field D |  |
| sender | | [SenderData](#SenderData) | Sender's data | YES |
| recipient | | [RecipientData](#RecipientData) | Recipient's data | YES |
| additional\_services | | List<[AdditionalServiceData](#AdditionalServiceData)> | Additional services | YES  (can be empty) |
| packages | | List<[PackageData](#PackageData)> | Package list | YES |

**SenderData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| sender\_name | String(50) | Sender's name (person or oganization) | YES |
| sender\_phone | String(25) | Sender's phone number | YES |
| sender\_email | String(25) | Sender's email address |  |
| sender\_street | String(50) | Sender's name of the street | YES |
| sender\_hnum | String(10) | Sender's number of the house | YES  **IMPORTANT NOTE: If You can't extract house number from address, do not put zero here instead put dot („.“)** |
| sender\_hnum\_suffix | String(10) | Sender's suffix number of the house | YES |
| sender\_zip | String(5) | Sender's area zip code | YES |
| sender\_city | String(25) | Sender's town or village name | YES |
| sender\_pickup\_center | String(5) | Sender's post office code or parcel locker for pickup | YES if pickup\_type 2 or 3  is chosen |

**RecipientData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| recipient\_name | String(50) | Recipient's name (person or oganization) | YES |
| recipient\_phone | String(25) | Recipient's phone number | YES |
| recipient\_email | String(25) | Recipient's email address |  |
| recipient\_street | String(50) | Recipient's name of the street | YES |
| recipient\_hnum | String(10) | Recipient's number of the house | YES  **IMPORTANT NOTE: If You can't extract house number from address, do not put zero here instead put dot („.“)** |
| recipient\_hnum\_suffix | String(10) | Recipient's suffix number of the house | YES |
| recipient\_zip | String(5) | Recipient's area zip code | YES |
| recipient\_city | String(25) | Recipient's name of the town or village | YES |
| recipient\_delivery\_center | String(5) | Recipient's code of post office or parcel locker for delivery | YES if delivery\_type 2 or 3  is chosen |

**List AdditionalServiceData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| additional\_service\_id | Integer | ID of additional service  Possible values:  **1** – Personal delivery  **3** – Delivery on Saturdays  **4** – With the advice of receipt (AR)  **9** – Cash-on-delivery shipment  **11** – Collect on suterdays  **29** – Send a notification to the sender  **30** – Send a notification to the recipient  **31** – Send a email to sender  **32** – Send an e-mail to the recipient  **38** – Non standard format  **45** – EasyReturn (third option)  **46** – Sensitive content  **47** – Consolidated shipment |  |

**List PackageData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| barcode | String(20) | Barcode given by Hrvatska pošta d.d. (before creating shipment order) | YES if barcode\_type 0  is chosen |
| barcode\_type | Integer | Barcode type  Possible values:  **0** – Barcode given by Hrvatska pošta d.d. (before creating shipment order)  **1** – Barcode given from response after creating shipment order | YES |
| barcode\_client | String(500) | Client's own barcode | YES if barcode\_type 1  is chosen |
| weight | Decimal | Package weight in kilograms  Number of decimal places: 3 | YES |
| volume\_x | Integer | Parcel's dimensions in centimeters |  |
| volume\_y | Integer | Parcel's dimensions in centimeters. |  |
| volume\_z | Integer | Parcel's dimensions in centimeters. |  |

**List EnumTime (values for: collect\_by\_time\_1 and collect\_by\_time\_2 parameters)**

|  |
| --- |
| Parameter |
| TIME\_0800 |
| TIME\_0830 |
| TIME\_0900 |
| TIME\_0930 |
| TIME\_1000 |
| TIME\_1030 |
| TIME\_1100 |
| TIME\_1130 |
| TIME\_1200 |
| TIME\_1230 |
| TIME\_1300 |
| TIME\_1330 |
| TIME\_1400 |
| TIME\_1430 |
| TIME\_1500 |
| TIME\_1530 |
| TIME\_1600 |
| TIME\_1630 |
| TIME\_1700 |
| TIME\_1730 |
| TIME\_1800 |
| TIME\_1830 |
| TIME\_1900 |
| TIME\_1930 |
| TIME\_2000 |
| TIME\_2030 |
| TIME\_2100 |
| TIME\_2130 |
| TIME\_2200 |
| TIME\_2230 |
| TIME\_2300 |
| TIME\_2330 |

### Request example

{

    "parcels":[

        {

            "client\_reference\_number":"a00612566",

            "service":"38",

            "payed\_by":2,

            "delivery\_type":2,

            "value":99.00,

            "payment\_value":99.00,

            "pickup\_type":1,

            "parcel\_size":"S",

            "collect\_by\_date":"2024-03-24",

            "collect\_by\_time\_1":"TIME\_0900",

"collect\_by\_time\_2":"TIME\_1130",

            "comment": "This is test",

            "reference\_field\_B":"",

            "reference\_field\_C":"",

            "reference\_field\_D":"",

            "payer\_CECODE":"111111",

            "recipient":{

                "recipient\_name":"Pero Zvrndalo",

                "recipient\_phone":"0991234567",

                "recipient\_email":"bojan.kajfes@gmail.com",

                "recipient\_street":"SLOVENSKA",

                "recipient\_hnum":19,

                "recipient\_hnum\_suffix":"/2",

                "recipient\_zip":"10000",

                "recipient\_city":"Zagreb",

                "recipient\_delivery\_center":10000

            },

            "sender":{

                "sender\_name":"Hrvatska Pošta",

                "sender\_phone":"+38598271423",

                "sender\_email":"branislav.jankovic+ssp2l@vinted.com",

                "sender\_street":"Božidara Magovca 14 a",

                "sender\_hnum":".",

                "sender\_hnum\_suffix":"",

                "sender\_zip":"10000",

                "sender\_city":"Zagreb"

            },

            "additional\_services":[

                {

                    "additional\_service\_id": 9

                },

                {

                    "additional\_service\_id": 32

                }

            ],

            "packages":[

                {

                    "barcode":"",

                    "barcode\_type":1,

                    "barcode\_client":"123456",

                    "weight": 5,

                    "volume\_x": 5,

                    "volume\_y": 5,

                    "volume\_z": 10

                }

            ]

        }

    ],

    "return\_address\_label":**true**

}

### Response example

{

    "ShipmentOrdersList": [

        {

            "ClientReferenceNumber": "a00612566",

            "ResponseStatus": 0,

            "Packages": [

                {

                    "barcode": "LD055099035HR",

                    "barcode\_type": 1,

                    "barcode\_client": "123456",

                    "weight": 5.0,

                    "volume\_x": 5,

                    "volume\_y": 5,

                    "volume\_z": 10

                }

            ],

            "ErrorCode": **null**,

            "ErrorMessage": **null**,

            "MessageID": ""

        }

    ],

    "ShipmentsLabel": "base64\_string\_representation\_of\_pdf"

}

### Response parameters

**Class CreateShipmentOrdersMainResponse**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| ShipmentOrdersList | List<CreateShipmentOrdersResponse> | List of created or rejected shipment orders depending of potentional errors |
| ShipmentsLabel | String | Shipping labels for created shipment orders in PDF format |

