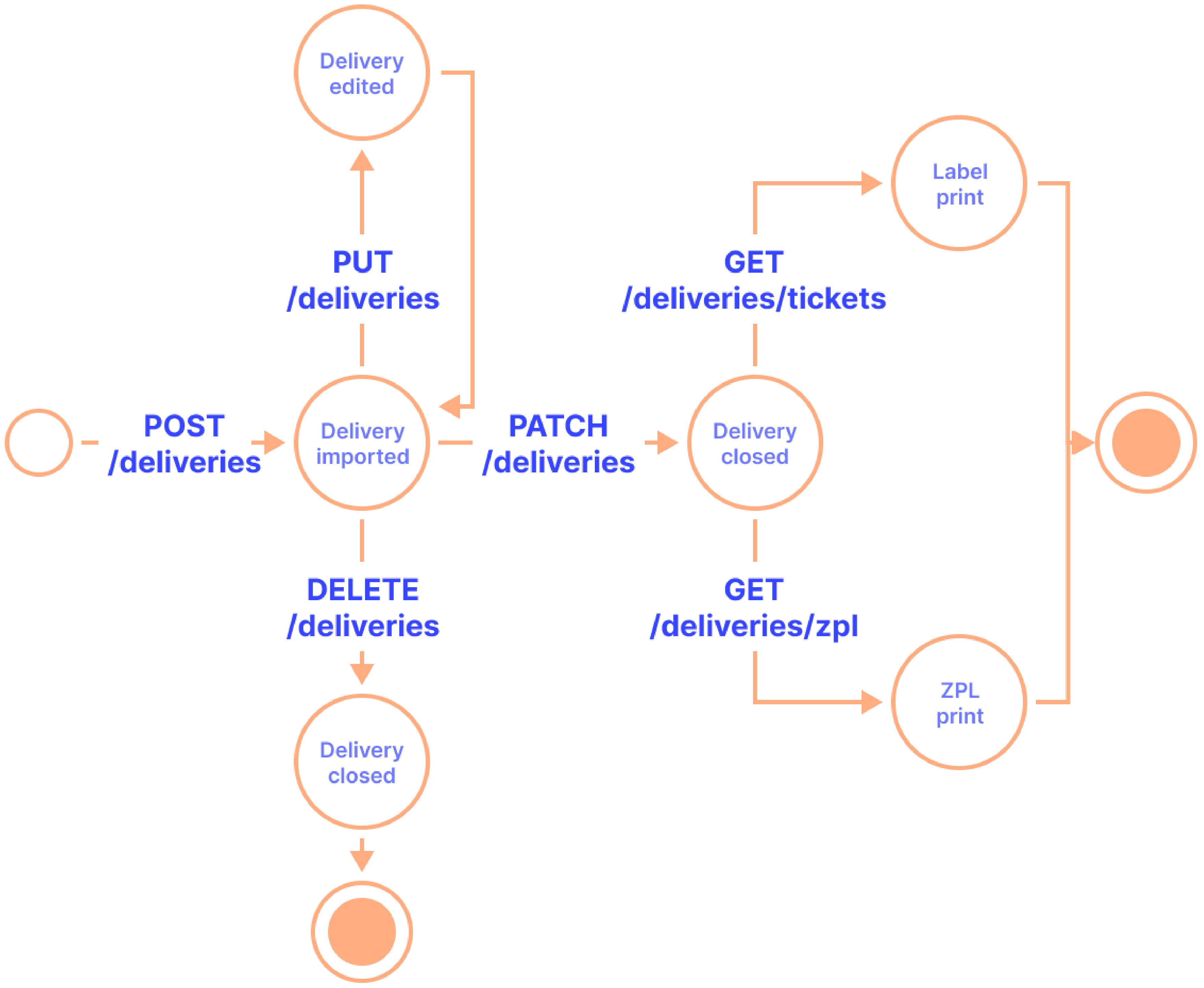
API Reference

For the correct functionality of the methods it is necessary to maintain the sequence according to the diagram. This means that it is necessary first to **Insert** (in this phase you can **cancel** and **edit**), and then **Close** the delivery. **Printing Labels** and print protocols can only be done for closed shipment.



Terminology

We use unified terminology in Foxdeli, API and for CSV Imports.A closer description of expressions is in the following table.

Czech expression	English Expression	API field	Description
Zásilka	Delivery	`delivery`	
Balík	Package	`package`	The shipment has one and more packages.
Přepravce	Agent	`agent`	IE.: Česká pošta, GLS, PPL, Zásilkovna,..
Přepravní služba	Delivery Type	`deliveryType`	IE.: Package to hand, Package to office, PickupPlaces,..
Doplňková služba	Extra Service	`extraService`	IE.: COD, Insurrance, SMS notification,..
Foxdeli ID zásilky	Delivery Foxdeli ID	`deliveryId`	ID in foxdeli system.

Číslo zásilky	Delivery Number	`deliveryNumber`	Delivery tracking number.
Číslo balíku	Package Number	`packageNumber`	The package number that is part of the shipment (mostly the same with the shipment number).
Svozové místo	Collection Place	`collectionPlace`	Collection place where the carrier will pick up deliveries.
Odběrné místo	Pickup Place	`pickupPlace`	Pickup place for delivery of shipments (dropoff point, pickup point, parcelshop, atd.)
Zpětná zásilka	Back Delivery	-	The shipment that travels from the customer back to you (mostly on the address of your collection).
Přímá zásilka	Direct Delivery	-	Shipment that is sent from the address to address.So it is not sent from your standard collection.
Výměnná zásilka	Swap Delivery	-	The shipment that is sent to the recipient with the prerequisite that it will be exchanged for the original shipment to give you back.
Vícekusá zásilka	Multipackage Delivery	-	Shipment that contains more than one package.

Supported methods

Endpoints for https://rest.foxdeli.com

Path	Supported HTTP methods	OAuth Scope	Endpointu description
/	• GET	-	A special unversioned test source without authentication
/v4/deliveries	• POST - Import new shipments • GET - getting information about deliveries • PATCH - close devliery • PUT - edit delivery • DELETE - cancel delivery	`deliveries`	Work with one or a few deliveries at once
/v4/deliveries/tickets	• GET	`deliveries`	Generating labels to PDF file for one or more deliveries
/v4/deliveries/zpl	• GET	`deliveries`	Generating labels in in ZPL format for one or more deliveries
/v4/deliveries/traces	• GET	`deliveries`	Track and trace states for one or more deliveries

/v4/collection-protocols	- POST - Import new shipments - GET - Getting information about deliveries	`collection-protocols`	Create a collection protocol, including PDF file generation
/v4/collection-places	- GET	`collection-places`	Obtaining information about all active collection places
/v4/list	- GET	`list`	Set of Endpoints to get enumerations (carrier lists, optional services, deliveries, label formats, etc.)

