



HP Shipping Service API

Technical documentation

Table of contents

1. Changelog	4
2. Getting started.....	5
3. API URL-s.....	6
4. Ping API availability.....	7
Definition	7
Response example	7
5. Authorization – DX_Security_WebAPI	8
Response parameters	8
6. Shipment methods	9
6.1 CreateShipmentOrders	9
Definition	9
Request parameters	9
Request example	15
Response example	16
Response parameters	17
6.2. CancelShipmentOrders.....	17
Definition	17
Request parameters	17
Request example	18
Response example.....	18
Response parameters	18
6.3. FetchShipmentStatus	19
Definition	19
Request parameters	19
Request example	19
Response example.....	19
Response parameters	20
6.4. FetchShippingLabels	21
Definition	21
Request parameters	21
Request example	21
Response example.....	22

Response parameters	22
Response example	22
Response parameters	22
7. Delivery point methods	23
7.1. FetchParcelDeliveryPoint.....	23
Definition	23
Request parameters	23
Request example	24
Response example.....	24
Appendix A – SCANS - STATUS AND REASONS FOR PARCEL COLLECTION NOT DONE AND NON-DELIVERY	27
STATUS	27
REASONS OF PARCEL COLLECTION IS NOT DONE (NPR)	27
REASONS OF NON-DELIVERY (IND)	27
Appendix B – API validation errors	29
Appendix C – List of comments on scan NPA	33
Appendix D – EasyReturn.....	36
Return through service 39.....	36
Return through service 40.....	36
Return through additional service 45	37
Appendix E – Examples of method calls in C#, JavaScript and PHP.....	39

1. 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 Added 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 usually)
6	17.07.2024.	Described parcel dimensions in cm, payed_by: 3 option removed from API
7	12.08.2024.	PAK status described
8	18.09.2024.	Recipient mobile phone validation explained
9	8.10.2024.	Removed service: 24 – pallet shipment Added service: 46 – pallet shipment D+5 Max insured value – 13,300.00 EUR Max COD amount – 13,300.00 EUR
10	21.10.2024.	Length of the parameter „recipient_street“ increased from 50 to 75 characters. Added new additional service: 54 – Empty Euro pallet return
11	9.12.2024.	New barcode format added, type 3 - BASE64 representation of PDF (CODE 128)

2. Getting started

This DXWebAPI service enables contract users of HP - 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 HP - Hrvatska pošta d.d.. If you don't have the required login credentials please contact HP - Hrvatska pošta d.d..

For development purposes, on test environment, following credentials could be used, which are related 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/>

3. API URL-s

ENVIRONMENT	URL
TEST - DX_Security_WebAPI	https://dxwebapit.posta.hr:9000/api/authentication/client_auth
PRODUCTION - DX_Security_WebAPI	https://dxwebapi.posta.hr:9000/api/authentication/client_auth
TEST – DXWebAPI	https://dxwebapit.posta.hr:9020/api/{format}/{methodName}
PRODUCTION – DXWebAPI	https://dxwebapi.posta.hr:9020/api/{format}/{methodName}

4. Ping API availability

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
```

5. 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 HP - 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.eyJDRUNPREUiOiI2NzU5ODQiLCJmYXQiOiJlbnV4cCI6MTY2OTMwOTk0CwiaWF0IjoxNjY5Mjk1NTg4fQ.Gq0jwJNdfMH0KOSXpkYGBemPnILtksVYXtTx40VwcLw",
  "expiresIn": "14400"
}
```

Response parameters

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

6. Shipment methods

6.1 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 >	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 <u>Possible values:</u> 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 <u>Possible values:</u> 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) 46 – Pallet shipment D+5	YES
payed_by	Integer	Shipment payer <u>Possible values:</u> 1 – Sender 2 – Recipient	
delivery_type	Integer	Shipment delivery type <u>Possible values:</u> 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	
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 <u>Possible values:</u> 1 – Address 2 – Post office 3 – Parcel locker	
pickup_description	String(500)	Additional description about pickup.	
parcel_size	String	Parcel size <u>Possible values:</u> X – Parcel dimension XS (9x16x64 cm) S – Parcel dimension S (9x38x64 cm) M – Parcel dimension M (19x38x64 cm) L – Parcel dimension L (39x38x64 cm)	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	Sender's data	YES
recipient	RecipientData	Recipient's data	YES
additional_services	List< AdditionalServiceData >	Additional services	YES (can be empty)
packages	List< PackageData >	Package list	YES