**Class CreateShipmentOrdersResponse**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| ClientReferenceNumber | String | Unique client ID of shipment order |
| ResponseStatus | EnumResponseStatus | **0** – OK  **1** – ERROR |
| Packages | List<PackageData> | List of created packages |
| ErrorCode | Integer | Error code |
| ErrorMessage | String | Error message |
| MessageID | String | Error ID |

## 6.2. CancelShipmentOrders

### Definition

This method is used to cancel shipping orders.

|  |  |
| --- | --- |
| Method | URL example - TEST |
| POST | https://dxwebapit.posta.hr:9020/api/shipment/cancel\_shipment\_orders |

### Request parameters

**Class CancelOrdersData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| parcels | List<[CancelOrderData](#CancelOrderData)> | List of shipment orders to cancel | YES |

**Class** **CancelOrderData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| client\_reference\_number | String(100) | Unique client ID of shipment order | YES |

### Request example

{

    "parcels": [

        {

            "client\_reference\_number": "unique\_client\_id"

        },

        {

            "client\_reference\_number": "test12"

        }

    ]

}

### 

### Response example

[

    {

        "ClientReferenceNumber": "unique\_client\_id",

        "ResponseStatus": 0,

        "ErrorCode": **null**,

        "ErrorMessage": **null**

    },

    {

        "ClientReferenceNumber": "test12",

        "ResponseStatus": 0,

        "ErrorCode": **null**,

        "ErrorMessage": **null**

    }

]

### Response parameters

**Class CancelShipmentOrdersResponse**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| ClientReferenceNumber | String | Unique client ID of shipment order |
| ResponseStatus | EnumResponseStatus | **0** – OK  **1** – ERROR |
| ErrorCode | Integer | Error code |
| ErrorMessage | String | Error message |

## 

## 6.3. FetchShipmentStatus

### Definition

This method is used to retrieve shipping orders status. Method is simmilar to „GetShipmentStatus“ method mentioned above but it usese POST request instead GET.

|  |  |
| --- | --- |
| Method | URL example - TEST |
| POST | https://dxwebapit.posta.hr:9020/api/shipment/fetch\_shipment\_status |

### Request parameters

**Class BarcodesData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| barcodes | List<[BarcodeData](#BarcodeData)> | List of barcodes to retrieve status | YES |

**Class BarcodeData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| barcode | String(20) | Package barcode to retrieve status | YES |

### Request example

{    "barcodes": [

        {

            "barcode": "LE347087966HR"

        },

        {

            "barcode": "LE347087952HR"

        }

    ]

}

### Response example

[

    {

        "Barcode": "LE347087966HR",

        "PackageScansList": [

            {

                "Scan": "NOV",

                "ScanDescription": "Definirani podaci o pošiljci",

                "ScanTime": "2022-11-16T11:15:33.257",

                "Center": "ROOT",

                "Comment": **null**,

                "CommentNPA": "5000002"

            }

        ],

        "ResponseStatus": 0,

        "ErrorCode": **null**,

        "ErrorMessage": **null**

    },

    {

        "Barcode": "LE347087952HR",

        "PackageScansList": [

            {

                "Scan": "NOV",

                "ScanDescription": "Definirani podaci o pošiljci",

                "ScanTime": "2022-11-16T11:30:22.763",

                "Center": "ROOT",

                "Comment": **null**,

                "CommentNPA": "5000002"

            }

        ],

        "ResponseStatus": 0,

        "ErrorCode": **null**,

        "ErrorMessage": **null**

    }

]

### Response parameters

**Class GetShipmentStatusResponse**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Barcode | String | Package barcode |
| PackageScansList | List<[PackageScanData](#PackageScanData)> | List of package scans |
| ResponseStatus | EnumResponseStatus | **0** – OK  **1** – ERROR |
| ErrorCode | Integer | Error code |
| ErrorMessage | String | Error message |

**Class PackageScanData**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| Scan | String | Scan code  **List of scans:** [Appendix A](#DodatakA) |
| ScanDescription | String | Scan description |
| ScanTime | DateTime | Scan date and time |
| Center | String | Center in which the scan was made |
| Comment | String | Comment on scan |
| CommentNPA | String | Comment on scan NPA  **List of comments:** [Appendix B](#DodatakB) |

## 6.4. FetchShippingLabels

### Definition

This method is used for retrieval of package address labels.

It is possible to retrieve the address label by parcel barcodes **and/or** by client order number.

|  |  |
| --- | --- |
| Method | URL example - TEST |
| POST | https://dxwebapit.posta.hr:9020/api/shipment/fetch\_shipping\_labels |

### Request parameters

**Class LabelsData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| client\_reference\_number | String(100) | Unique client ID of shipment order | YES if barcodes is NULL or EMPTY |
| barcodes | List<[BarcodeData](#ClassBarcodeData)> | List of barcodes to retrieve labels | YES if CRN  is NULL or EMPTY |
| A4 | Bool | Flag for A4 paper size (4 labels on 1 page)  Possible values:  **false** – NO  **true** – YES  If NULL 0 is default. |  |
| format | Integer | Label format:  Possible values:  **1** – BASE64 representation of PDF  **2** – ZPL (for ZEBRA printers)  If NULL 1 is default. |  |

**Class BarcodeData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| barcode | String(20) | Package barcode |  |

### Request example

{

"client\_reference\_number": "test",

"barcodes": [

        {

            "barcode": "LE900001297HR"

        },

        {

            "barcode": "LE900001310HR"

        }

    ],

"A4":**false**

}

### Response example

{

    "PackageLabel": "base64\_string\_representation\_of\_pdf",

    "ResponseStatus": 0,

    "ErrorCode": **null**,

    "ErrorMessage": **null**

}

### Response parameters

**Class GetShippingLabelsResponse**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| PackageLabel | String | Package labels of created shipment orders in „base64“ string format or Zebra ZPL definition of label |
| ResponseStatus | EnumResponseStatus | **0** – OK  **1** – ERROR |
| ErrorCode | Integer | Error code |
| ErrorMessage | String | Error message |

### Response example

{

    "PackageLabel": "base64string",

    "ResponseStatus": 0,

    "ErrorCode": **null**,

    "ErrorMessage": **null**

}

### Response parameters

**Class GetShippingLabelsResponse**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| PackageLabel | String | Package labels of created shipment orders in „base64“ string format |
| ResponseStatus | EnumResponseStatus | **0** – OK  **1** – ERROR |
| ErrorCode | Integer | Error code |
| ErrorMessage | String | Error message |

# Delivery point methods

## FetchParcelDeliveryPoint

### Definition

This method is used to retrieve accurate and available information about working hours, addresses and other information of Croatian Post offices and parcel lockers.

|  |  |
| --- | --- |
| Method | URL example - TEST |
| POST | https://dxwebapit.posta.hr:9020/api/delivery\_point/fetch\_parcel\_delivery\_point |

### Request parameters

**Class ParcelDeliveryPointData**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | REQUIRED |
| facilityType | String | Parameter that filters data on post offices and parcel lockers  Possible values:  **ALL** – Retrieves post offices and parcel lockers  **PU** – Retrieves only post lockers  **PAK** – Retrieves only parcel lockers | YES |
| nextWeek | Integer | A parameter that determines whether the working hours for post offices will be retrieved for the current week or for the current and the following week  Possible values:  **0** – Retrieves post office working hours for the current week  **1** – Retrieves post office working hours for the current and the following week | YES |
| searchText | String | A parameter that filters the retrieved data by name, location, postal code or parcel locker code and street  Empty string does not filter results | YES |