Deliveries

deliveries POST

Import deliveries to Foxdeli

Deliveries imported to Foxdeli will be created with state**1_0_0 - Work in progress**. In this state they are not yet sent to carrier API. That happens in the next step using PATCH.

During import basic validartion and reformat occurs..
Whether the delivery fulfills criteria given by carrier is verified during closure using PATCH.

Example of HTTP request (REQUEST)

According to exapmle bellow body can contain following items. Order of fields may differ and is not guaranteed. Type of values is only informative so it doesn't matter if request contains `2.3` or `"2.3"`.

URI can be extended by query parameter `fields` to specify what data should response include as with method GET.

Endpoint

```
POST: https://rest.foxdeli.com/v4/deliveries
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Ukázka těla

```
            "variableSymbol": "12345678",
            "cod": 1200,
            "codCurrency": "CZK",
            "value": 2000,
            "valueCurrency": "CZK",
            "packages": [
                {
                    "barcode": null,
                    "weight": 3,
                    "length": 15,
                    "width": 40,
                    "height": 20,
                    "containerCode": "EUP",
                    "containerItems": 5
                }
            ],
            "agent": "GLS",
            "deliveryType": "BP",
            "sender": {
                "type": "collectionPlace",
                "collectionPlace": "sokolovska-21"
            },
            "recipient": {
                "firstname": null,
                "surname": "Společnost s.r.o.",
                "contactPerson": null,
                "phone": "+420777111000",
                "email": "email@recipient.cz",
                "type": "address",
                "address": {
                    "city": "Praha",
                    "street": "Revoluční 11",
                    "postalCode": "11000",
                    "state": "CZ"
                }
            },
            "extraServices": [
                {
                    "code": "cod",
                    "arguments": []
                },
                {
                    "code": "email_advice_unload",
                    "arguments": {
                        "email": "email@recipient.com"
                    }
                },
                {
                    "code": "sms_advice_unload",
                    "arguments": {
                        "phone": "+420777111000"
                    }
                }
            ],
            "ticketNote": "Dodat do 2. podlaží",
            "externalId": "1234567"
        }
    ]
}
```

Description of request fields (REQUEST)

Field	Data type	Required	Validation	Description
`recipient` 1] `sender`	string[]	Required		The sender and recipient fields are typically identical, ie.They contain the same mandatory and optional parameters.
↳ `recipient|sender.type` 2]	string	Required	enum value, max 15 chars.	Recipient type.It decides on other fields that need to be mentioned for the recipient.Allowed values are: **address**, **collectionPlace**, **pickUpPlace**.

				normal address.Address parameters are: `street` , `streetNumber` , `state` , `city` , `postalCode` ;
↳ `recipient|sender.address.street`	string	Required	max 110 char with house no.	Street name;Must contain the number descriptive unless specified in `streetNumber`
↳ `recipient|sender.address.streetNumber`	string	COnditional		Number of descriptive;Mandatory only if not mentioned for space in `street`
↳ `recipient|sender.address.state`	string	Required	ISO 3166-2, 2 zn.	State in formatISO 3166-2, Thus, for example CZ or SK.
↳ `recipient|sender.address.city`	string	Required	max 127 char	Město
↳ `recipient|sender.address.postalCode`	string	Required	validation by state, max 15 char.	Postal code; Without spaces
↳ `recipient|sender.collectionPlace` [1]	string	COnditional	enum value, max 63 char	Identifier of collection place if recipient or sender is collection place
↳ `recipient|sender.pickUpPlace` [1]	string	COnditional	enum value, max 63 char	Identifier of collection placeif recipient is (Pickup, Dropoff, Parcelshop)
↳ `recipient|sender.firstname`	string	Optional	max 63 char.	Name (omit for company name)
↳ `recipient|sender.surname`	string	Required	max 127 char.	Surname of company name
↳ `recipient|sender.contactPerson`	string	Optional	max 127 char.	Contact Person - Name and Surname or Contact Person (for company)
↳ `recipient|sender.email` [1]	string	COnditional	RFC 2822, max 255 char	Email - May not be filled if the phone is filled
↳ `recipient|sender.phone` [1]	string	COnditional	Formát dle státu. Vždy s předvolbou a bez mezer.	Telefon - May not be filled if it is filled with an email.
`value`	float	Required	0.00	Consignment value in the currency in the key `valueCurrency` . Decimal places are separated by a dot.
`valueCurrency`	string	Required	ISO_4217	Currency delivery Values in Format ISO 4217 , tedy například CZK či EUR
`cod`	float	Optional	0.00	The cost of shipment in the currency specified in the key `codCurrency`
`codCurrency`	string	COnditional	ISO_4217	Currency cash on shipment in format ISO 4217, tedy například CZK či EUR, povinné, pokud je vyplněna dobírka
`variableSymbol`	string	COnditional	max 10 char.	Variable symbol for cash on delivery (mandatory if cash on delivery is filled `cod` and `codCurrency`)
`packages.weight` [3]	float	Optional	0.00	Package weight in kg.Decimal places are separated by a dot.
`packages.height` [3]	int	COnditional		Package height in cm.If you indicate fill in the remaining dimensions.
`packages.length` [3]	int	COnditional		Package length in cm.If you indicate fill in the remaining dimensions.
`packages.width` [3]	int	COnditional		Package width in cm.If you indicate fill in the remaining dimensions.

			3 char.	`that support Cargo transport.You can find a list of cargo units` `enumeration`.
`packages.containerItems` 4]	int	COnditional		Number of cargo units that have identical parameters.Only if you export a Cargo shipment with `Packages.ContainerCode` .The minimum is 1. Maximum sets a specific carrier.
`deliveryType`	string	Required	enum value, 2 char.	Transport service - identifier.See the list of identifierspenumeration.
`agent`	string	Required	enum value, max 7 char.	Carrier - identifier.See the list of identifiersenumeration.
`extraServices.code`	string	Required	enum value, max 63 char.	Additional services - identifier.See the list of identifiersenumeration.
`extraServices.arguments`	string	Optional	According to the parameter, max 255 char.	arguments for the additional service.The list of requested arguments of individual services is part of the same enumeration as an additional service.
`ticketNote`	string	Optional	max 255 char.	Note on the label.We recommend the shortest notes (about 35 characters) as possible.
`externalId`	string	Optional	max 127 char.	The external delivery identifier is used to identify the shipment.This key can also be used to obtain delivery information.
`platformKey`	string	Optional	max 255 char.	Platform identifier. Leave blank. To be filled in only by providers of CMS, WMS, OpenSource and add-ons for marketplaces in agreement with our technical support.

1] Objects `Recipient` and `Sender` have identical function and parameters.

2] Values for the recipient type `recipient.type` and sender type `Sender.Type` decide on other fields that need to specify for recipients or sender:

- **address** - For this type it is necessary to specify the person's data (`recipient.firstname` , `recipient.surname` , `recipient.address.street` , `recipient.address.city` , `recipient.address.postalCode` ,...).
- **collectionPlace** - This type only calls for a collection site identifier in the field `collectionPlace` . Address-related fields are no longer listed.
- **pickUpPlace** - This type requires identifier of pickup place `pickUpPlace` and the recipient's data(`recipient.firstname` , `recipient.surname` , `recipient.email` , `recipient.phone`).

3] May be mandatory for some carriers.Learn from the message when closing the shipments.You can use the Delivery/PUT method to complete the value.

4] Parameters for Cargo Transport.For deliveries outside Cargo, parameters do not fill or send as **null**.

Implicit additional services

These types decide on the implicit use of additional services direct and backward shipment:

- **Běžné svozové zásilky** jsou ze svozového místa ➞ na ➞ adresu (`recipient.type` má hodnotu **address** a `sender.type` má hodnotu **collectionPlace**).
- **Přímé zásilky** jsou z adresy ➞ na ➞ adresu (`recipient.type` a `sender.typedocs.js` mají hodnotu **address**).
- **Zpětné zásilky** jsou z adresy ➞ na ➞ svozové místo. (`recipient.type` má hodnotu **collectionPlace** a `sender.type` má hodnotu **address**).

In order to apply the reverse and direct shipments according to the above rules, it is necessary to have a filled type (regular | irregular collection) for the carrier. Collection place without type behaves a regular address - for example Transport from the collection place without the type of collection to the address will use a direct shipment. Excepton are carriers who do not differentiate collectable and non collectable deliveres (ie TNT) - they do not have these additional services.

unfilled entries and more. The data in response may differ slightly from those that were in the request.For example, there may be slight formatting modifications such as white space removal, value type change or similar changes.

Ukázka hlavičky

