

# MPL API

## technical description



# Content index

<b>Content index</b> .....	2
<b>1. Document</b> .....	4
1.1 Acronyms.....	4
1.2 Terms used .....	4
1.3 Document History .....	4
<b>2. View of synthesis</b> .....	6
<b>3. Objective of the document</b> .....	7
<b>4. Introduction to MPL API</b> .....	8
4.1 Overview.....	8
4.2 Interface components .....	8
<b>5. Connecting to MPL API</b> .....	9
5.1 General Terms and Conditions.....	9
5.2 Devportal access.....	9
5.3 API Access.....	9
5.4 Sandbox (test) environment.....	10
5.5 Live environment.....	10
5.6 API versioning.....	10
<b>6. Integration recommendations</b> .....	11
6.1 Business.....	11
6.1.1 Address.....	11
6.1.2 Shipment handling.....	11
6.1.3 Switching from EFJ generation to MPL API .....	11
6.1.4 Prices, closing .....	11
6.2 Information Technology .....	12
6.2.1 Parcel life cycles.....	12
6.2.2 Query grouping.....	12
6.2.3 Data package size .....	12
6.2.4 Error management .....	12
<b>7. MPL API services</b> .....	13
7.1 Business Services .....	13
7.2 Description of the process.....	15
7.3 Parcel dispatch process .....	17
7.4 HTTP Header information.....	17
7.4.1 Description .....	17
7.4.2 Request message.....	17

7.4.3	Example message .....	18
7.5	OAuth2 token request.....	19
7.5.1	Request message.....	19
7.5.2	Response message .....	19
7.5.3	Example message .....	20
7.6	Business service calls.....	20
<b>8.</b>	<b>Non-functional properties</b> .....	<b>21</b>
8.1	Availability .....	21
8.1.1	Service period.....	21
8.1.2	Maintenance Period .....	21
8.1.3	Unavailability.....	21
8.2	Improved safety and security.....	21
8.3	Unsupported JSON tags.....	21
<b>9.</b>	<b>Frequently Asked Questions</b> .....	<b>22</b>
<b>10.</b>	<b>Annex - Data types</b> .....	<b>23</b>
<b>11.</b>	<b>Annex - Samples</b> .....	<b>24</b>
11.1	Example: submitting data for domestic parcels.....	24
11.1.1	Parcel home delivery with cash on delivery.....	24
11.1.2	Parcel delivery to a PostaPoint, with value insurance .....	25
11.1.3	Two-parcel consignment home delivery with cash on delivery and value insurance... 26	
11.1.4	Parcel home delivery with an exchange parcel.....	27
11.1.5	Parcel home delivery, with inverse (~return delivery).....	29
11.1.6	Parcel delivery to a parcel terminal, with eased handling, goods payment and value insurance .....	30
11.2	Example: submitting data for domestic international parcels.....	31
11.2.1	Submitting a non-EU parcel.....	31
11.2.2	Submitting an EU parcel .....	33
11.3	Closing the posting of a parcel .....	34

# 1. Document

## 1.1 Acronyms

Acronym	Description
Base64	Base64 is a content encoding scheme based on a 64-character alphabet, supporting the conversion of binary data or content including special characters into ASCII character strings.
HTTPS	Hypertext Transfer Protocol over SSL
REST	Representational State Transfer
URL	Uniform Resource Locator

**Table 1 - Acronyms**

## 1.2 Terms used

Acronym	Description
Parcel	A single mail item to be delivered to the addressee.
Shipment	A set of parcels to be sent to the same addressee. We distinguish between shipments to be handled together (all parcels can only be delivered together) and shipments not to be handled together (the parcels within them can be delivered separately).
List	A data file, containing mail items and customer data, that includes the data of parcels to be dispatched on the same day and at the same location. Without a completed list, Magyar Posta cannot process the parcel data.
Address label	The address label that is to be attached to the parcel contains the sender's and the addressee's data, parcel identifier, etc.
Tracking number	Parcel ID

**Table 2 - Descriptions**

## 1.3 Document History

Alternative	Datum	Note
1.0	01/10/2019	Document established
1.1	19/10/2020	Minor clarifications, detailing the transition from EFJ generation to API

2.0	12/02/2021	Remove document redundancy, insert Devportal documentation link
2.1	26/07/2021	Samples, new chapter: 7.3

**Table 3 - Document History**

## 2. View of synthesis

Magyar Posta Zrt. offers MPL API web services that allow its partners

- to create shipments;
- to request delivery notes;
- to query parcel address labels; and
- to close posting lists

in order to use our parcel delivery services.

The MPL API services are free of charge, although customers may incur expenses developing their own related components.

Magyar Posta Zrt. does not assume any liability for such expenses arising from development, implementation and testing activities.

The following link provides on-line documentation on MPL API business calls:

<https://devportal.posta.hu/api/7>

### 3. Objective of the document

This document provides guidelines for MPL API customers, and a detailed description for the integration of MPL API REST web services.

The document elaborates on the following:

- Web service interface specification
- Error codes returned by the API
- Non-functional description of the API, including the availability of the service, and security requirements

The document is mainly targeted at developers and system integrators.

MPL API REST operations and HTTP methods described in detail in this document are as follows:

- POST /oauth2/token
- GET /v2/mplapi/shipments
- POST /v2/mplapi/shipments
- POST /v2/mplapi/shipments/{trackingNumber}/item
- GET /v2/mplapi/shipments/{trackingNumber}
- DELETE /v2/mplapi/shipments/{trackingNumber}
- POST /v2/mplapi/shipments/close
- GET /v2/mplapi/shipments/label
- POST /v2/mplapi/addresses/cityToZipCode
- POST /v2/mplapi/addresses/zipCodeToCity
- POST /v2/mplapi/deliveryplace
- POST /v2/mplapi/reports

## 4. Introduction to MPL API

### 4.1 Overview

In the simplest case, the logical process is the following:

- **Creation of shipments** – Generation of any number of shipments and address labels
- **Closing of posting list** – At the end of the day or any time sooner, closing of the posting list and requesting of the **combined** delivery note, which includes data of all shipments to be dispatched currently in a single document (mandatory function).  
**It is important that these parcels should appear in a single posting list.**

### 4.2 Interface components

Figure 1 shows the interface between MPL API and the calling system.