### Request example

1. Example of a service call that retrieves all post offices and parcel lockers with business hours for the current week, without filtering the results

{

"facilityType": "ALL",

"nextWeek": 0,

"searchText": ""

}

2. Example of a service call that retrieves only parcel lockers

{

"facilityType": "PAK",

"nextWeek": 0,

"searchText": ""

}

3. Example of a service call that retrieves only post offices with business hours for the current and following week

{

"facilityType": "PU",

"nextWeek": 1,

"searchText": ""

}

4. Example of a service call that retrieves data by filtering

{

"facilityType": "ALL",

"nextWeek": 0,

"searchText": "62102"

}

{

"facilityType": "PU",

"nextWeek": 0,

"searchText": "Zagreb"

}

### Response example

"postaInfoList":

[

    {

        "type": "PU",

        "name": "nameo of the post office",

        "zip": "post office zip code and identifier",

        "city": "city name",

        "address": "street and house number",

        "tel": "contact phone number of post office",

        "fax": "contact fax number of post office",

        "geoLatitude": geo location latitude,

        "getLongitude": geo location longitude,

        "workingHours":

        [

            {

            "weekDay": 0,

            "date": "string",

            "openingTime":

                {

                "hour": 0,

                "minute": 0,

                "second": 0

                },

            "closingTime":

                {

                "hour": 0,

                "minute": 0,

                "second": 0

                },

            "closed": "string"

            }

        ]

    }

],

"paketomatInfoList":

[

    {

    "type": "PAK",

    "name": "nameo of parcel locker",

    "zip": "zip code"

    "city": "city name",

    "address": "street and house number where locker is located",

    "geoLat": geo location latitude,

    "getLng": geo location longitude,

    "code": " parcel locker identifier "

    }

]

}

The service returns two lists: one for post offices "postaInfoList" and one for parcel lockers "paketomatInfoList".

PostaInfoList

Within the list for post offices, the service returns information about working hours "workingHours" structured in such a way that working hours are returned separately for each day of the week. WorkingHours contains the following parameters within its structure:

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| weekDay | Integer | Day of week  1 – monday  .  .  .  7 - sunday |
| date | String | Designates the date |
| openingTime | String | Designates the opening time of the postal office |
| closingTime | String | Designates the closing time of the postal office |
| Closed | String | Designates whether the postal office is open or closed  0 – PO closed   1. – PO open |

# Appendix A – SCANS - STATUS AND REASONS FOR PARCEL COLLECTION NOT DONE AND NON-DELIVERY

## STATUS

* NOV – new parcel
* SPAK – the sender has inserted the parcel into the parcel locker
* ZAP – debit courier for parcel collection
* PRIPAK – parcel collection from parcel locker is done
* NPR – parcel collection is not done
* PRI – parcel collection is done
* ZPR – receipt of the parcel into the system-scanned in central HUB
* OTP – parcel transit between post depos
* ZOT – receipt of the parcel from other post depo
* ZAD – debit courier for parcel delivery
* IOD – delivery note in post depo-discharge of
* IND – non-delivery
* ODG – delivery delay for warehousing
* POV – parcel return
* IPP – finally delivery to the sender
* PRE – routed - change destination
* UKO – parcel loading to the container
* IZM – change parcel data.
* STO – Shipment canceled.

## REASONS OF PARCEL COLLECTION IS NOT DONE (NPR)

* 1050002 Recipient absent
* 1050006 Recipient absent
* 1050003 Unknown recipient
* 1050005 Parcel is not ready
* 1050007 Undoable
* 1050001 No parcel for collection
* 1050008 Sender info about collection attempt

## REASONS OF NON-DELIVERY (IND)

* 3900002 undeliverable
* 7010002 died
* 7010009 lost before delivery
* 7010010 lost after delivery
* 3900023 incomplete address
* 3900022 delivery agreed
* 3900025 second attempt delivery
* 3900004 refused by recipient
* 7010003 returned
* 3900027 the courier did not reach the address
* 3900017 recipient informed
* 7010004 for payment/delivery in post office
* 7010018 new attempt of delivery on order
* 3900007 recipient present, without money for payment
* 3900001 recipient absent
* 7010001 resend to
* 7010017 delivery in other post office
* 3900028 The user does not respond - informed
* 3900029 Wrong contact phone number of the user - notified
* 3900018 Package contents do not match
* 3900019 Damaged shipment
* 3900021 Issue at sorting center