```
HTTP/1.1 201
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
ETag => "5887cca6c93d143c689ced8bc5080651"
Location => http://rest.foxdeli.com/v4/deliveries
```

Ukázka těla

```
    "message": "Deliveries successfully created!",
    "data": [
        {
            "agent": "GLS",
            "closed": "2020-09-23T15:05:40+02:00",
            "cod": 1200,
            "codCurrency": "CZK",
            "created": "2020-09-22T13:21:08+02:00",
            "deliveryId": 15023456,
            "deliveryMetaData": null,
            "deliveryNumber": null,
            "deliveryType": "BP",
            "externalId": "1234567",
            "foxdeliDetailUrl": "https://app.foxdeli.com/deliveries/15023456",
            "foxdeliTrackingUrl": "https://tt.foxdeli.com/status/mujshop/15023456?
sig=NGVjMjdjMmVhYjJiAtc5MGVmMzFmNjcyMzBlZjE5ZDliN2MxZjU1YmY3ZWE0MjQ1ZWUwZjQ4ZDg5MjJhZTZiNQ%3D%3D",
            "extraServices": [
                {
                    "code": "cod",
                    "arguments": []
                },
                {
                    "code": "email_advice_unload",
                    "arguments": {
                        "email": "email@recipient.com"
                    }
                },
                {
                    "code": "sms_advice_unload",
                    "arguments": {
                        "phone": "+420777111000"
                    }
                }
            ],
            "important": false,
            "inDelay": false,
            "notDelivered": 0,
            "notPickedUp": 0,
            "packages": [
                {
                    "barcode": "",
                    "weight": 3,
                    "length": 15,
                    "width": 40,
                    "height": 20
                }
            ],
            "recipient": {
                "firstname": null,
                "surname": "Společnost s.r.o.",
                "contactPerson": null,
                "phone": "+420777111000",
                "email": "email@recipient.cz",
                "type": "address",
                "address": {
                    "city": "Praha",
                    "street": "Revoluční 11",
                    "postalCode": "11000",
                    "state": "CZ"
                }
            },
            "sender": {
                "type": "collectionPlace",
                "collectionPlace": "sokolovska-21"
            },
            "source": 3,
            "sourceName": "API",
            "state": "1.0.0",
            "stateName": "Rozpracované",
            "stateChanged": "2020-09-11T07:09:39+02:00",
            "stateCategory": "1",
            "stateCategoryName": "Rozpracované",
            "stateSubcategory": "1.0",
            "stateSubcategoryName": "Rozpracované",
            "ticketNote": "Dodat do 2. podlaží",
            "value": 2000,
    ]
}
```

Description "Selected "Fields of successful response(RESPONSE)

Field	Data type	Description
`data[].deliveryId`	int	Delivery internal number.You can use this number to obtain information about the shipment over other methods in our API.
`data[].deliveryNumber`	string	Delivery number of the carrier - is supplemented after the shipment is closed.Mostly it is also " he tracking number".
`data[].created`	string	Delivery creation date.
`data[].packages.barcode`	string	Bale barcode (will be on the label).Compared to Deliverynumber, mostly differ in the control digit.
`data[].important`	bool	FLAG, which reflects a problematic shipment (for example, a large delay, nonwalk, etc.).
`data[].inDelay`	bool	FLAG, which reflects the delayed shipment (compared to the standard time delivery of shipments of the same carrier).
`data[].source`	int	The id of the origin of the delivery, possible values are: **prázdné** – Unknown **1** - Manual **2** - CSV **3** - API
`data[].sourceName`	string	Naming the origin of the delivery.
`data[].state` 1]	string	The main status of the shipment.
`data[].stateName` 1]	string	Delivery's main status name.
`data[].stateChanged` 1]	string	Change time (in case of exportation is time of creation in Foxdeli).
`data[].stateCategory` 1]	string	Delivery status category.
`data[].stateCategoryName` 1]	string	Delivery status category name.
`data[].stateSubcategory` 1]	string	Delivery status subcategory.
`data[].stateSubcategoryName` 1]	string	Name of delivery Subcategory status.
`data[].foxdeliDetailUrl`	string	URL delivery Details in FoxDeli.com (requires login)
`data[].foxdeliTrackingUrl`	string	URL details of the shipment leading to Branded Track & Amp;Trace Page (according to License)
`data[].monitored`	bool	Indications Whether the delivery behaves as a monitored shipment or standard. 2]

1] Fields related toTrack&Trace states. State enumerations can be foudn here.

2] Monitored shipments are only intended to monitor Track&Amp;Trace that are created by the Monitored-Deliveries method POST. More details here.

Validation and error codes

`field`) and the remaining deliveries insert in another request. These deliveries that have occurred an error you probably will want to mark in your system as erroneous and fix them. Most mistakes are translated into Czech - this can be requested by the previously mentioned HTTP header `Accept-language: cs` .

Error response example(RESPONSE)

If the shipments limit according to the current license is reached, it is returned **HTTP code 403**.

Ukázka hlavičky