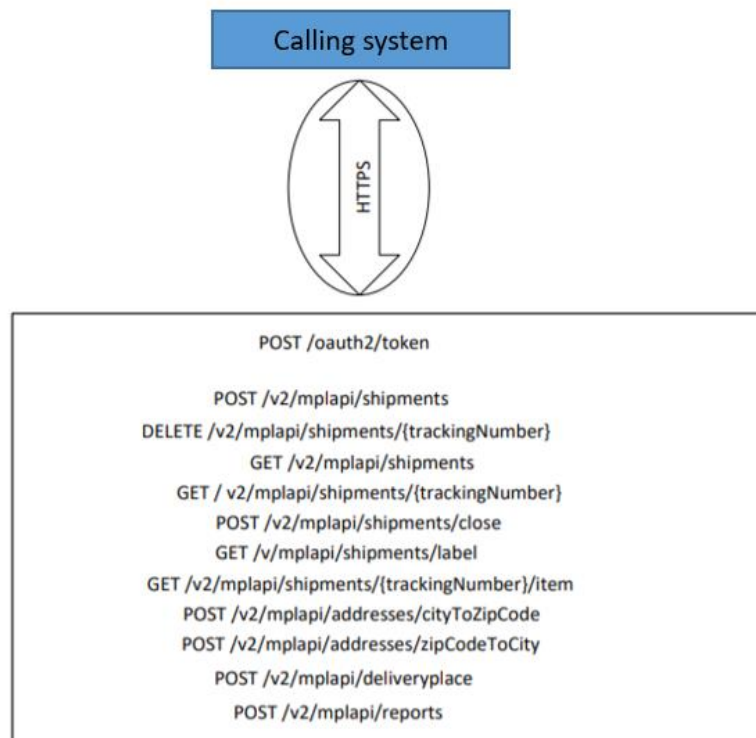


Figure 1 – MPL API V2



## 5. Connecting to MPL API

The following flowchart shows a high-level description of the steps of connecting to the MPL API

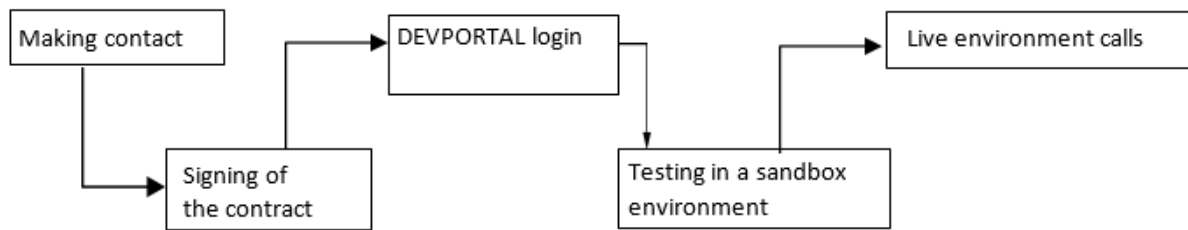


Figure 2 – Process of connecting to the MPL API

### 5.1 General Terms and Conditions

[https://www.posta.hu/general\\_terms\\_and\\_conditions](https://www.posta.hu/general_terms_and_conditions)

### 5.2 Devportal access

You can access the Devportal in the live environment at the following URL:

<https://devportal.posta.hu>

After logging in to the Devportal, you can view and copy the **API Key(client\_id)** and **API Secret (client\_secret)** values that are required for accessing the API Sandbox and the live environment. The two environments must use separate logins, which cannot be mixed between environments.

### 5.3 API Access

You can access the MPL API v2 in the live environment at the following URL:

<https://core.api.posta.hu/v2/mplapi>

You can access the MPL API v2 in the sandbox postal environment at the following URL:

<https://sandbox.api.posta.hu/v2/mplapi/>

Every business call should include the value of the Authorization key, otherwise you will receive an HTTP 401 (Unauthorized) error message.

#### 5.4 Sandbox (test) environment

The Sandbox (test) endpoint published on the Internet is backed by a so called Mock service. By setting parameters, the Mock services can be configured so that its operation may approximate the live environment background service as much as possible.

#### 5.5 Live environment

Behind the live endpoint published on the Internet is a live environment background service.

#### 5.6 API versioning

Magyar Posta Zrt. has been developing its services all the time, which entails that from time to time a new version of MPL API services is rolled out.

Magyar Posta Zrt. plans to maintain 3 versions of the MPL API:

- Current version
- Previous version
- Cancelled version

You always need to connect to the current version of the MPL API service as it is the version that provides the longest stable operating time without having to make changes to the calling system. If integration is already being developed in a calling system when a new version of the MPL API is released, the calling system can continue using the **previous version**.

We recommend that you should not use a cancelled version to integrate you calling system.

## 6. Integration recommendations

API calls generate data traffic, and MPL incurs expenses as a result, and for this reason MPL reserves the right to limit access to the API service in case of extreme overuse.

For optimal operation, we suggest that the following should be considered when integrating the MPL API.

### 6.1 Business

#### 6.1.1 Address

The MPL API returns a warning if the addressee's address could be incorrect, but accepts the data submitted despite the warning. We recommend that, if possible, you check the address and, if it is incorrect, erase the submitted data and resubmit it.

#### 6.1.2 Shipment handling

Shipment management is described in the GTC comprehensively, here we only briefly summarise the available options for clarity.