# Appendix B – API validation errors

|  |  |
| --- | --- |
| **Validation error** | **Explanation** |
| Ako je tip usluge Besplatni povrat (Easy Return) D+3, platitelj mora biti primatelj i način plaćanja mora biti Ugovoreni (R1). | For Easy Return service the payer must be the recipient and legal entity, and payment type must be contracted. |
| Ako je tip usluge Besplatni povrat (Easy Return) D+3, tada naručitelj mora biti primatelj. | For Easy Return service the principal must be the recipient and legal entity. |
| Ako je tip usluge Besplatni povrat (Easy Return) D+3, tada naručitelj/platitelj ne smije biti fizička osoba. | For Easy Return service the principal and the recipient must be legal entity. |
| CollectByDate mora biti unesen jer je unesen CollectByTime1. | If pickup time is given, pickup date must also be given. |
| CollectByDate mora biti unesen jer je unesen CollectByTime2. | If pickup time is given, pickup date must also be given. |
| CollectByTime1 ili CollectByTime2 mora biti unesen jer je unesen CollectByDate. | If pickup date is given, pickup time must also be given. |
| Dozvoljen je upis samo jedne veličine pretinca. | Shipments with parcel locker pickup or delivery must have only one compartment size. |
| Duplikati u AdditionalServiceList. | The same additional service is given multiple times. |
| Duplikati u PackageList. | The same package is given multiple times. |
| Fizička osoba i blokirani klijent smiju platiti samo gotovinom ili virmanom. | If you get this message, it means you are blocked. Contact your sales representative. |
| Iznos CODAmount ne smije biti veći od 13 272,28 eura. | COD amount for COD without money order cannot be over 13,272.28 EUR. |
| Iznos CODAmount ne smije biti veći od 3318,07 eura. | COD amount for COD with postal money order cannot be over 3,318.07 EUR. |
| Iznos CODAmount veći je od dozvoljenog. | COD amount for COD is higher than allowed. |
| Ako je isporuka u paketomatu, ne smije biti navedena otkupnina (CODAmount). | Shipments with parcel locker delivery cannot have CODAmount. |
| Iznos Value ne smije biti veći od 13 272,28 eura. | Insured value cannot be over 13,272.28 EUR. |
| Lista PackageList ne smije biti prazna. | There are no packages in package list. |
| Ne smije biti naveden Barcode ako se koristi BarcodeType BARCODE\_CUSTOM. | If you use your own barcodes (BARCODE\_CUSTOM), Barcode field must be empty. |
| Nedostaje podatak: Barcode. | Missing data: Barcode. If you use BARCODE\_3P, Barcode field cannot be empty. |
| Nedostaje podatak: BarcodeClient. | Missing data: BarcodeClient. If you use BARCODE\_CUSTOM, BarcodeClient field cannot be empty. |
| Nedostaje podatak: Weight. | Missing data: Weight. If the field is specified, a value must be given (0 is allowed). |
| Nedostaje podatak: ShippingDocumentNumber. | Missing data: ShippingDocumentNumber. If additional service 5 is used, ShippingDocumentNumber must be given. |
| Nedostaje podatak: Dopunska usluga Id=9 (Otkupna pošiljka). | Missing data: Additional service Id=9 (COD shipment). If CODAmount is given, additional service 9 must be used. |
| Neispravan podatak: service. | Incorrect data: service. Mandatory field. |
| Neispravan podatak: payed\_by. | Incorrect data: payed\_by. |
| Neispravan podatak: delivery\_type. | Incorrect data: delivery\_type. Mandatory field. |
| Neispravan podatak: additional\_service\_id. | Incorrect data: additional\_service\_id. |
| Neispravan podatak: client\_reference\_number. | Incorrect data: client\_reference\_number. Mandatory field. |
| Neispravan podatak: sender\_name. | Incorrect data: sender\_name. Mandatory field. |
| Neispravan podatak: sender\_phone. | Incorrect data: sender\_phone. Mandatory field. |
| Neispravan podatak: sender\_city. | Incorrect data: sender\_city. Mandatory field. |
| Neispravan podatak: sender\_zip. | Incorrect data: sender\_zip. Mandatory field. |
| Neispravan podatak: sender\_street. | Incorrect data: sender\_street. Mandatory field. |
| Neispravan podatak: sender\_hnum. | Incorrect data: sender\_hnum. Mandatory field. **IMPORTANT NOTE: If You can't extract house number from address, do not put zero here instead put dot.** |
| Neispravan podatak: sender\_pick\_up\_center. | Incorrect data: sender\_pick\_up\_center. For post office or parcel locker delivery (pickup\_type 2 or 3) sender\_pick\_up\_center field cannot be empty. |
| Neispravan podatak: recipient\_name. | Incorrect data: recipient\_name. Mandatory field. |
| Neispravan podatak: recipient\_phone. | Incorrect data: recipient\_phone. Mandatory field. |
| Neispravan podatak: RecipientPhone | Incorrect data: recipient\_phone. Mandatory field. |
| Neispravan podatak: recipient\_city. | Incorrect data: recipient\_city. Mandatory field. |
| Neispravan podatak: recipient\_zip. | Incorrect data: recipient\_zip. Mandatory field. |
| Neispravan podatak: recipient\_street. | Incorrect data: recipient\_street. Mandatory field. |
| Neispravan podatak: recipient\_hnum. | Incorrect data: recipient\_hnum. Mandatory field. **IMPORTANT NOTE: If You can't extract house number from address, do not put zero here instead put dot.** |
| Neispravan podatak: recipient\_delivery\_center. | Incorrect data: recipient\_delivery\_center. Mandatory field if delivery\_type 2 or 3 is chosen. |
| Neispravan podatak: payment\_value. | Incorrect data: payment\_value. Mandatory field if additional service 9 is chosen (Cash-on-delivery shipment). |
| Neispravan podatak: PaymentType. | Incorrect data: PaymentType. Mandatory field. |
| Neispravan podatak: pickup\_type. | Incorrect data: pickup\_type. If sender\_pick\_up\_center is given, pickup\_type must be given (2 or 3). |
| Neispravan podatak: Barcode. | Incorrect tracking number. Tracking number must be according to S10 UPU standard. |
| Nedozvoljen tip usluge za prikup i isporuku u paketomatu. | Unavailable service (MainService) for parcel locker pickup and delivery. |
| Nedozvoljen tip usluge za prikup na adresi i isporuku u paketomatu. | Unavailable service (MainService) for address pickup and parcel locker delivery. |
| Nedozvoljena dopunska usluga Nestandardni format. | Unavailable additional service: Non-standard format. |
| Nedozvoljena dopunska usluga Otpremnica. | Unavailable additional service: Bill of lading. |
| Nedozvoljena dopunska usluga S povratnicom. | Unavailable additional service: With the advice of receipt (AR). |
| Nedozvoljena dopunska usluga Uručiti osobno. | Unavailable additional service: Personal delivery. |
| Nedozvoljeni CECODE za ovog korisnika. | Unavailable client code in PayerCECODE fields. |
| Nije u šifrarniku: PayerCECODE. | Data in PayerCECODE field is not allowed. |
| Nije u šifrarniku: PaymentType. | Data in PaymentType field is not allowed. |
| Nije u šifrarniku: PickUpType. | Data in PickUpType field is not allowed. |
| PayedBy mora biti unesen jer je unesen PayerCECODE. | Id PayerCECODE is given, PayedBy must also be given. Only if additional service 5 is used. |
| PayerCECODE mora biti unesen jer je unesen PayedBy. | Id PayedBy is given, PayerCECODE must also be given. Only if additional service 5 is used. |
| Podatak mora biti numerička vrijednost: ClientReferenceNumber. | ClientReferenceNumber must contain only numeric characters. |
| Prikup je na adresi: promijenite vrstu prikupa ili izbrišite PickUpCenter. | PickUpType 1 (address) and PickUpCenter are given. For address pickup remove PickUpCenter. For post office or parcel locker pickup use PickUpType 2 or 3. |
| Tip usluge Paletizirana pošiljka ne smije imati dopunsku uslugu Paleta. | Service Paletized shipment (MainService 24) cannot have additional service Pallet (AdditionaService 10). Additional service Pallet is not used any more. |
| Najava za paket je već zaprimljena. | Package with given reception number (Barcode) already exists in 3PDX. |
| Već postoji Shipment order s istim ClientReferenceNumber. | Shipment with given ClientReferenceNumber already exists in 3PDX. |
| Vrsta isporuke je na paketomatu, ali navedena poslovnica isporuke nije paketomat. | Parcel locker delivery is given (DeliveryType 3), but specified delivery center (DeliveryCenter) is not a parcel locker. |
| Vrsta isporuke je na paketomatu, ali paketomat je neaktivan. | Parcel locker delivery is given (DeliveryType 3), but specified parcel locker (DeliveryCenter) is not active. |
| Vrsta isporuke je u poštanskom uredu, ali navedena poslovnica isporuke nije dostavni/isporučni poštanski ured. | Post office delivery is given (DeliveryType 2), but specified delivery center (DeliveryCenter) is not a delivery post office. |
| Vrsta prikupa je na paketomatu, ali navedena poslovnica prikupa nije paketomat. | Parcel locker pickup is given (PickUpType 3), but specified pickup center (PickupCenter) is not a parcel locker. |
| Vrsta prikupa je na paketomatu, ali paketomat je neaktivan. | Parcel locker pickup is given (PickUpType 3), but specified parcel locker (PickupCenter) is not active. |
| Za MainService 39 polje ReferenceFieldB ne smije biti prazno. | For Easy Return service (MainService 39) link to an outbound shipment is mandatory (in the ReferenceFieldB field). |
| Za navedeni tip usluge i za prikup u paketomatu platitelj ne može biti primatelj. | For parcel locker pickup the payer cannot be the recipient (except for Easy Return). |
| Za navedeni tip usluge nije dozvoljen prikup u paketomatu ako je dostava na adresi. | Shipments with parcel locker pickup cannot have address delivery (except for Easy Return). |
| Za pošiljku s prikupom/isporukom u paketomatu mora se upisati veličina pretinca. | Shipments with parcel locker pickup or delivery must have compartment size specified (additional service 41, 42, 43 or 44). |
| Za prikup i isporuku u paketomatu platitelj mora biti pošiljatelj. | For parcel locker pickup and delivery the payer must be the sender. |
| Za prikup ili isporuku u paketomatu maksimalna masa paketa je 30 kg. | For parcel locker pickup or delivery maximum package weight is 30 kg. |
| Za prikup ili isporuku u paketomatu ne smije biti više od jednog paketa. | Shipments with parcel locker pickup or delivery can have only one package. |
| Za dopunsku uslugu Id=47 (Konsolidirana pošiljka) jedno od sljedećih polja mora biti uneseno: (reference\_field\_B, reference\_field\_C, reference\_field\_D). | For additional service Id=47 (Consolidated shipment) one of the following fields must be specified: (reference\_field\_B, reference\_field\_C, reference\_field\_D). |