```
HTTP/1.1 422
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
```

Ukázka těla

```
{
    "code": 422,
    "status": "error",
    "message": "Validation failed",
    "errors": [
        {
            "message": "Unknown extra service \"email_advice\" for given delivery type and address combination. Allowed codes are => email_advice_unload, sms_advice_unload",
            "field": "[0].extraServices[0].code",
            "value": "email_advice"
        }
    ]
}
```

Inserting more deliveries in one request

It is inappropriate and inefficient to insert multiple deliveries using several HTTP requests.Prefer to use one request with multiple items inside the field `deliveries`. This means that the treatment of data of your deliveries must be strict enough to pass deliveries to validation checks in our API.
If only the validation error occurs, it is possible to obtain specific shipments where the insertion failed and the request to repeat the request.

More information to repsonse

You will also find the `location` header in response that leads to a page with just created delvieries. Also, the `ETAG` header is present serving as cache for GET requests.

deliveries PATCH

Closure of imported deliveries for single collection place.

This operation sends data to particular carriers.

Method returns collection detail (for collection places, that do not use regular piccollectionkup), and delivery number of carrier and detail of deliveries.

Deliveries closed via Foxdeli will end up in state **2_0_0 - To send** and at this moment, carrier gets information about delivery to be collected and request for pickup (collection). Next state **3_0_0 - Sent** is automatically set based on carrier API response that courrier picked it up(usually first scan of label).

Example HTTP request (REQUEST)

Request accepts field `deliveries`, with one or more deliveries to close.

URI can be extended with query parametr `fields` to specify returned data fileds as with GET method.

Endpoint

```
PATCH: https://rest.foxdeli.com/v4/deliveries
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Ukázka těla

```
{
    "deliveries": [
        {
            "deliveryId": 15023456,
            "closed": true
        }
    ]
}
```

Description of request fields (REQUEST)

Field	Data type	Required	Description
`deliveryId`	int	Required	Foxdeli ID of delivery, that you received upon delivery creation via POST.
`closed`	bool	Required	Flag to close delivery. If you send multiple deliveries only those set to close `true` will be closed.
`?fields`	string	Optional	Parameter sent in URL. For more information see [deliveries/GET](#)

Example of successull response (RESPONSE)

All ordered collections are in the response and also includes data of all closed deliveries inlcuding **carriers delivery numbers** see. example response.

Struktura výpisu zásilek je totožná jako u odpovědi POST požadavku.

Ukázka hlavičky

```
HTTP/1.1 200
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
```

Ukázka těla

```
    "message": "Deliveries successfully closed!",
    "data": {
        "collectionOrders": [
            {
                "agent": "GLS",
                "scheduled": "2020-09-23",
                "collectionPlace": "sokolovska-21"
            }
        ],
        "deliveries": [
            {
                "agent": "GLS",
                "closed": "2020-09-23T15:05:40+02:00",
                "cod": 1200,
                "codCurrency": "CZK",
                "created": "2020-09-22T13:21:08+02:00",
                "deliveryId": 15023456,
                "deliveryMetaData": null,
                "deliveryNumber": "12859588454",
                "deliveryType": "BP",
                "externalId": "1234567",
                "foxdeliDetailUrl": "https://app.foxdeli.com/deliveries/15023456",
                "foxdeliTrackingUrl": "https://tt.foxdeli.com/status/mujshop/15023456?
sig=NGVjMjdjMmVhYjJiAtc5MGVmMzFmNjcyMzBlZjE5ZDliN2MxZjU1YmY3ZWE0MjQ1ZWUwZjQ4ZDg5MjhZTZiNQ%3D%3D",
                "extraServices": [
                    {
                        "code": "cod",
                        "arguments": []
                    },
                    {
                        "code": "email_advice_unload",
                        "arguments": {
                            "email": "email@recipient.com"
                        }
                    },
                    {
                        "code": "sms_advice_unload",
                        "arguments": {
                            "phone": "+420777111000"
                        }
                    }
                ],
                "important": false,
                "inDelay": false,
                "notDelivered": 0,
                "notPickedUp": 0,
                "packages": [
                    {
                        "barcode": "12859588454",
                        "weight": 3,
                        "length": 15,
                        "width": 40,
                        "height": 20,
                        "containerCode": "EUP",
                        "containerItems": 5
                    }
                ],
                "recipient": {
                    "firstname": null,
                    "surname": "Společnost s.r.o.",
                    "contactPerson": null,
                    "phone": "+420777111000",
                    "email": "email@recipient.cz",
                    "type": "address",
                    "address": {
                        "city": "Praha",
                        "street": "Revoluční 11",
                        "postalCode": "11000",
                        "state": "CZ"
                    }
                },
                "sender": {
                    "type": "collectionPlace",
                    "collectionPlace": "sokolovska-21"
                },
                "source": 3,
                "stateCategory": "2",
                "stateCategoryName": "K odeslání",
                "stateSubcategory": "2.0",
                "stateSubcategoryName": "K odeslání",
                "ticketNote": "Dodat do 2. podlaží",
                "value": 2000,
                "valueCurrency": "CZK",
                "variableSymbol": "12345678",
                "monitored": false
            }
        ]
    }
}
```

Description "of selceted" fields of successful response (RESPONSE)

Field	Data type	Description
`data[].collectionOrders[].agent`	string	Abbreviation of carrier where pickup was ordered
`data[].collectionOrders[].scheduled`	string	Date of pickup order YYYY-MM-DD, precise time will be shared directly by courrier.
`data[].collectionOrders[].collectionPlace`	string	Identifier of collection place, where pickup was ordered.
`deliveries[].deliveryNumber`	string	Carrieres delivery number - added after delivery closure. Mostly also revocation number.
`deliveries[].packages.barcode`	string	Barcode of package(also present on label). Compared to deliveryNumber differs in control number.

> ⚠ **Warning**
>
> You can only close deliveries with same pickup place of same carrier. All deliveries must be still notclosed a notcancelled.

Validation and error codes

This operation sends data to particular carriers.
In case of missing delivery **404** is returned. If you try to access delivery without permission **403** is returned.
For validation errors **422** is returned.