A parcel can be posted to an address in three ways:

- In a separate call individually: the sent parcels will be completely independent of each other, they will not be placed into a shipment
- Sent in one shipment, **not** to be handled together: the sent parcels will be in one shipment, which, however, may not be delivered by Magyar Posta at the same time
- Sent in one shipment, to be handled together: the sent parcels will be in one shipment, which will be delivered by the Magyar Posta at the same time

#### 6.1.3 Switching from EFJ generation to MPL API

There are two important differences between EFJ local generation and MPL API:

- The tracking number generating location (locally for EFJ, at the post office for MPL API)
- Data verification via MPL API is immediate and does not return a tracking number in case of an error, i.e. no incorrect parcel data can be sent

After submitting the parcel data, the address label can also be generated locally, in which case it is the responsibility of the caller to generate the address label in the appropriate format. If the address label in PDF format returned by the MPL API is suitable, it is recommended to use it.

#### 6.1.4 Prices, closing

When a close call is made, the MPL API will return the tracking numbers and their corresponding prices from the given list. The prices are based on the data submitted and are therefore only

indicative, the exact prices are determined during the actual dispatch. It is important to note that in the case of a closure, several lists may be created depending on the data submitted.

## **6.2 Information Technology**

MPL reserves the right to reject incorrectly addressed parcels in the future.

### **6.2.1 Parcel life cycles**

Parcels submitted through MPL API can be managed (erased, address label generated etc.) until they are closed. Once closed, the parcel data is no longer accessible via MPL API.

### **6.2.2 Query grouping**

If the caller's business process allows, we ask that calls that can be made with more than one piece of data (e.g. when sending shipments or requesting an address labels) are preferably made with more than one piece of data, i.e. one time with 100 pieces of data, not 100 times with one piece of data. An address label can be requested immediately when submitting the shipment. If this is in line with the caller's business process, it is advisable to do this in one step and not to submit the shipment and request the address label separately.

### **6.2.3 Data package size**

For some calls, we have introduced a limit on the amount of data that can be sent/requested at one time. The reason for this is to limit the size of the responses, since many calls return a PDF address label with base64 encoding.

### **6.2.4 Error management**

After submitting the data for each shipment, if the MPL API returns an error code, the incorrect data(s) must be corrected and resubmitted. Error codes can be used for repair if you want to automate the process to some extent. If all errors are left to the user's discretion, it is sufficient to display the error message.

## 7. MPL API services

This chapter describes services, message structures and fields that are provided by the MPL API v2 and appear in request and response messages.

These operations are available in the Swagger specification that you can view and download after logging in to the Devportal:

<https://devportal.posta.hu/api/7>

### 7.1 Business Services

The MPL API v2 provides our partners with functions to create, delete, and query shipments, to query address labels, and to close posting lists.

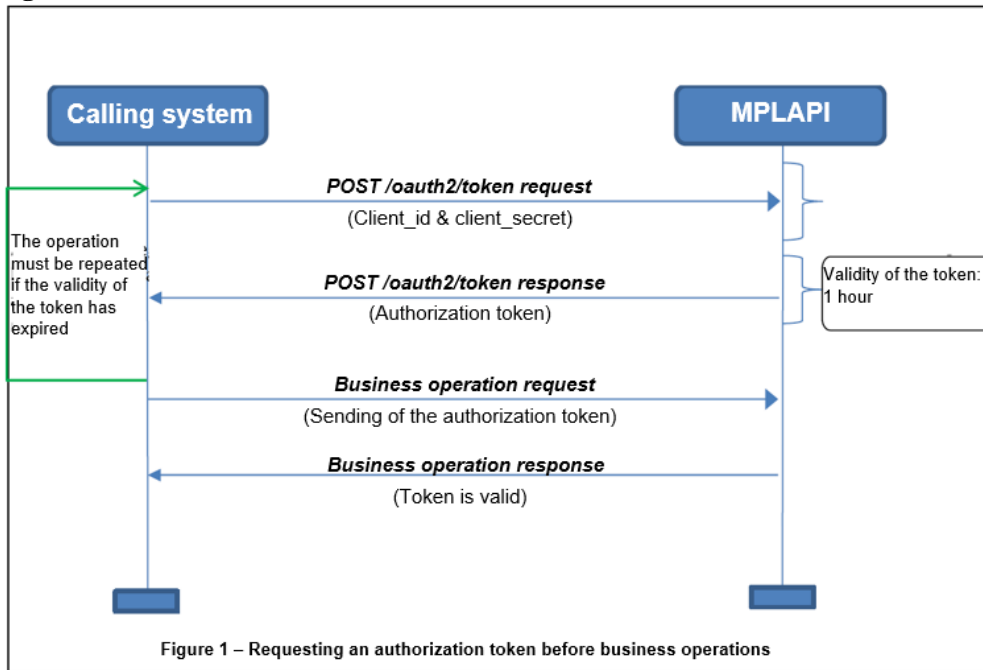
Feature	API function	Description	Technology	Connection type
<b>Improved safety and security</b>				
OAuth2 token request	POST /oauth2/token	For OAuth2 authorization, it creates an authorization token that should be inserted into the http header of business calls	JSON over HTTPS (REST)	Synchronized Request/Response
<b>Shipment</b>				
Shipment creation	POST /v2/mplapi/shipments	Shipment creation	JSON over HTTPS (REST)	Synchronized Request/Response
Shipment address label query	GET /v2/mplapi/shipments/label	Shipment address label query	JSON over HTTPS (REST)	Synchronized Request/Response
Shipment query	GET /v2/mplapi/shipments	Shipment query	JSON over HTTPS (REST)	Synchronized Request/Response