# Appendix C – List of comments on scan NPA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **code** | **name** | **group\_code** | **event\_id** | **name2** |
| 1 | 1050001 | Parcels missing | NULL | 8 | Nerealiziran prikup |
| 2 | 1050002 | recipient absent | NULL | 8 | Nerealiziran prikup |
| 3 | 1050003 | unknown recipient | NULL | 8 | Nerealiziran prikup |
| 4 | 1050004 | Non standard shipment | NULL | 8 | Nerealiziran prikup |
| 5 | 1050005 | parcel is not ready | NULL | 8 | Nerealiziran prikup |
| 6 | 1050006 | recipient absent | NULL | 8 | Nerealiziran prikup |
| 7 | 1050007 | Not doable | NULL | 8 | Nerealiziran prikup |
| 8 | 2300001 | Parcel Shop closed | NULL | 47 | Unsuccessful handover to parcelShop |
| 9 | 2300002 | Parcel Shop full | NULL | 47 | Unsuccessful handover to parcelShop |
| 10 | 2300003 | Unsuccessful handover to parcel Shop | NULL | 47 | Unsuccessful handover to parcelShop |
| 11 | 2303001 | Parcel Shop full | NULL | 50 | Neuspješna predaja pošiljke u paketomat |
| 12 | 2303002 | Parcel Shop out of order | NULL | 50 | Neuspješna predaja pošiljke u paketomat |
| 13 | 2303003 | Unsuccessful handover to parcel Shop | NULL | 50 | Neuspješna predaja pošiljke u paketomat |
| 14 | 2306001 | Post office closed | NULL | 53 | Neuspješna predaja pošiljke u poštanski ured |
| 15 | 2306002 | Post office full | NULL | 53 | Neuspješna predaja pošiljke u poštanski ured |
| 16 | 2306003 | Unsuccessful handover to Post office | NULL | 53 | Neuspješna predaja pošiljke u poštanski ured |
| 17 | 3900001 | recipient absent | 1 | 106 | Pošiljka neuručiva |
| 18 | 3900002 | undeliverable | 2 | 106 | Pošiljka neuručiva |
| 20 | 3900004 | refused by recipient | 2 | 106 | Pošiljka neuručiva |
| 23 | 3900007 | recipient present, without money for payment | 2 | 106 | Pošiljka neuručiva |
| 33 | 3900017 | recipient informed | 4 | 106 | Pošiljka neuručiva |
| 37 | 3900021 | delivery sort error | 8 | 106 | Pošiljka neuručiva |
| 38 | 3900022 | delivery agreed | 9 | 106 | Pošiljka neuručiva |
| 39 | 3900023 | incomplete address | 10 | 106 | Pošiljka neuručiva |
| 41 | 3900025 | second attempt delivery | 1 | 106 | Pošiljka neuručiva |
| 43 | 3900027 | the courier did not reach the address | 14 | 106 | Pošiljka neuručiva |
| 44 | 5000001 | Defined shipment data in the call center | NULL | 129 | Definirani podaci o pošiljci |
| 45 | 5000002 | Defined shipment data through the call center | NULL | 129 | Definirani podaci o pošiljci |
| 46 | 5000003 | Defined shipment data through the EPK | NULL | 129 | Definirani podaci o pošiljci |
| 47 | 5000004 | Defined shipment data through the DP | NULL | 129 | Definirani podaci o pošiljci |
| 48 | 7010001 | resend to | NULL | 134 | MIPS Nadoslano u |
| 49 | 7010002 | died | NULL | 134 | MIPS Nadoslano u |
| 50 | 7010003 | returned | NULL | 134 | MIPS Nadoslano u |
| 51 | 7010004 | for payment/delivery in post office | NULL | 134 | MIPS Nadoslano u |
| 52 | 7010005 | for the district | NULL | 134 | MIPS Nadoslano u |
| 53 | 7010006 | For the postman | NULL | 134 | MIPS Nadoslano u |
| 54 | 7010007 | More obligated | NULL | 134 | MIPS Nadoslano u |
| 55 | 7010008 | Less obligated | NULL | 134 | MIPS Nadoslano u |
| 56 | 7010009 | lost before delivery | NULL | 134 | MIPS Nadoslano u |
| 57 | 7010010 | lost after delivery | NULL | 134 | MIPS Nadoslano u |
| 58 | 7010011 | Postage incorrectly charged | NULL | 134 | MIPS Nadoslano u |
| 59 | 7010012 | Defective money order/parcel | NULL | 134 | MIPS Nadoslano u |
| 60 | 7010013 | For military post | NULL | 134 | MIPS Nadoslano u |
| 61 | 7010014 | Disposition | NULL | 134 | MIPS Nadoslano u |
| 62 | 7010015 | the money order/shipment did not arrive | NULL | 134 | MIPS Nadoslano u |
| 63 | 7010016 | For post office box | NULL | 134 | MIPS Nadoslano u |
| 64 | 7010017 | delivery in other post office | NULL | 134 | MIPS Nadoslano u |
| 65 | 7010018 | new attempt of delivery on order | NULL | 134 | MIPS Nadoslano u |
| 66 | 7010019 | Request denied | NULL | 134 | MIPS Nadoslano u |
| 67 | 1001001 | Barcode scanned | NULL | 11 | Prikup pošiljaka kod pošiljatelja |
| 68 | 1001002 | Barcode not scanned | NULL | 11 | Prikup pošiljaka kod pošiljatelja |
| 69 | 3900028 | the user does not respond - informed | 1 | 106 | Pošiljka neuručiva |
| 70 | 3900029 | wrong contact phone number of the user - notified | 1 | 106 | Pošiljka neuručiva |
| 71 | 5002001 | Change of invoice type R1 - DP | NULL | 1310 | Ažuriranje pošiljke |
| 72 | 5000005 | Defined shipment data through CE | NULL | 129 | Definirani podaci o pošiljci |
| 73 | 5000006 | Defined shipment dana on parcel Shop | NULL | 129 | Definirani podaci o pošiljci |
| 74 | 5006001 | The sender will store the parcel in the compartment of the parcels Shop | NULL | 1320 | Najava pošiljke paketomatu |
| 75 | 5006002 | The postman will store the package in the compartment of the parcel machine | NULL | 1320 | Najava pošiljke paketomatu |
| 76 | 1080001 | Free return | NULL | 1320 | Najava pošiljke paketomatu |
| 78 | 1050008 | Notified | NULL | 8 | Nerealiziran prikup |
| 79 | 2210001 | Return parcel delivered to parcel Shop | NULL | 212 | Pošiljka predana u paketomat |
| 80 | 3900018 | Parcel content mismatch | NULL | 106 | Pošiljka neuručiva |
| 81 | 3900019 | Parcel damaged | NULL | 106 | Pošiljka neuručiva |
| 97 | 8010001 | Parcel return | NULL | 70 | Utovar dostavnog vozila |
| 98 | 2303004 | The dimensions of the parcel do not fit in the package machine | NULL | 50 | Neuspješna predaja pošiljke u paketomat |

# Appendix D – EasyReturn

Easy Return, i.e. a shipment by which the customer returns the goods to the sender (e.g. webshop), can be realized in the DxWeb service through the additional service 45 or services 39 and 40.