> ℹ **How does validation work**
>
> Except for basic format validation foxdeli leaves validation on carrier API. Validity of entered data is verified upon delivery closure using deliveries/PATCH.
> If one or more deliveries is rejected by carrierall information is handed back **via API directly to eshop**. After data is fixedit is possible to send fixed batch again, just like for method deliveries/POST, but using PUT method, that will fix existing batch. Right after that you can send batch to carrier API using method PATCH.

deliveries PUT

Editing of open deliveries.

Request Description (REQUEST)

In request, it is necessary to complete all the mandatory data of the shipment as the post methods and add key `deliveryId` (to the delivery field `deliveries[].deliveryId`).

Description of response (RESPONSE)

In the `data` fields are the same as the post method. If you indicate in `IF-Match` ETAG value for edited deliveries and delivery data changed (since obtaining the given Etag), the edit will not be executed and returned to HTTP code **412**.

With this mechanism, it is possible to check unexpected changes in deliveries in the web application.For proper operation, you need to use the ETAG obtained for all deliveries (ETAGs have been obtained by GET without Query parameter `fields`).

Validation and error codes

Validation is identical to the POST method, on top of that it validates `deliveryId`.Validation errors are marked with HTTP code **422**.
The response is the same structure as for insertion of deliveries.
It is not possible to edit closed and canceled deliveries.

---

deliveries GET

Receive information about deliveries

HTTP request example (REQUEST)

Most often you will search using `deliveryId` parameter. THis parameter must containFoxdeli ID of delivery, that was returned in POST response during delivery import. Particular delivery numbers must be split by comma(,).

Order of ids doesn't influence order in respnse!.

Using query param `fields` you can specify what fileds will be included in response. Multiple values are separated using comma(,). Possible param values are: `deliveryId`, `externalId`, `deliveryNumber`, `state`, `stateCategory`, `stateSubcategory`, `stateName`, `stateCategoryName`, `stateSubcategoryName`, `value`, `valueCurrency`, `closed`, `cod`, `codCurrency`, `source`, `sourceName`, `variableSymbol`, `ticketNote`, `created`, `agent`, `deliveryType`, `extraServices`, `recipient`, `sender`, `packages`, `foxdeliDetailUrl`, `agentTrackingUrl`.

Invalid values are ignored. If parameter is missing all data is returned.

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/deliveries?
deliveryId=15023456,15023457&fields=deliveryId,externalId,deliveryNumber,state,stateName,stateCategory,stateCategoryName,foxdeliDeta
ilUrl,agentTrackingUrl
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
```

Succesful response example(RESPONSE)

Response keeps standard structure (see above). Field data contais requested data and additional info. Data in response might be different than data requested. For example different formating was used.

Ukázka hlavičky

```
HTTP/1.1 200
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
Cache-Control => max-age=0
Expires => Thu, 24 Sep 2020 13:08:01 GMT
ETag => "5887cca6c93d143c689ced8bc5080651"
```

Ukázka těla

```
    "message": "Deliveries successfully retrieved.",
    "data": [
        {
            "agentTrackingUrl": "https://gls-group.eu/CZ/cs/sledovani-zasilek?match=1285958845412859588454",
            "deliveryId": 15023456,
            "externalId": "1234567",
            "deliveryNumber": "12859588454",
            "state": "6.0.0",
            "stateName": "Zrušeno",
            "stateCategory": "6",
            "stateCategoryName": "Zrušeno",
            "foxdeliDetailUrl": "https://app.foxdeli.com/deliveries/15023456"
        },
        {
            "agentTrackingUrl": "https://gls-group.eu/CZ/cs/sledovani-zasilek?match=12859588455",
            "deliveryId": 15023457,
            "externalId": 1234568,
            "deliveryNumber": "12859588455",
            "state": "3.1.2",
            "stateName": "Na doručení dnes",
            "stateCategory": "3",
            "stateCategoryName": "Doručované",
            "foxdeliDetailUrl": "https://app.foxdeli.com/deliveries/15023457"
        }
    ]
}
```

Description of response fields (RESPONSE) 1]

Field	Data type	Description
`agentTrackingUrl`	string	Actual tracking URL for T&T page of carrier. Link can change as different carrier can be used. Link will be returned only if foxdeli knows tracking number from carrier API.

1] Method GET can receive all parameters as POST method.

Search by other delivery params

Y ou can aslo search deliveries by other parameters.
All required criteria will be connected using `AND` operator. Params are searched using **fulltext or by value** - table bellow describes fields with predefined searches. Search by value can be done using comma),_ separated values in accordance with `deliveryId` search. You can also compare with a value using prefix > or <, see examples bellow. For example all deliveries with COD over 5000. This search will suppress default search method - it will not use fulltext search or value search.

Type search	Fields
fulltext	`(sender|recipient).contactPerson` , `(sender|recipient).firstname` , `(sender|recipient).surname` , `(sender|recipient).email` , `(sender|recipient).phone` , `ticketNote` , `created`
value	`deliveryId` , `externalId` , `agent` , `deliveryType` , `deliveryNumber` , `state` , `stateCategory` , `stateSubcategory` , `value` , `valueCurrency` , `deliveryId` , `packages.weight` , `packages.width` , `packages.height` , `packages.length` , `cod` , `codCurrency` , `source` , `variableSymbol`

Search examples (query parameters in URL)

Query string	Description
`?deliveryId=>10441`	All deliveries with ID above 10441 (eg. newer than given deliverytydx) - suitable for sync of deliveries entered manually or via CSV.
`?externalId=>B5007`	All deliveries with higher external ID than B5007 (lexicographically).
`?created=2014-12-24`	Deliveries created on christmas day.
`?deliveryNumber=M89`	Fulltext search by carrier delivery number.

`?value=>10000&valueCurrency=CZK`	Deliveries with value higher than 10 000.
`?value[]=>10000&value[]=<20000&valueCurrency=CZK`	Deliveries with value between 10 000 - 20 000 Kč.
`?variableSymbol=1234567890,9876543210`	Deliveries with variable symbol 1234567890 or 9876543210.

HTTP caching

In case of repeatedqueries for same resource using same filter or `fields` param, we recommend using rpeviously obtained ETag (part of all GET queries) in hader `If-None-Match`. In such case unless resource changed HTTP status **304** is returned along with empty body. Same principle as with root resource .

Validation and error codes

In case that delivery not foundhttp code **404** is returned.
Beware of cases when you search by mutliple `deliveryId` and some of them are invalid HTTP code **200** is returned and response contains only existing deliveries.

If you request large amount fo deliveries remember that URL length is limited to 8000 bytes. In such case an error HTTP code **414** is returned. Request for large amount of deliveries may take longer - we recommend using smaller batches to request large amount of deliveries.

> **Limitation**
> Max number of deliveries is **100**.

deliveries DELETE

Cancel open deliveries.

Example of HTTP request (REQUEST)

Every delivery needs to have an ID `deliveryId` according to example. One can only cancel unclosed and uncancelled deliveries.

Endpoint

```
DELETE: https://rest.foxdeli.com/v4/deliveries
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Ukázka těla

```
{
    "deliveries": [
        {
            "deliveryId": 15023456
        },
        {
            "deliveryId": 15023457
        }
    ]
}
```

Response description (RESPONSE)

each delivery (ETag was obtained using GET without query param `fields`).

