

EKIS remote API documentation

The remote API responds to POST requests with *Content-Type: application/x-www-form-urlencoded* formatted data. Each request has to contain their assigned unique client “secret”.

Creating a shipment

The following is a table containing all possible fields for parcel creation:

Contents	Field name	Required?	Max length	Valid values
[base key-value] Secret key	secret	Yes	32	Your secret key
[array key]	parcels			
[inner array key] Some kind of internal ID		Yes	32	Unique parcel ID from your system Don't use simbols like "[" and "]" if not escaped
Shipment type	type	Yes	3	“le” (Latvia); “Be” (Lithuania and Estonia); “Ems” (all other countries)
Multiple parcels ^[19]	multiparcel		1	0; 1
Multiple parcels group ID	group_uid	Yes ^[10]	36	Unique ID for every multi-parcel group
Weight of the shipment (Kg)	package_weight		7	
Insurance	insured	Yes ^[12]	1	0; 1
Insurance sum	insurance_sum	Yes ^[11]	7	
Cash on delivery	post_payment		1	0; 1
COD amount	post_payment_sum	Yes ^[12]	7	
COD transfer account number	post_payment_account		21	
COD transfer detail text	post_payment_aim	Yes ^[12]	255	
COD payer	post_payment_payer	Yes ^[12]	100	

_COD company registration number	post_payment_reg_number	Yes ^[12]	16	
COD bank details	post_payment_details	Yes ^[12]	255	
First and Last name	name_surname	Yes ^[26]	35	
Recipient company name	company_name	Yes ^[26]	35	
Client ID	client_nr		30	
Country	country_id	Yes	5	WS ^[20]
City	city	Yes ^[27]	32	
Area	area	Yes ^[27]	32	
District	district		100	
Village	village		100	
Street	street	Yes ^[7]	80	
House nr	house	Yes ^[7]	32	
Apartment nr	apartment_nr		20	
Postal code	zipcode	Yes	7	
USA state code	zip_state	Yes ^[6]	2	WS ^[21]
Phone	phone	Yes	20	Latvia: not less than 8 digits; Lithuania: "3706" or "86" and 7 digits
E-mail address	email	Yes ^[8]	255	Email format
Comments	comments		255	
Preferred date	preferred_delivery_date		10	YYYY-MM-DD
Preferred time	preferred_delivery_time		11	hh:mm-hh:mm
Vendor ID	vendor_id		255	
Importer ID	importer_id		255	

Contents	package_contents	Yes ^[4]	20	“Prece” (Other); “Dāvana” (Gift); “Dokuments” (Document); “Paraugi” (Samples)
Currency	content_currency	Yes ^[5]	3	EUR; USD
Goods sent	[array key] ParcelContent	Yes ^[4]		
Name of goods sent	title	Yes	255	
Amount	amount	Yes	11	
Net weight, kg	weight	Yes	7	
Value	value	Yes	8	
Harmonized system code	hs_code	Yes ^[28]	22	
Origin country	origin_country_id	Yes ^[28]	5	WS ^[20]
Width in meters	width		7	XXXX.XXX
Height in meters	height		7	XXXX.XXX
Length in meters	length		7	XXXX.XXX
Delivery priority	delivery_priority	Yes ^[9]	2	X1; X2
Documents to return	documents_return		1	0; 1
Return of documents	[array key] ParcelDocument	Yes ^[13]		
Document number	document_nr		50	
Document type	document_type_id		10	"1" (PPR); "3" (Leasing document); "4" (Other)
Extra stacker for carry assist	carry_assist		1	0; 1
Fragile	fragile		1	0; 1
Oversize shipment	large_size		1	0; 1
Circle K ID	statoil_id		9	WS ^[22]
Latvijas Pasts locker ID	station_id		9	WS ^[23]
Lithuania terminal ID	terminal_id		11	WS ^[24]
Lithuania terminal package size	size	Yes ^[16]	50	“Small”; “Medium”; “Large”

Pickup at the post office	pickup		1	0; 1
Pickup postal code	pickup_zipcode	Yes ^[15]	7	WS ^[25]
SMS/E-mail invitation	sms_invite	Yes ^[14]	1	0; 1
What action to take in case delivery is impossible	irreversible_action		2	“AB” (Refusal of acceptance); “RD” (Return to sender after 30 days); “TF” (Immediately return to sender)
To incinerate on parcel return	return_incinerate		1	0; 1 (only non-EU)