## Return through service 39

In the case of service 39, the user submits both outgoing and return shipments to the DxWeb service and creates and prints both address labels (address label for return shipments is not mandatory). The address label for the return shipment is attached to the goods (outgoing shipment).

Outgoing shipment submitted to the DxWeb service should be standard and **referenced\_field\_B** must contain reference to return shipment (e.g. order or invoice number in the user system), whereas return shipments must meet the following conditions:

* **payed\_by** must be 2
* **service** must be 39
* Outgoing and return shipment must have the same value in the **reference\_field\_B** parameter like it is in outgoing shipment

To carry out the Easy Return, the customer must do the following:

* to collect at the address
  + enter the shipment number at [**https://posiljka.posta.hr**](https://posiljka.posta.hr) (from any address label)
  + press the return button and select collection at the address
* to return at the post office
  + submit the return shipment at any post office with the attached return address label
* to return at the parcel locker
  + submit the return shipment at any parcel locker with the attached return address label

## Return through service 40

In the case of service 40, the user subsequently (at the customer’s request) submits return shipment to the DxWeb service and creates and sends the customer the address label or just shipment number.

Return shipment must meet the following conditions:

* **payed\_by** must be 2
* **service** must be 40
* **pickup\_type** must be 1
* address collection
  + fields **collect\_by\_date**, **collect\_by\_time\_1** and **collect\_by\_time\_2** must be filled
* post office or parcel locker drop-off
  + fields **collect\_by\_date**, **collect\_by\_time\_1** and **collect\_by\_time\_2** must be empty.

To carry out the Easy Return, the customer must do the following:

* address collection
  + the customer needs not doing anything additional
* post office drop-off
  + submit the return shipment at any post office with the attached return address label or shipment number
* parcel locker drop-off
  + submit the return shipment at any parcel locker with the attached return address label or shipment number
  + Easy Return instructions should be followed at the parcel locker

## Return through additional service 45

In the case of the additional service 45, the user only submits an outgoing shipment to the DxWeb service and creates the appropriate address label. The outgoing shipment must have the Additional Service 45.

The customer can find the return shipment number through the Croatian Post shipment tracking site [**https://posiljka.posta.hr**](https://posiljka.posta.hr). To carry out the Easy Return, the customer must do the following:

* to collect at the address
  + enter the shipment number at [**https://posiljka.posta.hr**](https://posiljka.posta.hr) (from the address label)
  + press the return button and select collection at the address
* to return at the post office
  + enter the shipment number at [**https://posiljka.posta.hr**](https://posiljka.posta.hr) (from the address label)
  + press the return button and find the return shipment number
  + submit the return shipment at any post office with the return shipment number
* to return at the parcel locker
  + enter the shipment number at [**https://posiljka.posta.hr**](https://posiljka.posta.hr) (from the address label)
  + press the return button and find the return shipment number
  + submit the return shipment at any parcel locker with the return shipment number

# Appendix E – Examples of method calls in C#, JavaScript and PHP

* 1. **ClientAuth - authorization**

**C# - RestSharp**

var options = new RestClientOptions("https://dxwebapit.posta.hr:9000")

{

  MaxTimeout = -1,

};

var client = new RestClient(options);

var request = new RestRequest("/api/authentication/client\_auth", Method.Post);

request.AddHeader("Content-Type", "application/json");

var body = @"{

" + "\n" +

@"    ""username"": ""testweb"",

" + "\n" +

@"    ""password"": ""testweb""

" + "\n" +

@"}";

request.AddStringBody(body, DataFormat.Json);

RestResponse response = await client.ExecuteAsync(request);

Console.WriteLine(response.Content);

**PHP – cURL**

**<?php**

$curl = curl\_init();

curl\_setopt\_array($curl, array(

  CURLOPT\_URL => 'https://dxwebapit.posta.hr:9000/api/authentication/client\_auth',

  CURLOPT\_RETURNTRANSFER => true,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => true,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "username": "testweb",

    "password": "testweb"

}',

  CURLOPT\_HTTPHEADER => array(

    'Content-Type: application/json'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

echo $response;

**Javascript - Fetch**

*var* myHeaders = new Headers();

myHeaders.append("Content-Type", "application/json");

*var* raw = JSON.stringify({

  "username": "testweb",

  "password": "testweb"

});

*var* requestOptions = {

  method: 'POST',

  headers: myHeaders,

  body: raw,

  redirect: 'follow'

};

fetch("https://dxwebapit.posta.hr:9000/api/authentication/client\_auth", requestOptions)

  .then(*response* *=>* response.text())

  .then(*result* *=>* *console*.log(result))

  .catch(*error* *=>* *console*.log('error', error));

* 1. **CreateShipemntOrders**

**C# - RestSharp**

var options = new RestClientOptions("https://dxwebapit.posta.hr:9020")

{

  MaxTimeout = -1,

};

var client = new RestClient(options);

var request = new RestRequest("/api/shipment/create\_shipment\_orders", Method.Post);

request.AddHeader("Content-Type", "application/json");

request.AddHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

var body = @"{

" + "\n" +

@"    ""parcels"":[

" + "\n" +

@"        {

" + "\n" +

@"            ""client\_reference\_number"":""Test 12367"",

" + "\n" +

@"            ""service"":""38"",

" + "\n" +

@"            ""payed\_by"":1,

" + "\n" +

@"            ""delivery\_type"":3,

" + "\n" +

@"            ""PayerCECODE"": ""123456"",

" + "\n" +

@"            ""value"":27.46,

" + "\n" +

@"            ""payment\_value"":27.46,

" + "\n" +

@"            ""pickup\_type"":1,

" + "\n" +

@"            ""parcel\_size"":""S"",

" + "\n" +

@"            ""reference\_field\_B"":"""",

" + "\n" +

@"            ""reference\_field\_C"":"""",

" + "\n" +

@"            ""reference\_field\_D"":"""",

" + "\n" +

@"            ""sender"":{

" + "\n" +

@"                ""sender\_name"":""Hrvatska Pošta d.d."",

" + "\n" +

@"                ""sender\_phone"":""017757397"",

" + "\n" +

@"                ""sender\_email"":""no-reply@posta.hr"",

" + "\n" +

@"                ""sender\_street"":""POŠTANSKA ULICA"",

" + "\n" +

@"                ""sender\_hnum"":""9"",

" + "\n" +

@"                ""sender\_hnum\_suffix"":"""",

" + "\n" +

@"                ""sender\_zip"":""10410"",

" + "\n" +

@"                ""sender\_city"":""Velika Gorica"",

" + "\n" +

@"                ""sender\_pick\_up\_center"":""""