Response contains `code`, `status` and `message`.

Validaction and error codes

Validation errors are marked HTTP code **422**.

Monitored Deliveriesy

> **Monitor your shipment better even if you don't export packages via Foxdeli**
>
> Independently of the way you link with API, you can use one of the most important features of our system. Creating a Monitored delivery in Foxdeli allows you to get detailed states, including delayed or problematic deliveries and own **Track & amp;Trace** page for your customers.

monitored-deliveries POST

Import deliveries with tracking numbers to Foxdeli for **Track & amp;Trace** Monitoring

Deliveries imported into Foxdeli will be created with the status **2_0_0 - To be sent**. It is assumed that the delivery already exists in the system of the carrier and is waiting for the courier, or is already on the way to the recipient.
You pass the delivery data to FoxDeli at the same time with the **TRACKING NUMBER** which is the main difference from standard deliveries.

Changing the status of the delivery in Foxdeli occurs automatically after first checking its status at the carrier (in order to a few minutes if a new state is available).

Compared to the standard deliveries, this method requires only a few basic parameters necessary for the correct tracking.

After you create a monitored delivery, you get the option to check on states using methods **deliveries/traces** GET Including delivery information using **deliveries** GET.

Example HTTP request (REQUEST)

API accepts all parameters as the Delivery method POST However, in this case, only the parameters listed in the example are required.

Endpoint

```
POST: https://rest.foxdeli.com/v4/monitored-deliveries
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Ukázka těla

```
            "tackingNumber": "CF55465884",
            "referenceId": "20210001",
            "value": 2000,
            "valueCurrency": "CZK",
            "agent": "GLS",
            "deliveryType": "BP",
            "sender": {
                "type": "collectionPlace",
                "collectionPlace": "sokolovska-21"
            },
            "recipient": {
                "firstname": null,
                "surname": "Společnost s.r.o.",
                "contactPerson": null,
                "phone": "+420777111000",
                "email": "email@recipient.cz",
                "type": "address",
                "address": {
                    "city": "Praha",
                    "street": "Revoluční 11",
                    "postalCode": "11000",
                    "state": "CZ"
                }
            }
        }
    ]
}
```

Request field description (REQUEST)

Field	Data type	Mandatory	Validation	Description
`trackingNumber`	string	Required	max 63 chars.	The tracking number of the delivery you obtained from the carrier.If Multi-package it is required to send the number of the master package (mostly it is the one that is the first in order, or that is given by cash on delivery).
`referenceId`	string	Optional	max 255 chars.	Your delivery Identifier - Use if you do not want to work with our DeliveryID that will be returned after the delivery export.The number must be unique.This identifier can also be used to obtain information about delivery using the method deliveries/GET or deliveries/traces/GET.
`recipient` | `sender`	string[]	Required		The sender and recipient fields are typically identical, ie.They contain the same mandatory and optional parameters. [1]
`value`	float	Required	0.00	Delivery value in the currency in the key `valueCurrency`. [1]
`valueCurrency`	string	Required	ISO_4217	Currency delivery Value [1]
`cod`	float	Optional	0.00	The cost of delivery in the currency specified in the key `codCurrency` [1]
`codCurrency`	string	Conditional	ISO_4217	Currency of delivery [1]
`deliveryType`	string	Required	enum key, 2 chars.	Přepravní služba - Identifikátor. Seznam identifikátorů naleznete v příslušném číselníku. [1]
`agent`	string	Required	enum key, max 7 chars.	Přepravce - Identifikátor. Seznam identifikátorů naleznete v příslušném číselníku. [1]

[1] The field details are described in the Delivery method POST, more here

Successful response example (RESPONSE)

The response retains the standard structure (see above). In the Data field you will find data from request + additional information as unfilled items and more. The data in response may differ slightly from those that were in the request.For example, there may be slight formatting modifications such as white space removal, value type change or similar changes.

Ukázka hlavičky

```
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
ETag => "5887cca6c93d143c689ced8bc5080651"
Location => http://rest.foxdeli.com/v4/monitored-deliveries
```

Ukázka těla

```
{
    "code": 201,
    "status": "success",
    "message": "Deliveries successfully created!",
    "data": [
        {
            "trackingNumber": "CF55465884",
            "referenceId": "20210001",
            "agent": "GLS",
            "cod": 1200,
            "codCurrency": "CZK",
            "created": "2020-09-22T13:21:08+02:00",
            "deliveryId": 15023456,
            "deliveryMetaData": null,
            "deliveryNumber": "CF55465884",
            "deliveryType": "BP",
            "externalId": "1234567",
            "foxdeliDetailUrl": "https://app.foxdeli.com/deliveries/15023456",
            "foxdeliTrackingUrl": "https://tt.foxdeli.com/status/mujshop/15023456?
sig=NGVjMjdjMmVhYjJiAtc5MGVmMzFmNjcyMzBlZjE5ZDliN2MxZjU1YmY3ZWE0MjQ1ZWUwZjQ4ZDg5MjJhZTZiNQ%3D%3D",
            "extraServices": [],
            "important": false,
            "inDelay": false,
            "notDelivered": 0,
            "notPickedUp": 0,
            "packages": [
                []
            ],
            "recipient": {
                "firstname": null,
                "surname": "Společnost s.r.o.",
                "contactPerson": null,
                "phone": "+420777111000",
                "email": "email@recipient.cz",
                "type": "address",
                "address": {
                    "city": "Praha",
                    "street": "Revoluční 11",
                    "postalCode": "11000",
                    "state": "CZ"
                }
            },
            "sender": {
                "type": "collectionPlace",
                "collectionPlace": "sokolovska-21"
            },
            "source": 3,
            "sourceName": "API",
            "state": "2.0.0",
            "stateName": "K odeslání",
            "stateChanged": "2020-09-11T07:09:39+02:00",
            "stateCategory": "2",
            "stateCategoryName": "K odeslání",
            "stateSubcategory": "2.0",
            "stateSubcategoryName": "K odeslání",
            "value": 2000,
            "valueCurrency": "CZK",
            "variableSymbol": "12345678",
            "monitored": true
        }
    ]
}
```

Note. Field description of response can be found in the method description deliveries/POST here.

> POST method.
> Methods for editing PUT and deletion DELETE are in development.

monitored-deliveries PUT

Editing Monitored deliveries.

> ⚠ **Will be implemented**

monitored-deliveries DELETE

Canceling Monitored Shipments.

> ⚠ **Will be implemented**

Labels

deliveries/tickets GET - PDF Labels

Generating labels to PDF file in Base64.

Request description (REQUEST)

The labels can be generated on the defined positions on paper A4, the default is the first position on the page. Required deliveries are Defined via Query parameter `delivery `similar to obtaining deliveries (GET), ie more numbers must be separated by commas. The initial position of the first label is determined by the optional Query parameter `position`. This parameter can take values from one to the maximum according to the carrier. The number of positions for individual carriers can be found in label specification (field Arch A4).

Another optional parameter is `printFormat`. Possible values are in the table below.

Values for `printFormat`

Key	Description
default	Printing A4 (default when not specifying parameter).At the same time, `Position` Label positions.
single	Print labels on the disc.Each label in the PDF document will be a separate page in the exact size of the carrier label.

Example HTTP request (REQUEST)

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
```