- [1] Required, if type is set to "le" (to Latvia)
- [2] Required, if type is set to "Be" (to Lithuania or Estonia)
- [3] Required, if type is set to "Ems"
- [4] Required, if type is set to "Be" or "Ems" (not to Latvia)
- [5] Required, if type is set to "Ems" or when sending to Estonia
- [6] Required when sending to United States
- [7] Required when sending to United States or Russia
- [8] Required when sending to Lithuania
- [9] Required when sending to Latvia by courier
- [10] Required, if multiparcel set to 1
- [11] Required, if insured is set to 1
- [12] Required, if post_payment is set to 1
- [13] Required, if documents_return is set to 1
- [14] Required, if statoil_id is not empty
- [15] Required, if pickup is set to 1
- [16] Required, if terminal_id is not empty
- [17] -
- [18] -
- [19] Multiple parcels must be created with the same recipient data. Only one of the group should contain cash on delivery (fields that start with "post_payment"), insurance ("insured", "insurance_sum") and documents return ("documents_return", "document_type") data
- [20] Use "id" value from <https://express.pasts.lv/countriesApi/index> web service for countries list (JSON format)
- [21] Use key value from <https://express.pasts.lv/usaStateApi/index/lv> or <https://express.pasts.lv/usaStateApi/index/en> when "country_id" is set to 12^[20]
- [22] Use "id" value from <https://express.pasts.lv/dusApi/index> web service for Circle K gas stations (JSON format)
- [23] Use "id" value from <https://express.pasts.lv/stationApi/index> web service for Latvijas Pasts lockers (JSON format)
- [24] Use "id" value from <https://express.pasts.lv/terminalApi/index> web service for Lithuania post terminals list (JSON format)
- [25] Use "postcode" value from <https://express.pasts.lv/postOfficeApi> web service for Latvia post offices and "code" value from <https://express.pasts.lv/postOfficeApi/lt> for Lithuania post offices (JSON format)
- [26] At least one of fields "name_surname"/"company_name" should be filled
- [27] At least one of fields "city"/"area" should be filled
- [28] Required when sending outside of EU (except when "package_contents"="Dokuments")

PHP v5.6 example

```
<?php
function send($url, $data = null, $checkCert = true) {
    $curl = curl_init();

    curl_setopt($curl, CURLOPT_HEADER_OUT, true);
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_IPRESOLVE, CURL_IPRESOLVE_V4);
    //curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, $checkCert);
    //curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, $checkCert);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($curl, CURLOPT_POSTREDIR, true);
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_POST, true);

    if (is_string($data)) {
        $data = json_decode($data, true);
    }
    if (empty($data) || !is_array($data)) {
        return false;
    }

    // Content-Type: application/x-www-form-urlencoded
    $data = http_build_query($data);

    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);

    $response = curl_exec($curl);
    $errCode = curl_errno($curl);
    $errText = curl_error($curl);
    curl_close($curl);
    if (!empty($errCode)) {
        $response = array(
            'code' => $errCode,
            'text' => $errText
        );
    }
    return $response;
}

$url = 'https://express.pasts.lv/...';

$data = '{...}';

$response = send($url, $data);
if (!is_array($response)) {
    // JSON data
    $data = json_decode($response, true)

    // PDF data
    $f = fopen('output.pdf', 'w');
    fwrite($f, $response);
    fclose($f);
} else {
    die("ERROR ({$response['code']}) : {$response['text']}");
}
```

Python v3.8.1 example

```
import ssl, json
from urllib.parse import urlencode
from urllib.request import Request, urlopen

def http_build_query(data):
    parents = list()
    pairs = dict()

    def renderKey(parents):
        depth, outStr = 0, ''
        for x in parents:
            s = "[%s]" if depth > 0 or isinstance(x, int) else "%s"
            outStr += s % str(x)
            depth += 1
        return outStr

    def r_urlencode(data):
        if isinstance(data, list) or isinstance(data, tuple):
            for i in range(len(data)):
                parents.append(i)
                r_urlencode(data[i])
                parents.pop()
        elif isinstance(data, dict):
            for key, value in data.items():
                parents.append(key)
                r_urlencode(value)
                parents.pop()
        else:
            pairs[renderKey(parents)] = str(data)

    return pairs
    return urlencode(r_urlencode(data))

url = 'https://express.pasts.lv/...'
data = {...}

# Content-Type: application/x-www-form-urlencoded
request = Request(url, http_build_query(data).encode('utf-8'))

# Content-Type: application/json
json_data = json.dumps(data).encode('utf8')
request = Request(url, data=json_data, headers={'Content-Type': 'application/json'})

ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE

resp_data = urlopen(request, context=ctx).read()
# JSON data
resp = resp_data.decode()
print(resp)
# PDF data
with open('output.pdf', 'wb') as f:
    f.write(resp_data)
```