Shipment query by tracking number	GET /v2/ mplapi/shipments/ {trackingNumber}	Shipment query by tracking number	JSON over HTTPS (REST)	Synchronized Request/ Response
Item deletion by tracking number	DELETE /v2/ mplapi/shipments/ {trackingNumber}	Item deletion by tracking number	JSON over HTTPS (REST)	Synchronized Request/ Response
<b>Closing</b>				
Closing the posting list	POST /v2/ mplapi/shipments /close	Closing of a posting list.	JSON over HTTPS (REST)	Synchronized Request/ Response
Adding new parcels	POST /v2/ mplapi/shipments/ {trackingNumber}/item	Adding new parcels to an existing separately deliverable shipment.	JSON over HTTPS (REST)	Synchronised Request/ Response
Querying postcodes	POST /v2/ mplapi/addresses/ cityToZipCode	Querying postcodes	JSON over HTTPS (REST)	Synchronised Request/ Response
Querying settlements	POST /v2/ mplapi/addresses/ zipCodeToCity	Querying settlements	JSON over HTTPS (REST)	Synchronised Request/ Response
Querying alternative address locations	POST /v2/ mplapi/deliveryplace	Querying alternative address locations	JSON over HTTPS (REST)	Synchronised Request/ Response
Querying data	POST /v2/ mplapi/reports	Querying data	JSON over HTTPS (REST)	Synchronised Request/ Response

**Table 4 – Business services**

## 7.2 Description of the process

### Requesting an authorization token in the case of OAuth2 authorization



Normal daily operation with business requests:

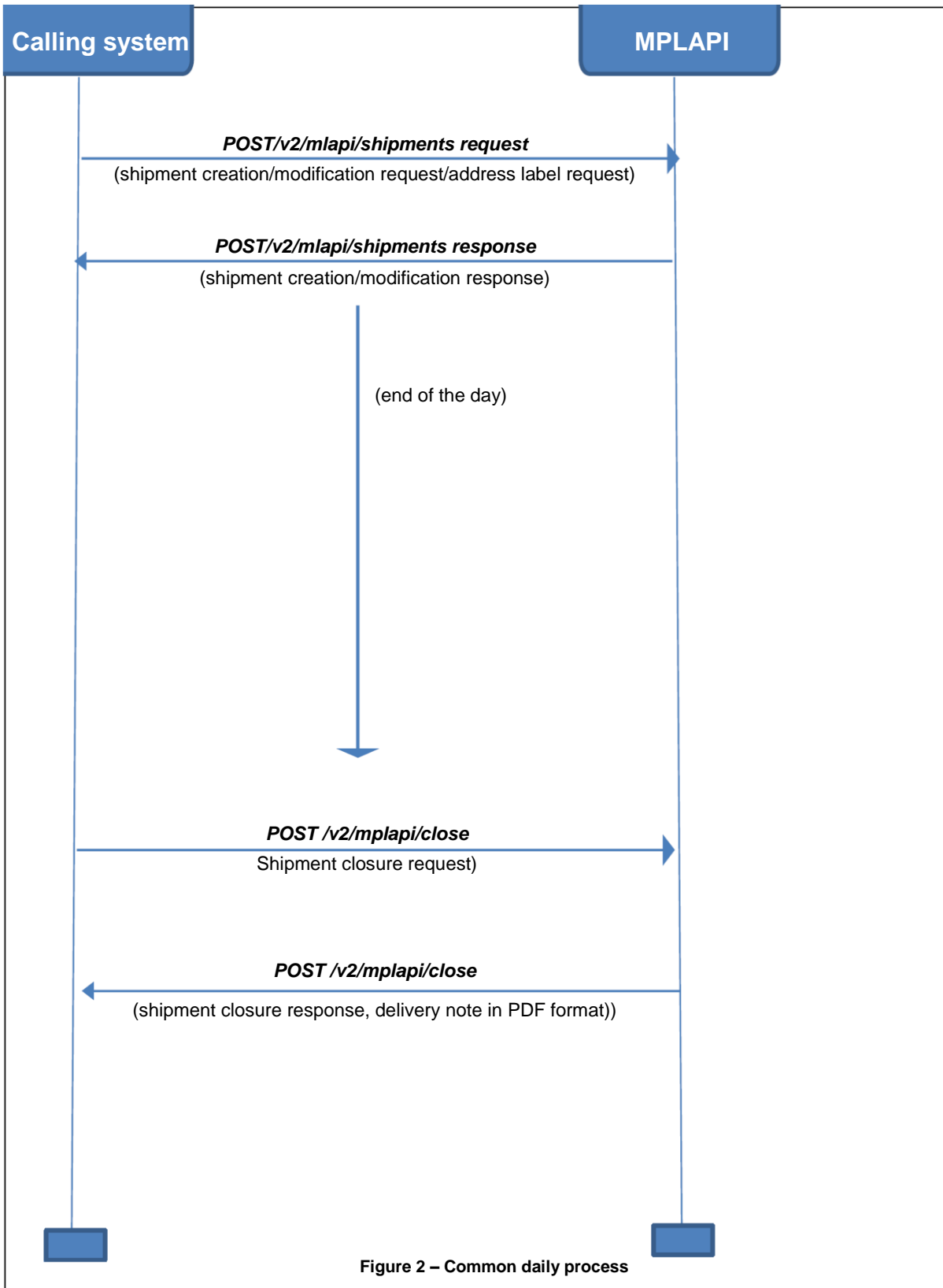
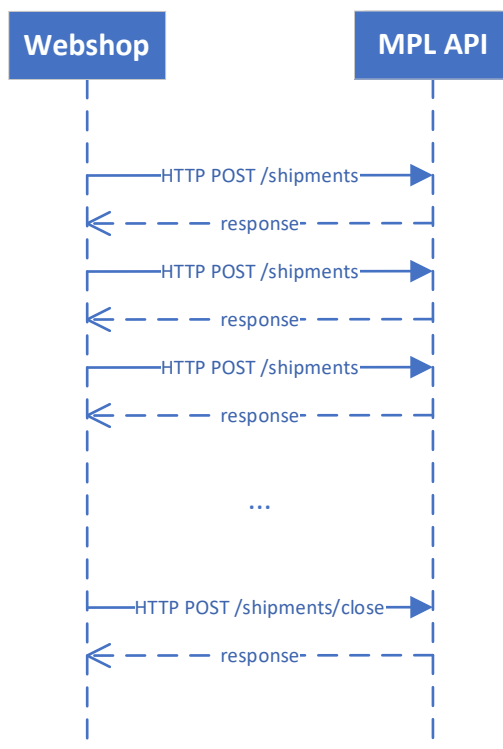


Figure 2 – Common daily process



### 7.3 Parcel dispatch process

Regardless of whether domestic and/or international parcels are being sent, a closing must be performed at the end of the parcel data submission. Without that no parcel may be dispatched. Parcel data can be sent one by one sequentially, multiple parcels at the same time in a single request, or in parallel for larger quantities. In the latter case, do not use more than 5 threads when submitting.