SenderData

Parameter	Type	Description	REQUIRED
sender_name	String(50)	Sender's name (person or organization)	YES
sender_phone	String(25)	Sender's phone number	YES
sender_email	String(100)	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 NOTE: On test use 61004 code for parcel_locker other may be inactive on test environment

RecipientData

Parameter	Type	Description	REQUIRED
recipient_name	String(50)	Recipient's name (person or organization)	YES
recipient_phone	String(25)	Recipient's phone number	<p>YES</p> <p>IMPORTANT NOTE: In case of delivery_type => 3, e.g. parcel locker delivery, phone number will be validated by next rules:</p> <ul style="list-style-type: none"> valid recipient Croatian mobile phone number must be specified <ul style="list-style-type: none"> landline numbers are not allowed foreign numbers are not allowed the number must be in correct format <ul style="list-style-type: none"> Croatian prefix (+385, 00385 or 385); not mandatory Croatian mobile prefix, with or without zero (091, 092, 095,

			<p>097, 098 or 099)</p> <ul style="list-style-type: none"> ▪ six or seven digits <p>examples of correct format: +385991234567 00385991234567 0991234567</p>
recipient_email	String(100)	Recipient's email address	
recipient_street	String(75)	Recipient's name of the street	YES
recipient_hnum	String(10)	Recipient's number of the house	<p>YES</p> <p>IMPORTANT NOTE: If You can't extract house number from address, do not put zero here instead put dot („.")</p>
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	<p>YES if delivery_type 2 or 3 is chosen</p> <p>NOTE: On test use 61004 code for parcel_locker other may be inactive on test environment</p>

List AdditionalServiceData

Parameter	Type	Description	REQUIRED
additional_service_id	Integer	<p>ID of additional service</p> <p><u>Possible values:</u></p> <p>1 – Personal delivery</p> <p>3 – Delivery on Saturdays</p> <p>4 – With the advice of receipt (AR)</p> <p>9 – Cash-on-delivery shipment</p> <p>11 – Collect on suterdays</p> <p>29 – Send a notification to the sender</p> <p>30 – Send a notification to the recipient</p> <p>31 – Send a email to sender</p> <p>32 – Send an e-mail to the recipient</p> <p>38 – Non standard format</p> <p>45 – EasyReturn (third option)</p> <p>46 – Sensitive content</p> <p>47 – Consolidated shipment</p> <p>54 – Empty Euro pallet return</p>	

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 <u>Possible values:</u> 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": "",
      "payerCECODE": "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+ssp21@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 potential 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 >	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 similar to „GetShipmentStatus“ method mentioned above but it uses 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 >	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 >	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
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

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 >	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) <u>Possible values:</u> false – NO true – YES If NULL 0 is default.	
format	Integer	Label format: <u>Possible values:</u> 1 – BASE64 representation of PDF (CODE 39) 2 – ZPL (for ZEBRA printers) If NULL 1 is default. 3 – BASE64 representation of PDF (CODE 128)	

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

7. Delivery point methods

7.1. 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 <u>Possible values:</u> 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 <u>Possible values:</u> 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": "name 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 "postalInfoList" and one for parcel lockers "paketomatInfoList".

PostalInfoList

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 – the parcel has been collected from the parcel locker (collected by carrier)
- 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 – final delivery to the destination
- IND – non-delivery
- ODG – delivery delay for warehousing
- POV – parcel return
- IPP – final delivery to the sender
- PRE – routed - change destination (change of address/change to delivery)
- UKO – parcel loading to the container
- IZM – change parcel data.
- STO – shipment canceled.
- PAK – parcel delivered in the parcel locker
- NIS – Undelivered shipment

REASONS OF PARCEL COLLECTION IS NOT DONE (NPR)

- 1050001 no parcel for collection
- 1050002 recipient absent
- 1050003 unknown recipient
- 1050004 non-standard package
- 1050005 parcel is not ready
- 1050006 recipient absent
- 1050007 undoable
- 1050008 sender informed about collection attempt

REASONS OF NON-DELIVERY (IND)