Shell "curl" command examples

Parcel create

```
curl -d "secret=SECRET-KEY&parcels[Internal-ID-1][type]=Ie&parcels[0][name_surname]=Aina Priede&parcels[0][city]=Riga&parcels[0][street]=Ziemeļu iela&parcels[0][house]=2&parcels[0][zipcode]=LV-1000&parcels[0][phone]=22222222&parcels[0][country_id]=7&parcels[0][pickup]=1&parcels[0][pickup_zipcode]=LV-1044&parcels[0][sms_invite]=1" -H "Content-Type: application/x-www-form-urlencoded" -L -X POST https://express.pasts.lv/parcelsApi/create
```

PDF label

```
curl -d "secret=SECRET-KEY&size=150x100&parcels[0]=CE100005322LV&parcels[1]=CE100005324LV&parcels[2]=CE100005326LV" -H "Content-Type: application/x-www-form-urlencoded" -L -X POST -o sticker.pdf https://express.pasts.lv/parcelsDocumentsApi/stickers
```

PDF stricker

```
curl -d "secret=SECRET-KEY&parcels[0]=CE100005322LV&parcels[1]=CE100005324LV&parcels[2]=CE100005326LV" -H "Content-Type: application/x-www-form-urlencoded" -L -X POST -o sticker.pdf https://express.pasts.lv/parcelsDocumentsApi/labels
```


Parcel creation requests

URL: <https://express.pasts.lv/parcelsApi/create>

Grouped to post office with post-payment and returnable documents

```
{
  "secret": "SECRET-KEY",
  "parcels": [
    {
      "type": "Ie",
      "multiparcel": 1,
      "group_uid": "MY-GROUP-1",
      "name_surname": "Aina Priede",
      "city": "Rīga",
      "street": "Ziemeļu iela",
      "house": "2",
      "zipcode": "LV-1000",
      "phone": "22222222",
      "country_id": 7,
      "pickup": 1,
      "pickup_zipcode": "LV-1044",
      "sms_invite": 1,

      "insured": 1,
      "insurance_sum": 23.89,
      "post_payment": 1,
      "post_payment_sum": 23.89,
      "post_payment_aim": "Pēcmaksa",
      "ParcelDocument": [
        {
          "document_type_id": 3,
          "document_nr": "38-5-37-36"
        }, {
          "document_type_id": 3,
          "document_nr": "38-5-37-36"
        }
      ]
    }, {
      "type": "Ie",
      "multiparcel": 1,
      "group_uid": "MY-GROUP-1",
      "name_surname": "Aina Priede",
      "city": "Rīga",
      "street": "Ziemeļu iela",
      "house": "2",
      "zipcode": "LV-1000",
      "phone": "22222222",
      "country_id": 7,
      "pickup": 1,
      "pickup_zipcode": "LV-1044",
      "sms_invite": 1
    }
  ]
}
```

```
# To post office
{
  "secret": "SECRET-KEY",
  "parcels": [{
    "type": "Ie",
    "name_surname": "Aina Priede",
    "city": "Rīga",
    "street": "Ziemeļu iela",
    "house": "2",
    "zipcode": "LV-1000",
    "phone": "22222222",
    "country_id": 7,
    "pickup": 1,
    "pickup_zipcode": "LV-1044",
    "sms_invite": 1
  }]
}

# Latvijas Pasts locker
{
  "secret": "SECRET-KEY",
  "parcels": {
    "Internal-ID-1": {
      "type": "Ie",
      "name_surname": "Rūdis Ābele",
      "phone": "22222222",
      "email": "abc@pasts.lv",
      "country_id": 7,
      "station_id": 99
    }
  }
}

# Circle K DUS
{
  "secret": "SECRET-KEY",
  "parcels": [{
    "type": "Ie",
    "name_surname": "Rūdis Ābele",
    "phone": "22222222",
    "email": "abc@pasts.lv",
    "country_id": 7,
    "statoil_id": 2
  }]
}
```

```
# Lithuania post office
{
  "secret": "SECRET-KEY",
  "parcels": {
    "Internal-ID-1": {
      "type": "Be",
      "name_surname": "Rūdis Ābele",
      "phone": "860000000",
      "email": "abc@pasts.lv",
      "country_id": 2,
      "city": "Kaunas",
      "street": "Veiverių g.",
      "house": "150B",
      "zipcode": "46391",
      "pickup": 1,
      "pickup_zipcode": "46391"
    }
  }
}
```

```
# Lithuania post terminal
{
  "secret": "SECRET-KEY",
  "parcels": [{
    "type": "Be",
    "name_surname": "Rūdis Ābele",
    "phone": "860000000",
    "email": "abc@pasts.lv",
    "country_id": 2,
    "city": "Kaunas",
    "zipcode": "46391",
    "size": "Small",
    "terminal_id": 16
  }]
}
```

```

# Baltic express and other countries
{
  "secret": "SECRET-KEY",
  "parcels": {
    "My-234-ID": {
      "type": "Be",
      "name_surname": "Some Person",
      "city": "Some city",
      "zipcode": "12345",
      "phone": "860000000",
      "country_id": 2,
      "ParcelContent": [{
        "name": "Leather Boots",
        "amount": 2,
        "value": 114.87,
        "weight": 6.348
      }]
    },
    "ID": {
      "type": "Ems",
      "company_name": "Some company",
      "name_surname": "Some Person",
      "country_id": 12,
      "city": "Some city",
      "street": "Some street",
      "house": "Some house",
      "zipcode": "12345",
      "zip_state": "AZ",
      "email": "test@example.com",
      "phone": "37060000000",
      "comments": "Order #000-TEST",
      "package_contents": "Prece",
      "content_currency": "EUR",
      "ParcelContent": [{
        "title": "Leather Boots",
        "amount": 2,
        "value": 114.87,
        "weight": 6.348
      }], {
        "title": "Plastic",
        "amount": 19,
        "value": 15.00,
        "weight": 0.170
      }]
    }
  }
}