### 7.4 HTTP Header information

#### 7.4.1 Description

The HTTP header is meant to provide authentication, authorization, security and logging functionality in the MPL API system.

#### 7.4.2 Request message

Each business call must be authorized using Basic or OAuth2 authorization. The necessary client\_id and client\_secret values are provided by the MPL API. Field values are available after logging in to the MPL API Developer portal.

Parameter	Optional	Description
Authorization	No	HTTP key value for authorization. Its type and value vary depending on authorization method (OAuth2 or Basic)

X-Accounting-Code	No	Code (customer code) assigned to the customer by Magyar Posta Zrt.  The MPL API inserts into the response message the X-Accounting-Code value submitted by the caller.
X-Request-ID	No	<b>Unique message identifier. For each MPL API call, this field should have a unique GUID value.</b>  e.g. 827f3343-2cfd-4e46-a646-065a0a7268c4  The MPL API checks the GUID type format. If the format is incorrect, the message will be rejected.  The MPL API will insert into the response message the X-Request-ID value sent by the caller so the request and the response message can be correlated.
X-Correlation-ID	Yes	GUID type correlated message identifier. Messages that a logically correlated should have the same identifier.  e.g. F1A67AEF-70CE-42EB-B692-9C4CA81E1266  The MPL API checks the GUID type format. If it is incorrect, the message will be rejected.  The MPL API inserts into the response message the X-Correlation-ID value submitted by the caller. If the calling system did not add an X-Correlation-ID field to the request, the response message will not contain one either.

**Table 5 – HTTP header information in API requests**

For OAuth2 authorization, an authorization token must be requested by calling the /oauth2/token function. The received token should be inserted into the HTTP header of each business call.

For basic authorization, the username and password pair should be base64 encoded and added to the HTTP header

#### **7.4.3 Example message**

Example of a request in the HTTP header.

Bearer type requests must be added for OAuth2 authorization while Basic type requests should be used for Basic authorization.

Parameter	Value
Authorization	Bearer APRug5AE4VGAzNKDPAoxugLiDp0b
Authorization	Basic Q2xpZW50SWRUaGF0Q2FuT25seVJIYWQ6c2VjcmV0MQ==

**Table 6 – Example of HTTP header values in the API request message**

## 7.5 OAuth2 token request

For OAuth2 authorization, a token must be requested before business calls.

### 7.5.1 Request message

URL: <https://core.api.posta.hu/oauth2/token>

http operation: **POST**

The request must be submitted as follows:

In the HTTP header a standard Basic authentication request should be used to submit the Authorization key value.

e.g. Authorization: Basic Q2xpZW50SWRUaGF0Q2FuT25seVJlYWQ6c2VjcmV0MQ==

- base64 encoded (API username (API Key): API account password (API Secret))  
in the message body
- OAuth2 grant type key value should be specified as the **client\_credentials** value in the body section of the message

Parameter	Length	Occurrence	Data Type	Description
client_id	N/A (XN)	1-1	String	<b>Mandatory.</b> Username is provided by the MPL API.
client_secret	N/A (XN)	1-1	String	<b>Mandatory.</b> Password is provided by the MPL API.
grant_type	N/A (XN)	1-1	String	<b>Mandatory.</b> Value: <b>client_credentials</b> <b>It should be added to the body section of the message.</b>

**Table 7 – Token request message**

Http header parameters must include the

**Content-Type:application/x-www-form-urlencoded**

parameter next to the **Authorization** key value.

### 7.5.2 Response message

The body of the response message contains the authorization token. If the response is successful, you will get an HTTP 200 (Ok) response code.

Field	Max. length	Occurrence	Data Type	Description
access_token	N/A (XN)	1-1	String	Authorization token
expires_in	N/A (XN)	1-1	String	Token expiration (in seconds) 3600 seconds

**Table 5 – Relevant fields of the token response message**

The authorization token expires in 3600 seconds. Once it has expired, you will get a 401 http code in the response message if you try to use it. In such a case, you will need to request a new token as described above.

### 7.5.3 Example message

#### Token Request

```

POST http://localhost:17463/oauth2/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic Q2xpZW50SWRUaGF0Q2FuT25seVJIYWQ6c2VjcmV0MQ==

grant_type=client_credentials

```

#### Token Response

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token": "APRug5AE4VGazNKDPAoxugLiDp0b",
  "issued_at": 1592910455065,
  "expires_in": 1799,
  "token_type": "Bearer"
}

```

### 7.6 Business service calls

The following link provides a detailed description of MPL API v2:  
<https://devportal.posta.hu/api/7>

## **8. Non-functional properties**

### **8.1 Availability**

#### **8.1.1 Service period**

The MPL API is available 24 hours a day on 365 days a year.

#### **8.1.2 Maintenance Period**

The planned maintenance period of the MPL API is specified in the General Terms and Conditions.

#### **8.1.3 Unavailability**

If the MPL API v2 interface is unavailable, then the calling system should be able to handle this situation.

### **8.2 Improved safety and security**

All MPL API operations are REST web service calls available through https protocol based requests. OAuth2 authorization can be used when calling service functions.

### **8.3 Unsupported JSON tags**

The MPL API ignores unsupported JSON tags.

## 9. Frequently Asked Questions

<https://devportal.posta.hu/faq>

## 10. Annex - Data types

The following JSON data types can be used in MPL API v2 operations:

<https://devportal.posta.hu/api/7>

## 11. Annex - Samples

The MPL API supports sending both domestic and international parcels on the same endpoint, so at first glance it may seem like a lot of data. To make things easier, here are some examples of how to send domestic and international parcels.

### 11.1 Example: submitting data for domestic parcels

#### 11.1.1 Parcel home delivery with cash on delivery

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments> address:

```
[
  {
    "sender": {
      "agreement": "10000319",
      "contact": {
        "name": "Feladó Flóra",
        "email": "fflora@gmail.com",
        "phone": "+361-1234567"
      },
      "address": {
        "postCode": "1215",
        "city": "Budapest",
        "address": "Szent István út 4",
        "remark": "készpénzes"
      }
    },
    "orderId": "0014",
    "developer": "AAA Kft",
    "webshopId": "1",
    "labelType": "A4",
    "item": [
      {
        "customData1": "ügyféladat1",
        "customData2": "ügyféladat2",
        "weight": {
          "value": "2560",
          "unit": "g"
        },
        "size": "L",
        "services": {
          "basic": "A_175_UZL",
          "extra": [
            "K_ORZ", "K_UVT", "K_ENY"
          ],
          "cod": "3000",
          "value": "3000",
          "deliveryMode": "HA"
        }
      }
    ],
    "recipient": {
      "contact": {
        "name": "Címzett Cecília",
        "email": "ccila@gmail.com",
        "phone": "+3670-1234567"
      },
      "address": {
        "postCode": "9023",
        "city": "Győr",
        "address": "Sport utca 1",
        "remark": "törékeny"
      }
    }
  }
]
```



```
    },
    "paymentMode": "UV_KP",
    "packageRetention": "10"
  }
]
```

Response for the request:

```
[{
  "webshopId": "1",
  "trackingNumber": "PKAAA50058206",
  "packageTrackingNumbers": ["PKAAA500582060011138001500"],
  "label": "JVBERi0xLjQKeLjz9..."
}]
```

### 11.1.2 Parcel delivery to a PostaPoint, with value insurance

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments> address:

```
[
  {
    "sender": {
      "agreement": "10000319",
      "contact": {
        "name": "Feladó Flóra",
        "email": "fflora@gmail.com",
        "phone": "+361-1234567"
      },
      "address": {
        "postCode": "1215",
        "city": "Budapest",
        "address": "Szent István út 4",
        "remark": "készpénzes"
      }
    },
    "orderId": "0014",
    "developer": "AAA Kft",
    "webshopId": "1",
    "labelType": "A5",
    "item": [
      {
        "customData1": "ügyféladat1",
        "customData2": "ügyféladat2",
        "weight": {
          "value": "2560",
          "unit": "g"
        },
        "size": "L",
        "services": {
          "basic": "A_175_UZL",
          "extra": [
            "K_ENY"
          ],
          "value": "3000",
          "deliveryMode": "PP"
        }
      }
    ],
    "recipient": {
      "contact": {
        "name": "Címzett Cecília",
        "email": "ccila@gmail.com",
        "phone": "+3670-1234567"
      }
    }
  }
]
```

```

    },
    "address": {
      "postCode": "9023",
      "city": "Győr",
      "address": "Sport utca 1",
      "remark": "MOL kútra",
      "parcelPickupSite": "12063 sz. MOL töltőállomás"
    }
  },
  "paymentMode": "UV_KP",
  "packageRetention": "10"
}
]

```

Response for the request:

```

[
  {
    "webshopId": "1",
    "trackingNumber": "PNAAA50000034",
    "packageTrackingNumbers": ["PNAAA500000340017101000000"],
    "label": "JVBERi0xLjQK..."
  }
]

```

### 11.1.3 Two-parcel consignment home delivery with cash on delivery and value insurance

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments> address:

```

[
  {
    "sender": {
      "agreement": "10000319",
      "contact": {
        "name": "Feladó Flóra",
        "email": "fflora@gmail.com",
        "phone": "+361-1234567"
      },
      "address": {
        "postCode": "1215",
        "city": "Budapest",
        "address": "Szent István út 4",
        "remark": "készpénzes"
      }
    },
    "orderId": "0014",
    "developer": "AAA Kft",
    "webshopId": "1",
    "labelType": "A5",
    "groupTogether": true,
    "item": [
      {
        "customData1": "ügyféladat1",
        "customData2": "ügyféladat2",
        "weight": {
          "value": "2560",
          "unit": "g"
        },
        "size": "L",
        "services": {
          "basic": "A_175_UZL",
          "extra": [
            "K_ORZ", "K_UVT", "K_ENY"
          ]
        }
      }
    ]
  }
]

```

```

        "cod": "3000",
        "value": "3000",
        "deliveryMode": "HA"
    }
},
{
    "weight": {
        "value": "1001",
        "unit": "g"
    },
    "size": "L",
    "services": {
        "basic": "A_175_UZL",
        "extra": [
            "K_ORZ", "K_UVT", "K_ENY"
        ],
        "cod": "4000",
        "value": "4000",
        "deliveryMode": "HA"
    }
},
],
"recipient": {
    "contact": {
        "name": "Címzett Cecília",
        "email": "ccila@gmail.com",
        "phone": "+3670-1234567"
    },
    "address": {
        "postCode": "9023",
        "city": "Győr",
        "address": "Sport utca 1",
        "remark": "törékeny"
    }
},
"paymentMode": "UV_KP",
"packageRetention": "10"
}
]

```

Response for the request:

```

[ {
    "webshopId": "1",
    "trackingNumber": "PKQZ050115785",
    "packageTrackingNumbers": ["PKQZ0501157850021138000000",...],
    "label": "JVBERi0..."
} ]

```

#### 11.1.4 Parcel home delivery with an exchange parcel

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments> address:

```

[
  {
    "sender": {
      "agreement": "55313477",
      "contact": {
        "name": "Feladó Flóra",
        "email": "fflora@gmail.com",
        "phone": "+361-1234567"
      },
      "address": {

```

```

        "postCode": "1215",
        "city": "Budapest",
        "address": "Szent István út 4",
        "remark": "készpénzes"
    },
    "orderId": "0014",
    "developer": "AAA Kft",
    "webshopId": "1",
    "labelType": "A5",
    "item": [
        {
            "customData1": "ügyféladat1",
            "customData2": "ügyféladat2",
            "weight": {
                "value": "2560",
                "unit": "g"
            },
            "size": "L",
            "services": {
                "basic": "A_175_UZL",
                "extra": [
                    "K_CSE"
                ],
                "deliveryMode": "HA"
            },
            "replacementPackage": {
                "customData1": "cserecsomag",
                "customData2": "visszaküldendő",
                "extra": ["K_ENY"],
                "weight": {
                    "value": "2000",
                    "unit": "G"
                },
                "value": "1000"
            }
        }
    ],
    "recipient": {
        "contact": {
            "name": "Címzett Cecília",
            "email": "ccila@gmail.com",
            "phone": "+3670-1234567"
        },
        "address": {
            "postCode": "9023",
            "city": "Győr",
            "address": "Sport utca 1",
            "remark": "törékeny",
        }
    },
    "paymentMode": "UV_KP",
    "packageRetention": "10"
}
]

```

Response for the request:

```

[
{
    "webshopId": "1",
    "trackingNumber": "PNBBZ50000092",
    "replacementTrackingNumber": "PIBBZ50000106",
    "replacementLabels": [
        {

```

```
"trackingNumber": "PIBBZ5000106",  
"label": "JVBERi0xLjQKJ..."
```

```
}]
```

### 11.1.5 Parcel home delivery, with inverse (~return delivery)

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments> address:

```
[  
  {  
    "sender": {  
      "agreement": "10000319",  
      "contact": {  
        "name": "Feladó Flóra",  
        "email": "fflora@gmail.com",  
        "phone": "+361-1234567"  
      },  
      "address": {  
        "postCode": "1215",  
        "city": "Budapest",  
        "address": "Szent István út 4",  
        "remark": "készpénzes"  
      }  
    },  
    "orderId": "0014",  
    "developer": "AAA Kft",  
    "webshopId": "1",  
    "labelType": "A4",  
    "item": [  
      {  
        "customData1": "ügyféladat1",  
        "customData2": "ügyféladat2",  
        "weight": {  
          "value": "2560",  
          "unit": "g"  
        },  
        "size": "L",  
        "services": {  
          "basic": "A_175_UZL",  
          "extra": [  
            "K_INV"  
          ],  
          "deliveryMode": "HA"  
        }  
      }  
    ],  
    "recipient": {  
      "contact": {  
        "name": "Címzett Cecília",  
        "email": "ccila@gmail.com",  
        "phone": "+3670-1234567"  
      },  
      "address": {  
        "postCode": "9023",  
        "city": "Győr",  
        "address": "Sport utca 1",  
        "remark": "törékeny",  
      }  
    },  
    "paymentMode": "UV_KP",  
    "packageRetention": "0"  
  }  
]
```

Response for the request:

```
[
  {
    "webshopId": "1",
    "trackingNumber": "PIBBZ50000119",
    "packageTrackingNumbers": [
      "PIBBZ500001190019023000000"
    ],
    "label": "JVBERi0xLjQKJ..."
  }
]
```

### 11.1.6 Parcel delivery to a parcel terminal, with eased handling, goods payment and value insurance

HTTP POST a <https://core.api.posta.hu/v2/mplapi/shipments> címre:

```
[
  {
    "sender": {
      "agreement": "10000319",
      "contact": {
        "name": "Feladó Flóra",
        "email": "fflora@gmail.com",
        "phone": "+361-1234567"
      },
      "address": {
        "postCode": "1215",
        "city": "Budapest",
        "address": "Szent István út 4",
        "remark": "készpénzes"
      }
    },
    "orderId": "0014",
    "developer": "AAA Kft",
    "webshopId": "1",
    "labelType": "A4",
    "item": [
      {
        "customData1": "ügyféladat1",
        "customData2": "ügyféladat2",
        "weight": {
          "value": "2560",
          "unit": "g"
        },
        "size": "L",
        "services": {
          "basic": "A_175_UZL",
          "extra": [
            "K_UVT", "K_ENY"
          ],
          "cod": "3000",
          "value": "3000",
          "deliveryMode": "CS"
        }
      }
    ],
    "recipient": {
      "contact": {
        "name": "Címzett Cecília",
        "email": "ccila@gmail.com",
        "phone": "+3670-1234567"
      }
    }
  }
]
```

```

    },
    "address": {
      "postCode": "9023",
      "city": "Győr",
      "address": "Sport utca 1",
      "remark": "törékeny",
      "parcelPickupSite": "20 sz. automata - Tatabánya 1 posta"
    },
    "disabled": true
  },
  "paymentMode": "UV_KP",
  "packageRetention": "10"
}
]

```

Response for the request:

```

[
  {
    "webshopId": "1",
    "trackingNumber": "PKBBZ50000122",
    "packageTrackingNumbers": [
      "PKBBZ500001220012801003000"
    ],
    "label": "JVBERi0xLjQKJ..."
  }
]

```

## 11.2 Example: submitting data for domestic international parcels

### 11.2.1 Submitting a non-EU parcel

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments> address:

```

[
  {
    "sender": {
      "agreement": "10000117",
      "contact": {
        "name": "Feladó Árvíztűrő Tükörfúrógép",
        "email": "felado.emailcime@posta.hu",
        "phone": "+3620-1234567",
        "organization": "Küldő Cég Árvíztűrő Tükörfúrógép"
      },
      "address": {
        "postCode": "1111",
        "city": "Budapest",
        "address": "Zenta utca 1",
        "remark": "Feladó megjegyzése Árvíztűrő tükörfúrógép",
        "countryCode": "HU"
      },
      "accountNo": "22222222-22222222",
    },
    "orderId": "2400",
    "developer": "AAA Kft",
    "webshopId": "3600",
    "item": [
      {
        "customData1": "Ügyféladat Egy",
        "customData2": "Ügyféladat Kettő",
        "weight": {
          "value": 12002,

```

```

        "unit": "G"
    },
    "size": "",
    "services": {
        "basic": "A_121_CSG",
        "extra": ["K_ENY", "K_UVT"],
        "cod": 999,
        "value": 653,
        "deliveryMode": "HA",
        "codCurrency": "EUR",
        "customsValue": 198,
        "otherComment": "Általános megjegyzés",
        "customsValueCurrency": "EUR",
        "produceContent": "11",
        "DescriptionOfGoods": 32
    },
    "ewcCode": "150101",
    "documents": [{
        "authorisationNr": "PSZ12345",
        "name": "325 Proforma számla"
    }, {
        "authorisationNr": "PSZ88006600",
        "name": "325 Proforma számla"
    }, {
        "authorisationNr": "KE1234567",
        "name": "811 Kiviteli engedély"
    }, {
        "authorisationNr": "KE100000",
        "name": "811 Kiviteli engedély"
    }, {
        "authorisationNr": "SZT9876543",
        "name": "861 Származási tanúsítvány"
    }, {
        "authorisationNr": "SZT100000",
        "name": "861 Származási tanúsítvány"
    }
    ],
    "customs": [{
        "produceCount": 3,
        "produceName": "Csomag 1",
        "produceValue": 10,
        "tariffCode": "987654",
        "country": "HU",
        "weightValue": 1234,
        "value": 0
    },
    {
        "produceCount": 5,
        "produceName": "Csomag 2",
        "produceValue": 22,
        "tariffCode": "98765432",
        "country": "CZ",
        "weightValue": 1200,
        "value": 0
    },
    {
        "produceCount": 2,
        "produceName": "Csomag 3",
        "produceValue": 8,
        "tariffCode": "9876543210",
        "country": "AT",
        "weightValue": 850,
        "value": 0
    },
    {
        "produceCount": 6,
        "produceName": "Csomag 4",

```



```

        "produceValue": 7,
        "tariffCode": "1020304050",
        "country": "TR",
        "weightValue": 100,
        "value": 0
    }],
    },
    "recipient": {
        "contact": {
            "name": "AYVALIK BALIKESİR",
            "email": "ayvalik.balikesir@gmail.com",
            "phone": "+902428152116",
            "organization": "Egy Török Cég Árvíztűrő Tükörfúrógép"
        },
        "address": {
            "postCode": "43502",
            "city": "Antalya",
            "address": "AKSU S PARK A N 13/1, Sıhhiye",
            "remark": "Nem jó a csengő, dudáljon már a futár",
            "countryCode": "TR"
        },
    },
    "paymentMode": "UV_AT",
    "packageRetention": 5
}]]

```

Response for the request:

```

[{"webshopId": "3600",
  "trackingNumber": "CC900105354HU",
  "packageTrackingNumbers": ["CC900105354HU"],
  "label": "JVBERi0xLjQKeLjz9MKMSAwIG9iago8..."}]

```

### 11.2.2 Submitting an EU parcel

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments> address:

```

[{"sender": {
  "agreement": "10000117",
  "contact": {
    "name": "Feladó Árvíztűrő Tükörfúrógép",
    "email": "felado.emailcime@posta.hu",
    "phone": "+3620-1234567",
    "organization": "Küldő Cég Árvíztűrő Tükörfúrógép"
  },
  "address": {
    "postCode": "1111",
    "city": "Budapest",
    "address": "Zenta utca 1",
    "remark": "Feladó megjegyzése Árvíztűrő tükörfúrógép",
    "countryCode": "HU"
  },
  "accountNo": "22222222-22222222",
},
  "orderId": "2400",
  "developer": "AAA Kft",
  "webshopId": "3600",
  "item": [{"

```

```

        "customData1": "Ügyféladat Egy",
        "customData2": "Ügyféladat Kettő",
        "weight": {
            "value": 12002,
            "unit": "G"
        },
        "size": "",
        "services": {
            "basic": "A_125_HAR",
            "extra": ["K_TER"],
            "deliveryMode": "HA",
            "customsValue": 0,
            "otherComment": "Általános megjegyzés",
            "customsValueCurrency": "EUR",
            "produceContent": "11",
            "DescriptionOfGoods": 32
        },
        "ewcCode": "150101",
    }
},
"recipient": {
    "contact": {
        "name": "Jürgen Hörmann",
        "email": "jurgen.hormann@gmail.com",
        "phone": "+431252307217",
        "organization": "Osztrák Cég Árvízturnó Tükörfúrógép"
    },
    "address": {
        "postCode": "B318",
        "city": "Vienna",
        "address": "Quellenstrasse 51",
        "remark": "Nem jó a csengő, dudáljon a futár",
        "countryCode": "AT"
    },
},
"paymentMode": "UV_AT",
"packageRetention": 5
}]

```

Response for the request:

```

[ {
    "webshopId": "3600",
    "trackingNumber": "JJH30AAAAAAT90032420",
    "packageTrackingNumbers": ["JJH30AAAAAAT90032420"],
    "label": "JVBERi0xLjQKJ..."
} ]

```

### 11.3 Closing the posting of a parcel

HTTP POST to <https://core.api.posta.hu/v2/mplapi/shipments/close> address:

```

{
    "checkList": true,
    "checkListWithPrice": true,
}

```

Response for the request:

```

[ {
    "manifest": "JVBERi0xL...",
} ]

```

```
"trackingNrPrices": [ {  
  "trackingNumber": "PNQZ0501157850011138000000",  
  "price": 1250  
}]  
}]
```