" + "\n" +

@"            },

" + "\n" +

@"            ""recipient"":{

" + "\n" +

@"                ""recipient\_name"":""Pero Zvrndalo"",

" + "\n" +

@"                ""recipient\_phone"":""385998274492"",

" + "\n" +

@"                ""recipient\_email"":""bojan.kajfes3@gmail.com"",

" + "\n" +

@"                ""recipient\_street"":""Slovenska 19"",

" + "\n" +

@"                ""recipient\_hnum"":""."",

" + "\n" +

@"                ""recipient\_hnum\_suffix"":"""",

" + "\n" +

@"                ""recipient\_zip"":""10000"",

" + "\n" +

@"                ""recipient\_city"":""Zagreb"",

" + "\n" +

@"                ""recipient\_delivery\_center"":""61010""

" + "\n" +

@"            },

" + "\n" +

@"            ""additional\_services"":[

" + "\n" +

@"                {

" + "\n" +

@"                    ""additional\_service\_id"":9

" + "\n" +

@"                }

" + "\n" +

@"            ],

" + "\n" +

@"            ""packages"":[

" + "\n" +

@"                {

" + "\n" +

@"                    ""barcode"":"""",

" + "\n" +

@"                    ""barcode\_type"":1,

" + "\n" +

@"                    ""barcode\_client"":""123458"",

" + "\n" +

@"                    ""weight"":0.029

" + "\n" +

@"                }

" + "\n" +

@"            ]

" + "\n" +

@"        }

" + "\n" +

@"    ],

" + "\n" +

@"    ""return\_address\_label"":true

" + "\n" +

@"}";

request.AddStringBody(body, DataFormat.Json);

RestResponse response = await client.ExecuteAsync(request);

Console.WriteLine(response.Content);

**JavaScript - Fetch**

*var* myHeaders = new Headers();

myHeaders.append("Content-Type", "application/json");

myHeaders.append("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

*var* raw = JSON.stringify({

  "parcels": [

    {

      "client\_reference\_number": "Test 12367",

      "service": "38",

      "payed\_by": 1,

      "delivery\_type": 3,

      "PayerCECODE": "123456",

      "value": 27.46,

      "payment\_value": 27.46,

      "pickup\_type": 1,

      "parcel\_size": "S",

      "reference\_field\_B": "",

      "reference\_field\_C": "",

      "reference\_field\_D": "",

      "sender": {

        "sender\_name": "Hrvatska Pošta d.d.",

        "sender\_phone": "017757397",

        "sender\_email": "no-reply@posta.hr",

        "sender\_street": "POŠTANSKA ULICA",

        "sender\_hnum": "9",

        "sender\_hnum\_suffix": "",

        "sender\_zip": "10410",

        "sender\_city": "Velika Gorica",

        "sender\_pick\_up\_center": ""

      },

      "recipient": {

        "recipient\_name": "Pero Zvrndalo",

        "recipient\_phone": "385998274492",

        "recipient\_email": "bojan.kajfes3@gmail.com",

        "recipient\_street": "Slovenska 19",

        "recipient\_hnum": ".",

        "recipient\_hnum\_suffix": "",

        "recipient\_zip": "10000",

        "recipient\_city": "Zagreb",

        "recipient\_delivery\_center": "61010"

      },

      "additional\_services": [

        {

          "additional\_service\_id": 9

        }

      ],

      "packages": [

        {

          "barcode": "",

          "barcode\_type": 1,

          "barcode\_client": "123458",

          "weight": 0.029

        }

      ]

    }

  ],

  "return\_address\_label": true

});

*var* requestOptions = {

  method: 'POST',

  headers: myHeaders,

  body: raw,

  redirect: 'follow'

};

fetch("https://dxwebapit.posta.hr:9020/api/shipment/create\_shipment\_orders", requestOptions)

  .then(*response* *=>* response.text())

  .then(*result* *=>* *console*.log(result))

  .catch(*error* *=>* *console*.log('error', error));

**PHP – cURL**

**<?php**

$curl = curl\_init();

curl\_setopt\_array($curl, array(

  CURLOPT\_URL => 'https://dxwebapit.posta.hr:9020/api/shipment/create\_shipment\_orders',

  CURLOPT\_RETURNTRANSFER => true,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => true,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "parcels":[

        {

            "client\_reference\_number":"Test 12367",

            "service":"38",

            "payed\_by":1,

            "delivery\_type":3,

            "PayerCECODE": "123456",

            "value":27.46,

            "payment\_value":27.46,

            "pickup\_type":1,

            "parcel\_size":"S",

            "reference\_field\_B":"",

            "reference\_field\_C":"",

            "reference\_field\_D":"",

            "sender":{

                "sender\_name":"Hrvatska Pošta d.d.",

                "sender\_phone":"017757397",

                "sender\_email":"no-reply@posta.hr",

                "sender\_street":"POŠTANSKA ULICA",

                "sender\_hnum":"9",

                "sender\_hnum\_suffix":"",

                "sender\_zip":"10410",

                "sender\_city":"Velika Gorica",

                "sender\_pick\_up\_center":""

            },

            "recipient":{

                "recipient\_name":"Pero Zvrndalo",

                "recipient\_phone":"385998274492",

                "recipient\_email":"bojan.kajfes3@gmail.com",

                "recipient\_street":"Slovenska 19",

                "recipient\_hnum":".",

                "recipient\_hnum\_suffix":"",

                "recipient\_zip":"10000",

                "recipient\_city":"Zagreb",

                "recipient\_delivery\_center":"61010"

            },

            "additional\_services":[

                {

                    "additional\_service\_id":9

                }

            ],

            "packages":[

                {

                    "barcode":"",

                    "barcode\_type":1,

                    "barcode\_client":"123458",

                    "weight":0.029

                }

            ]

        }

    ],

    "return\_address\_label":true

}',

  CURLOPT\_HTTPHEADER => array(

    'Content-Type: application/json',

    'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

echo $response;

* 1. **CancelShipmentOrders**

**C# - RestSharp**

var options = new RestClientOptions("https://dxwebapit.posta.hr:9020")

{

  MaxTimeout = -1,

};

var client = new RestClient(options);

var request = new RestRequest("/api/shipment/cancel\_shipment\_orders", Method.Post);

request.AddHeader("Content-Type", "application/json");

request.AddHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

var body = @"{

" + "\n" +

@"    ""parcels"": [

" + "\n" +

@"        {

" + "\n" +

@"            ""client\_reference\_number"": ""Test 12366""

" + "\n" +

@"        },

" + "\n" +

@"        {

" + "\n" +

@"            ""client\_reference\_number"": ""Test 12365""

" + "\n" +

@"        }

" + "\n" +

@"    ]

" + "\n" +

@"}

" + "\n" +

@"";

request.AddStringBody(body, DataFormat.Json);

RestResponse response = await client.ExecuteAsync(request);

Console.WriteLine(response.Content);

**JavaScrtipt - Fetch**

*var* myHeaders = new Headers();

myHeaders.append("Content-Type", "application/json");

myHeaders.append("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

*var* raw = JSON.stringify({

  "parcels": [

    {

      "client\_reference\_number": "Test 12366"