```

The response will contain data in JSON format.

A successful request would be a response containing shipment numbers or error messages in the case of missing or invalid fields in the request, for example:

```

{
  "Internal-ID-1": {
    "phone": "Tālruņa numurs nav norādīts"
  },
  "Other-2": "CE100005324LV",
  "Another-ID": "CE100005326LV",
  "467-183-5.43": "CE100005328LV"
}

```

An malformed request would produce an error, for example:

```
{
  "error": "Unknown user"
}
```

Parcel deletion request example:

URL: <https://express.pasts.lv/parcelsApi/delete>

```
{
  "secret": "SECRET-KEY",
  "parcels": [
    "CE100005322LV",
    "CE100005324LV",
    "CE100005326LV"
  ]
}
```

The response will contain data in JSON format and will look like this:

```
{
  "result": "ok"
}
```

Label file generation request example:

URL: <https://express.pasts.lv/parcelsDocumentsApi/labels>

```
{
  "secret": "SECRET-KEY",
  "parcels": [
    "CE100005322LV",
    "CE100005324LV",
    "CE100005326LV",
    "CE100005328LV"
  ]
}
```

The response will contain binary data (PDF file).

PNS 40 Label file generation request example:

URL: <https://express.pasts.lv/api/apiForm/pns040> & Content-Type: application/json

```
{
  "auth": "SECRET-KEY",
  "strid": "CE100005322LV"
}
```

The response will contain binary data (PDF file).

EMS/EPG Label file generation request (for "type"="Ems" parcels) example:

URL: <https://express.pasts.lv/api/apiForm/ems> & Content-Type: application/json

```
{
  "auth": "SECRET-KEY",
  "strid": ["EA100005322LV", "CE100005324LV"]
}
```

The response will contain binary data (PDF file).

Stickers (PDF) file generation request example:

URL: <https://express.pasts.lv/parcelsDocumentsApi/stickers>

```
{
  "secret": "SECRET-KEY",
  "size": "150x100",
  "parcels": [
    "CE100005322LV",
    "CE100005324LV",
    "CE100005326LV",
    "CE100005328LV"
  ]
}
```

Possible "size" values:

- "150x100"; (preferred!)
- ~~"130x75";~~
- ~~"105x37";~~
- ~~"70x42";~~
- ~~"70x37";~~
- ~~"64x32";~~
- ~~"157x98";~~ (only for Be parcel type)

* crossed "size" values from 01.01.2021 will be disabled.