- 3900001 recipient absent
- 3900002 undeliverable
- 3900003 passed away
- 3900004 refused by recipient
- 3900005 didn't ask for an order
- 3900006 moved away
- 3900007 recipient present, without money for payment
- 3900017 recipient informed
- 3900018 package contents do not match
- 3900019 damaged shipment
- 3900021 issue at sorting center
- 3900022 delivery agreed

- 3900023 incomplete address
- 3900024 unknown reason
- 3900025 second attempt delivery
- 3900027 the courier did not reach the address
- 3900028 the user does not respond - informed
- 3900029 wrong contact phone number of the user - notified
- 7010001 resend to
- 7010002 died
- 7010003 returned
- 7010004 for payment/delivery in post office
- 7010009 lost before delivery
- 7010010 lost after delivery
- 7010017 delivery in other post office
- 7010018 new attempt of delivery on order

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.
collect_by_date mora biti unesen jer je unesen collect_by_time1.	If pickup time is given, pickup date must also be given.
collect_by_date mora biti unesen jer je unesen collect_by_time2.	If pickup time is given, pickup date must also be given.
collect_by_time1 ili collect_by_time2 mora biti unesen jer je unesen collect_by_date.	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 13 300,00 eura.	COD amount for COD without money order cannot be over 13,300.00 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 13 300,00 eura.	Insured value cannot be over 13,300.00 EUR.
Lista PackageList ne smije biti prazna.	There are no packages in package list.
Ne smije biti naveden Barcode ako se koristi barcode_type 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_pickup_center. For post office or parcel locker delivery (pickup_type 2 or 3) sender_pickup_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 (main service) for parcel locker pickup and delivery.
Nedozvoljen tip usluge za prikup na adresi i isporuku u paketomatu.	Unavailable service (main service) 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 šifrniku: PayerCECODE.	Data in PayerCECODE field is not allowed.
Nije u šifrniku: PaymentType.	Data in PaymentType field is not allowed.
Nije u šifrniku: 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: client_reference_number.	client_reference_number must contain only numeric characters.
Prikup je na adresi: promijenite vrstu prikupa ili izbrišite sender_pickup_center.	pickup_type 1 (address) and sender_pickup_center are given. For address pickup remove sender_pickup_center. For post office or parcel locker pickup use pickup_type 2 or 3.
Tip usluge Paletizirana pošiljka ne smije imati dopunsku uslugu Paleta.	Service Paletized shipment (main service 24) cannot have additional service Pallet (AdditionalService 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 client_reference_number.	Shipment with given client_reference_number already exists in 3PDX.
Vrsta isporuke je na paketomatu, ali navedena poslovnica isporuke nije paketomat.	Parcel locker delivery is given (delivery_type 3), but specified delivery center (recipient_delivery_center) is not a parcel locker.
Vrsta isporuke je na paketomatu, ali paketomat je neaktivan.	Parcel locker delivery is given (delivery_type 3), but specified parcel locker (recipient_delivery_center) is not active.
Vrsta isporuke je u poštanskom uredu, ali navedena poslovnica isporuke nije dostavni/isporučni poštanski ured.	Post office delivery is given (delivery_type 2), but specified delivery center (recipient_delivery_center) is not a delivery post office.
Vrsta prikupa je na paketomatu, ali navedena poslovnica prikupa nije paketomat.	Parcel locker pickup is given (pickup_type 3), but specified pickup center (sender_pickup_center) is not a parcel locker.
Vrsta prikupa je na paketomatu, ali paketomat je neaktivan.	Parcel locker pickup is given (pickup_type 3), but specified parcel locker (sender_pickup_center) is not active.
Za service 39 polje reference_field_B ne smije biti prazno.	For Easy Return service (main service 39) link to an outbound shipment is mandatory (in the reference_field_B 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).
Za tu pošiljku nije dozvoljena dopunska usluga Povrat prazne euro-palete.	Unallowed AdditionalService (54) for the specified MainService.

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	resent 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 bought goods to the sender (e.g. a Webshop), can be realized in the DxWeb service through the additional service 45 or services 39 and 40. It can also be realized through standard shipment.

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> (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
 - follow the instructions at the parcel locker

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>. 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> (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> (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> (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
 - follow the instructions at the parcel locker

Return through standard shipment

Returns can also be realized through standard shipments (not Easy Return service). In this case the user subsequently (at the customer's request) submits return shipment to the web service and sends to the customer the address label (or just shipment tracking number). In this case the user must distinguish the pickup channel (address, post office or parcel locker).