    },

    {

      "client\_reference\_number": "Test 12365"

    }

  ]

});

*var* requestOptions = {

  method: 'POST',

  headers: myHeaders,

  body: raw,

  redirect: 'follow'

};

fetch("https://dxwebapit.posta.hr:9020/api/shipment/cancel\_shipment\_orders", requestOptions)

  .then(*response* *=>* response.text())

  .then(*result* *=>* *console*.log(result))

  .catch(*error* *=>* *console*.log('error', error));

**PHP – cURL**

**<?php**

$curl = curl\_init();

curl\_setopt\_array($curl, array(

  CURLOPT\_URL => 'https://dxwebapit.posta.hr:9020/api/shipment/cancel\_shipment\_orders',

  CURLOPT\_RETURNTRANSFER => true,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => true,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "parcels": [

        {

            "client\_reference\_number": "Test 12366"

        },

        {

            "client\_reference\_number": "Test 12365"

        }

    ]

}

',

  CURLOPT\_HTTPHEADER => array(

    'Content-Type: application/json',

    'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

echo $response;

* 1. **FetchShipmentStatus**

**C# - RestSharp**

var options = new RestClientOptions("https://10.0.226.46:9020")

{

  MaxTimeout = -1,

};

var client = new RestClient(options);

var request = new RestRequest("/api/shipment/fetch\_shipment\_status", Method.Post);

request.AddHeader("Content-Type", "application/json");

request.AddHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

var body = @"{

" + "\n" +

@"    ""barcodes"": [

" + "\n" +

@"      {

" + "\n" +

@"        ""barcode"": ""LD055029851HR""

" + "\n" +

@"      }

" + "\n" +

@"    ]

" + "\n" +

@"  }

" + "\n" +

@"";

request.AddStringBody(body, DataFormat.Json);

RestResponse response = await client.ExecuteAsync(request);

Console.WriteLine(response.Content);

**JavaScript - Fetch**

*var* myHeaders = new Headers();

myHeaders.append("Content-Type", "application/json");

myHeaders.append("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

*var* raw = JSON.stringify({

  "barcodes": [

    {

      "barcode": "LD055029851HR"

    }

  ]

});

*var* requestOptions = {

  method: 'POST',

  headers: myHeaders,

  body: raw,

  redirect: 'follow'

};

fetch("https://10.0.226.46:9020/api/shipment/fetch\_shipment\_status", requestOptions)

  .then(*response* *=>* response.text())

  .then(*result* *=>* *console*.log(result))

  .catch(*error* *=>* *console*.log('error', error));

**PHP – cURL**

**<?php**

$curl = curl\_init();

curl\_setopt\_array($curl, array(

  CURLOPT\_URL => 'https://10.0.226.46:9020/api/shipment/fetch\_shipment\_status',

  CURLOPT\_RETURNTRANSFER => true,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => true,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "barcodes": [

      {

        "barcode": "LD055029851HR"

      }

    ]

  }

',

  CURLOPT\_HTTPHEADER => array(

    'Content-Type: application/json',

    'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

echo $response;

* 1. **FetchShippingLabels**

**C# - RestSharp**

var options = new RestClientOptions("https://dxwebapit.posta.hr:9020")

{

  MaxTimeout = -1,

};

var client = new RestClient(options);

var request = new RestRequest("/api/shipment/fetch\_shipping\_labels", Method.Post);

request.AddHeader("Content-Type", "application/json");

request.AddHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

var body = @"{

" + "\n" +

@"    ""barcodes"": [

" + "\n" +

@"        {

" + "\n" +

@"            ""barcode"": ""LD055017663HR""

" + "\n" +

@"        }

" + "\n" +

@"    ],

" + "\n" +

@"    ""A4"": false,

" + "\n" +

@"    ""format"": 2

" + "\n" +

@"}";

request.AddStringBody(body, DataFormat.Json);

RestResponse response = await client.ExecuteAsync(request);

Console.WriteLine(response.Content);

**JavaScript - Fetch**

*var* myHeaders = new Headers();

myHeaders.append("Content-Type", "application/json");

myHeaders.append("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

*var* raw = JSON.stringify({

  "barcodes": [

    {

      "barcode": "LD055017663HR"

    }

  ],

  "A4": false,

  "format": 2

});

*var* requestOptions = {

  method: 'POST',

  headers: myHeaders,

  body: raw,

  redirect: 'follow'

};

fetch("https://dxwebapit.posta.hr:9020/api/shipment/fetch\_shipping\_labels", requestOptions)

  .then(*response* *=>* response.text())

  .then(*result* *=>* *console*.log(result))

  .catch(*error* *=>* *console*.log('error', error));

**PHP – cURL**

**<?php**

$curl = curl\_init();

curl\_setopt\_array($curl, array(

  CURLOPT\_URL => 'https://dxwebapit.posta.hr:9020/api/shipment/fetch\_shipping\_labels',

  CURLOPT\_RETURNTRANSFER => true,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => true,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "barcodes": [

        {

            "barcode": "LD055017663HR"

        }

    ],

    "A4": false,

    "format": 2

}',

  CURLOPT\_HTTPHEADER => array(

    'Content-Type: application/json',

    'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

echo $response;

* 1. **FetchParcelDeliveryPoint**

**C# - RestSharp**

var options = new RestClientOptions("https://dxwebapit.posta.hr:9020")

{

  MaxTimeout = -1,

};

var client = new RestClient(options);

var request = new RestRequest("/api/delivery\_point/fetch\_parcel\_delivery\_point", Method.Post);

request.AddHeader("Content-Type", "application/json");

request.AddHeader("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

var body = @"{

" + "\n" +

@"    ""facilityType"": ""PAK"",

" + "\n" +

@"    ""nextWeek"": 0,

" + "\n" +

@"    ""searchText"": """"

" + "\n" +

@"}

" + "\n" +

@"";

request.AddStringBody(body, DataFormat.Json);

RestResponse response = await client.ExecuteAsync(request);

Console.WriteLine(response.Content);

**JavaScript - Fetch**

*var* myHeaders = new Headers();

myHeaders.append("Content-Type", "application/json");

myHeaders.append("Authorization", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");

*var* raw = JSON.stringify({

  "facilityType": "PAK",

  "nextWeek": 0,

  "searchText": ""

});

*var* requestOptions = {

  method: 'POST',

  headers: myHeaders,

  body: raw,

  redirect: 'follow'

};

fetch("https://dxwebapit.posta.hr:9020/api/delivery\_point/fetch\_parcel\_delivery\_point", requestOptions)

  .then(*response* *=>* response.text())

  .then(*result* *=>* *console*.log(result))

  .catch(*error* *=>* *console*.log('error', error));

**PHP – cURL**

**<?php**

$curl = curl\_init();

curl\_setopt\_array($curl, array(

  CURLOPT\_URL => 'https://dxwebapit.posta.hr:9020/api/delivery\_point/fetch\_parcel\_delivery\_point',

  CURLOPT\_RETURNTRANSFER => true,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => true,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "facilityType": "PAK",

    "nextWeek": 0,

    "searchText": ""

}

',

  CURLOPT\_HTTPHEADER => array(

    'Content-Type: application/json',

    'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJDRUNPREUiOiIxMTExMTEiLCJuYmYiOjE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiaWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

echo $response;