The response will contain binary data (PDF file).

Stickers (EPL) file generation request example (only to Latvia and Lithuania):

URL: <https://express.pasts.lv/parcelsDocumentsApi/epl>

```
{
  "secret": "SECRET-KEY",
  "size": "100x150",
  "parcels": [
    "CE100005322LV",
    "CE100005324LV",
    "CE100005326LV",
    "CE100005328LV"
  ]
}
```

Possible "size" values: "100x150"

The response will contain binary data (EPL text).

Manifest file generation request example:

URL: <https://express.pasts.lv/parcelsDocumentsApi/manifest>

```
{
  "secret": "SECRET-KEY",
  "parcels": [
    "CE100005322LV",
    "CE100005324LV",
    "CE100005326LV",
    "CE100005328LV"
  ]
}
```

The response will contain binary data (PDF file).

Courier request:

URL: <https://express.pasts.lv/courierApplicationsApi/create>

```
{
  "secret": "SECRET-KEY",
  "applications": {
    "My-ID-792376": {
      # YYYY-MM-DD
      "when": "2019-11-29",
      # XX:XX (09:00-17:59, where time_to >= time_from)
      "time_from": "13:08",
      # XX:XX (09:00-17:59, where time_from <= time_to)
      "time_to": "14:58",
      # LV format phone
      "phone": "22222222",
      # Optional
      "email": "abc@pasts.lv",
      # Max 255 char
      "address": "Ziemeļu iela 10, Lidosta",
      # Integer (can be 0)
      "pallet_count": "0",
      # Integer (can be 0)
      "parcel count": "0",
      # "Vieglais kravas buss"; "Smagā kravas automašīna"
      "transport": "Vieglais kravas buss",
      # Optional (max 500 char)
      "comments": "Komentāri"
    }
  }
}
```

A failed request would be a response containing error messages in the case of missing or invalid fields in the request, for example:

```
{
  "My-ID-792376": {
    "address": "Lūdzu, norādiet adresi"
  }
}
```

The response will contain data in JSON format and will look like this:

```
{
  "My-ID-792376": True
}
```

Courier request v2:

URL: <https://express.pasts.lv/courierApplicationsApi/create>

```
{
  "secret": "SECRET-KEY",
  "applications": {
    "My-ID-3954653": {
      # [required] YYYY-MM-DD
      "when": "2022-02-01",
      # [required] hh:mm (min 9:00 or current time + 1h
      # & 1h interval till "time_to")
      "time_from": "15:26",
      # [required] hh:mm (max 18:00 & 1h interval from "time_from")
      "time_to": "16:26",
      # [required] 1 - bus; 2 - truck
      "transport_type": 1,
      # [optional] Parcel numbers separated with comma "," or semicolon ";"
      "parcel_strids": "CE123456789LV",
      # [optional] Single Krl manifest number
      "manifest": "12345",
      # [optional] cubic meters in m³
      "cbm": 1.000,
      # [optional] loading meters
      "ldm": 0.417,
      # [optional] weight in Kg
      "weight": 22.101,
      # [required] company name or name and surname
      "company_name": "VAS \"Latvijas Pasts\"",
      # [required] phone no.
      "phone": "37122222222",
      # [optional] e-mail
      "email": "info@pasts.lv",
      # [optional] e-mail
      "comments": "My comments",
      # [required] city or area
      "city": null,
      # [required] area or city
      "area": "Mārupes novads",
      # [optional] district
      "district": "Mārupes pagasts",
      # [optional] village
      "village": "Lidosta \"Rīga\"",
      # [optional] street
      "street": "Ziemeļu iela",
      # [optional] house
      "house": "10",
      # [optional] apartment no.
      "apartment_nr": null,
      # [required] postcode
      "zipcode": "LV-1053"
    }
  }
}
```

A failed request would be a response containing error messages in the case of missing or invalid fields in the request, for example:

```
{
  "My-ID-3954653": {
    "zipcode": "Lūdzu, norādiet indeksu"
  }
}
```

The response will contain data in JSON format and will look like this:


```
{
  "My-ID-3954653": True
}
```

Parcel tracking

URL: <https://express.pasts.lv/parcelsApi/track>

You can send a request in following format to get parcel tracking data:

```
{
  "secret": "SECRET-KEY",
  "id": "CE100005322LV",
  "fields": [
    "event",
    "date",
    "location",
    "next_office"
  ]
}
```

Without key “fields”, default values in response will be (“event”, “date”).