Response description (RESPONSE)

As all the responses, this respects the desired format according to the header `Accept`. Binary data generated PDF file are returned in format [base64 accroding to MIME spec]().

In your application, it is necessary before saving the file to decode content back into binary data.In most languages this is not a problem, see [PHP]() and [Java]().

Succesful response example (RESPONSE)

The label data in the example is just sample and do not contain the actual label.

Ukázka hlavičky

```
HTTP/1.1 200
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
Cache-Control => max-age=0
Expires => Thu, 24 Sep 2020 13:08:01 GMT
ETag => "7e43086bf308a5bfc7c58741b4443683"
```

Ukázka těla

```
{
    "code": 200,
    "status": "success",
    "message": "Tickets successfully generated",
    "data": [
        {
            "created": "2020-09-22T14:21:31+02:00",
            "size": 82348,
            "contents": "JVBERi0xLg0KdHJhaWxlcjw8L1Jvb3Q8PC9QYWdlczw8L0tpZHNbPDwvTWVkaWFCb3hbMCAwIDMgM10+Pl0+Pj4+Pj4="
        },
        {
            "created": "2020-09-22T14:21:31+02:01",
            "size": 67798,
            "contents": "BHBERi0xLg0KdHJhaWxlcjw8L1Jvb3Q8PC9QYWdlczw8L0tpZHNbPDwvTWVkaWFCb3hbMCAwIDMgM10+Pl0+Pj4+Pj8="
        }
    ]
}
```

Description of "selected" fields of successful response (RESPONSE)

Field	Data type	Description
`data[].created`	string	Date of creation of a PDF file.
`data[].size`	int	Information about file size before encoding into Base64
`data[].contents`	string	PDF file content in Base64 MIME format.Keep in mind that generating large PDF files is time-consuming operations and total processing can take several seconds. We recommend reasonable distribution for several doses (for example, up to 50pcs at once).

Validation and error codes

Warning

1) It is not possible to generate labels for several carriers at once, as each of them has other labels.
2) It is not possible to generate labels for deliveries with Additional Service "Sendback" because the package is not picked up on your collection place.Printing would not make sense.
3) The processing of a large number of labels in one request can have a negative impact on Reponse Time. If you experience problems with mass retrieval of labels, we recommend queries to be divided into smaller batches.

deliveries/zpl GET - ZPL labels

Generating labels to ZPL format.

Request description (REQUEST)

Labels can be generated for multiple deliveries at once, but always from just one carrier.

Required deliveries are defined via Query parameter `DeliveryID` similarly to obtaining deliveries, ie more numbers separated by comma. Generataion of labels can only be used for carriers with integrated ZPL support. Label size and dpi are defined by parameter `size` and `dpi`. If the parameters are not specified a default combination of size and dpi label is used.

ZPL formats

SUpported ZPL formats can be found in label specification or in structured list in API see desc.

Example HTTP request (REQUEST)

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/deliveries/zpl?deliveryId=15023456,15023457&size=10x15&dpi=300
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Response description(RESPONSE)

The answer contains the data of all labels to the required delivery.If the deliver is multipackage, see the labels for each piece.

Successful response example (RESPONSE)

The label data in the example is just sample and do not contain the actual label.

Ukázka hlavičky

```
HTTP/1.1 200
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
Cache-Control => max-age=0
Expires => Thu, 24 Sep 2020 13:15:01 GMT
ETag => "153ee69dbc346eac73ccedb5c03ac4d5"
```

```
    "status": "success",
    "message": "ZPL tickets successfully retrieved",
    "data": [
        {
            "deliveryId": 15023456,
            "contents":
"^XA^CI28^CWZ,E:TT0003M_.FNT^FS^BY3^FO60,10^B2B,150,N,N,N^FD000609431990^FS^FT50,360^AZB50,50^FWB^FD0006^FS^FT50,210^AZB40,40^FWB^FD0943199^FS^FT50,40^AZB30,30^FWB^FD0^F..."
        },
        {
            "deliveryId": 15023457,
            "contents":
"^XA^CI28^CWZ,E5GGVD_.FNT^FS^BY3^FO60,10^B2B,150,N,N,N^FD000609431990^FS^FT50,360^AZB50,50^FWB^FD0006^FS^FT50,210^AZB40,40^FWB^FD0943199^FS^FT50,40^AZB30,30^FWB^FD0^DF..."
        }
    ]
}
```

Validation and error codes

Labels can only be generated for closed deliveries. In the event of no finding of some of the deliveries or label, HTTP code **404** is returned. When attempting to access a delivery with insufficient privileges, HTTP code **403** is returned. Other validation errors, such as attempts to print unclosed delivery are indicated by code **422** .

Traces

deliveries/traces GET

Getting detailed track and trace states to selected deliveries.

Example HTTP request (REQUEST)

Required deliveries are defined via Query Parameter `DeliveryID` similar to obtaining deliveries, ie more numbers must be separated by commas.

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/deliveries/traces?deliveryId=15023456,15023457
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Response description (RESPONSE)

In response, you will find states to all the required deliveriesrs in the key `traces` .
The time of the last check of the states is stored under the key `lastChecked` For the delivery (before first check may not contain any value).
If the delivery has not yet been sent, the field may be `traces` empty.