Return shipment must meet the following conditions:

- `payed_by` must be 2
- `service` can be 26, 29, 32 or 38
- `pickup_type` must be:
 - 1 for home collection
 - 2 for post office drop-off
 - `sender_pickup_center` can be any zip code
 - you can always put 10000
 - 3 for parcel locker drop-off
 - `sender_pickup_center` can be any locker code
 - you can always put 61001
- parcel locker drop-off
 - provide compartment size (`parcel_size`: X, S, M or L)

- if it is unknown, use parcel_size S.

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

- home collection
 - the customer does not have to do anything additional
- post office drop-off
 - drop-off the return shipment at any post office with the attached return address label or just shipment tracking number
- parcel locker drop-off
 - drop-off the return shipment at any parcel locker with the attached return address label or just shipment tracking number
 - follow the instructions at the parcel locker.

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

1.1 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));
```


2.1 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.eyJDRUNPREUiOiIxMTEzMTEiLCJuYmYiOiJlE3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiWF0IjoxNzAyMzc0MTc2fQ.jqmhC7zruvhWzItP-DW2inIJR3wvqkR0anW1IHjIz7A");
var body = @"{
" + "\n" +
@"    ""parcels"":[
" + "\n" +
@"        {
" + "\n" +
@"            ""client_reference_number"":""Test 12367"",
" + "\n" +
@"            ""service"":""38"",
" + "\n" +
@"            ""paid_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);
```



```
        "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_order',
    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.eyJDRUNPREUiOiIzMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiYWV0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anWlIHjIz7A'
),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;

```

2.2 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.eyJDRUNPREUiOiIxMTEzMTEiLCJuYmYiOiJlMTEzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiWF0IjoxNzAyMzc0MTE2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");
var body = @"{
" + "\n" +
@"    ""parcels"": [
" + "\n" +
@"        {
" + "\n" +
@"            ""client_reference_number"": ""Test 12366""
" + "\n" +
@"        },
" + "\n" +
@"        {
" + "\n" +
@"            ""client_reference_number"": ""Test 12365""
" + "\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.eyJkdRUNPREUiOiIxMTEzMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiawWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zrurvhwzItP-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_orde
rs',
    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.eyJDRUNPREUiOiIzMTE
xMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiYWV0IjoxNzAyMzc0MTc2fQ.jqmhc7z
ruvhwzItP-DW2inIUR3wvqkR0anWlIHjIz7A'
    ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

2.3 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.eyJDRUNPREUuOiIxmTEzMTEiLCJuYmYiOiJlMDEzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiWF0IjoxNzAyMzc0MTE2fQ.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.eyJDRUNPREUiOiIxmTEzMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiWF0IjoxNzAyMzc0MTc2fQ.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));
```


2.4 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.eyJDRUNPREUuOiIxMTEzMTEiLCJuYmYiOiJlMTEzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiWF0IjoxNzAyMzc0MTE2fQ.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.eyJDRUNPREUiOiIxMTEuMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiWF0IjoxNzAyMzc0MTc2fQ.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.eyJDRUNPREUoiOiIxMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiYWV0IjoxNzAyMzc0MTc2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A'
    ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```


3.1 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.eyJDRUNPREUiOiIxMTEzMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiYWV0IjoxNzAyMzc0MTE2fQ.jqmhc7zruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A");
var body = @"{
" + "\n" +
@"    ""facilityType"": ""PAK"",
" + "\n" +
@"    ""nextWeek"": 0,
" + "\n" +
@"    ""searchText"": """"
" + "\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.eyJDRUNPREUyOiIzMTEzMTEiLCJuYmYiOiJlMzMDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiWF0IjoxNzAyMzc0MTc2fQ.jqmhc7zrurvhwzItP-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_d
elivery_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.eyJDRUNPREUiOiIxmTE
xMTEiLCJuYmYiOiJlZ3MDIzNzQxNzYsImV4cCI6MTcwMjM4ODU3NiwiawWF0IjoxNzAyMzc0MTc2fQ.jqmhc7z
ruvhwzItP-DW2inIJR3wvqkR0anW1IHjIz7A'
    ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```