Available values for “fields” key: “event”, “date”, “location”, “next_office”.

The response will contain event code* and date data in JSON format by default and will look like this:

```
[{
  "event": "TN001",
  "date": "2013-05-07 07:45:05"
},
{
  "event": "TN032",
  "date": "2013-05-07 07:45:05"
}]
```

With “fields” key set:

```
[{
  "event": "TN001",
  "date": "2013-05-07 07:45:05",
  "location": "LVRED",
  "next_office": ""
},
{
  "event": "TN032",
  "date": "2013-05-07 07:45:05",
  "location": "LVRED",
  "next_office": ""
}]
```

Table with all event codes and their corresponding names:

Event code	Event name
TNLO1	EPG parcel - information received
TNLV1	Receive item from local customer (XML)
TN001	Receive item from customer
TN002	Send item to domestic location
TN031	Send item to customs
TN032	Receive item at delivery office
TN033	Receive item at location
TN035	Send item to domestic location

TN036	Unsuccessful item delivery attempt
TN037	Deliver item
TN041	Record recipient information (Inb)
TN050	Print advice of arrival
TN074	Send item out for physical delivery (Inb)
TN128	Item returned to sender

Using delivery locations web services

There are 5 delivery location web services available:

- Latvia post offices - <https://express.pasts.lv/postOfficeApi>
- Circle K gas stations - <https://express.pasts.lv/dusApi/index>
- Latvijas Pasts lockers - <https://express.pasts.lv/stationApi/index>
- Lithuania post offices - <https://express.pasts.lv/postOfficeApi/lt>
- Lithuania post terminals - <https://express.pasts.lv/terminalApi/index>

Please note, that above listed services don't use same field names and no unique ID codes are shared between them.

Latvia post offices WS fields:

id – post office identifier
 postcode – post office code (unique post office identifier)
 label – the name of post office
 working_monday – monday working hours
 working_tuesday – tuesday working hours
 working_wednesday – wednesday working hours
 working_thursday – thursday working hours
 working_friday – friday working hours
 working_saturday – saturday working hours
 working_sunday – sunday working hours
 region – region (address unit)
 city – city (address unit)
 area – area (address unit)
 district – district (address unit)
 village – village (address unit)
 street – street (address unit)
 house – house (address unit)
 zipcode – postal code (address unit)
 address – full address (sum of all address units)
 lat – latitude (GPS coordinate)
 long – longitude (GPS coordinate)

Circle K gas stations WS fields:

id – Circle K gas station identifier
 title – the name of Circle K gas station
 code – Circle K gas station code (unique Circle K gas station identifier)

working_monday – monday working hours
working_tuesday – tuesday working hours
working_wednesday – wednesday working hours
working_thursday – thursday working hours
working_friday – friday working hours
working_saturday – saturday working hours
working_sunday – sunday working hours
region – region (address unit)
city – city (address unit)
area – area (address unit)
district – district (address unit)
village – village (address unit)
street – street (address unit)
house – house (address unit)
zipcode – postal code (address unit)
address – full address (sum of all address units)
lat – latitude (GPS coordinate)
long – longitude (GPS coordinate)

Latvijas Pasts lockers WS fields:

id – Latvijas Pasts locker identifier
code – Latvijas Pasts locker code (unique Latvijas Pasts locker identifier)
max_size – locker max accepted size
working_monday – monday working hours
working_tuesday – tuesday working hours
working_wednesday – wednesday working hours
working_thursday – thursday working hours
working_friday – friday working hours
working_saturday – saturday working hours
working_sunday – sunday working hours
additional_information – additional information
business_place – business place name
location – specific locker location (inside/outside)
title – the name of Circle K gas station
city – city (address unit)
area – area (address unit)
district – district (address unit)
village – village (address unit)
street – street (address unit)
house – house (address unit)
zipcode – postal code (address unit)
address – full address (sum of all address units)
lat – latitude (GPS coordinate)
long – longitude (GPS coordinate)
sizes – array with size title and dimensions in cm

Lithuania post offices WS fields:

code – post office code (unique post office identifier)

label – the name of post office
address – full address (sum of all address units)
lat – latitude (GPS coordinate)
long – longitude (GPS coordinate)

Lithuania post terminals WS fields:

id – post office identifier
title – the name of post office
postal_code – postal code (address unit)
code – post terminal code (unique post terminal identifier)
address – full address (sum of all address units)