Successful response example (RESPONSE)

Ukázka hlavičky

```
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
Cache-Control => max-age=0
Expires => Thu, 26 Sep 2020 13:15:01 GMT
ETag => "8887cca6c93d143c689ced8bc5080651"
```

Ukázka těla

```
    "message": "Traces successfully retrieved",
    "data": [
        {
            "deliveryId": 15023456,
            "lastChecked": "2020-09-25T10:30:00+02:00",
            "traces": [
                {
                    "type": "state",
                    "date": "2020-09-20T18:00:37+02:00",
                    "text": "Zásilka byla vydána.",
                    "flag": "",
                    "state": "4.0.0",
                    "stateSubcategory": "4.0",
                    "stateCategory": "4"
                },
                {
                    "type": "flag",
                    "date": "2020-09-20T00:00:00+02:00",
                    "text": "Zpožděné doručení",
                    "flag": "notDelivered3Days",
                    "state": "",
                    "stateSubcategory": "",
                    "stateCategory": ""
                },
                {
                    "type": "flag",
                    "date": "2020-09-19T00:00:00+02:00",
                    "text": "Mírně zpožděné doručení",
                    "flag": "notDelivered2Days",
                    "state": "",
                    "stateSubcategory": "",
                    "stateCategory": ""
                },
                {
                    "type": "state",
                    "date": "2020-09-17T14:24:35+02:00",
                    "text": "Zásilka byla přijata na cílovém výdejním místě Nymburk, Pražská 2261.",
                    "flag": "",
                    "state": "3.1.4",
                    "stateSubcategory": "3.1",
                    "stateCategory": "3"
                },
                {
                    "type": "state",
                    "date": "2020-09-17T10:20:51+02:00",
                    "text": "Zásilka byla odeslána na pobočku Nymburk, Pražská 2261",
                    "flag": "",
                    "state": "3.1.3",
                    "stateSubcategory": "3.1",
                    "stateCategory": "3"
                },
                {
                    "type": "state",
                    "date": "2020-09-17T05:19:33+02:00",
                    "text": "Zásilka byla přijata na depu Zásilkovna, DEPO, Praha 19, Satalice, U Arborky 696.",
                    "flag": "",
                    "state": "3.1.3",
                    "stateSubcategory": "3.1",
                    "stateCategory": "3"
                },
                {
                    "type": "state",
                    "date": "2020-09-17T05:19:22+02:00",
                    "text": "Zásilka byla připravena k odeslání na depu Zásilkovna, DEPO, Praha 19, Satalice, U Arborky 696.",
                    "flag": "",
                    "state": "3.1.3",
                    "stateSubcategory": "3.1",
                    "stateCategory": "3"
                },
                {
                    "type": "state",
                    "date": "2020-09-16T20:34:47+02:00",
                    "text": "Zásilka byla odeslána na pobočku DEPO, Praha 19, Satalice, U Arborky 696",
                    "flag": "",
                    "state": "3.1.3",
                    "stateSubcategory": "3.1",
                    "type": "state",
                    "date": "2020-09-16T19:51:27+02:00",
                    "text": "Zásilka byla odeslána na pobočku DEPO, Praha 19, Satalice, U Arborky 696",
                    "flag": "",
                    "state": "3.1.3",
                    "stateSubcategory": "3.1",
                    "stateCategory": "3"
                },
                {
                    "type": "state",
                    "date": "2020-09-16T19:32:48+02:00",
                    "text": "Zásilka byla přijata na depu Zásilkovna, DEPO, Ostrava, Frýdecká 700/475.",
                    "flag": "",
                    "state": "3.1.3",
                    "stateSubcategory": "3.1",
                    "stateCategory": "3"
                },
                {
                    "type": "state",
                    "date": "2020-09-16T19:32:41+02:00",
                    "text": "Zásilka byla připravena k odeslání na depu Zásilkovna, DEPO, Ostrava, Frýdecká 700/475.",
                    "flag": "",
                    "state": "3.1.3",
                    "stateSubcategory": "3.1",
                    "stateCategory": "3"
                },
                {
                    "type": "state",
                    "date": "2020-09-16T14:20:35+02:00",
                    "text": "Internetový obchod předal informace o zásilce.",
                    "flag": "",
                    "state": "2.0.0",
                    "stateSubcategory": "2.0",
                    "stateCategory": "2"
                },
                {
                    "type": "state",
                    "date": "2020-09-16T14:20:34+02:00",
                    "text": "Zásilka uzavřena v foxdeli",
                    "flag": "",
                    "state": "2.0.0",
                    "stateSubcategory": "2.0",
                    "stateCategory": "2"
                },
                {
                    "type": "state",
                    "date": "2020-09-16T14:20:32+02:00",
                    "text": "Zásilka vytvořena v foxdeli",
                    "flag": "",
                    "state": "1.0.0",
                    "stateSubcategory": "1.0",
                    "stateCategory": "1"
                }
            ]
        },
        {
            "deliveryId": 15023457,
            "lastChecked": "2020-09-25T10:30:00+02:00",
            "traces": []
        }
    ]
}
```

Dscription of "selected" fields in `data` of successful response (RESPONSE)

Field	Data type	Description
`deliveryId`	int	Tracking number.When checking multiple shipments in one query, distinguish particular items according to this key and not in order.
`lastChecked`	string	Last delivery check Time in the carrier API.

`traces[]`	array	Individual delivery states sorted from the latest after the oldest (desc).
↳ `traces[].type`	string	Status type. We differentiate **state** - delivery state **flag** - delivery flag
↳ `traces[].date`	string	Date of status change on carrier side.
↳ `traces[].text`	string	Information from the carrier or Foxdeli.
↳ `traces[].flag`	string	The value specifies what Flag is used.Possible Flags can be found in the Status tablehere.
↳ `traces[].state`	string	Current exact status of the shipment in the greatest detail.Status and categories can be found here.
↳ `traces[].stateSubcategory`	string	More accurate detail of the status of the main category.
↳ `traces[].stateCategory`	string	Main categories of status reflecting Section of deliveries in FoxDeli.

Validation and error codes

Statuses can only be obtained for closed deliveries.In case of not finding some of the deliveries, HTTP code is returned **404**. HTTP code **403** is returned when attempting to access a consignment with insufficient authorization. Other validation errors, such as an attempt to obtain a state of unclosed shipment are indicated by code **422**.

Collection Protocols

collection-protocols POST

Creation of collection protocol including PDF file generation.

Request description(REQUEST)

Collection protocol is used as an overview of shipments supplied to carrier during given collection. To create a collection protocol it is enough to define ID fo collection place and carrier. Values of fields are identical to fields used in creation of shipments, eg. `collectionPlace` and `agent`.

You can also send the deliveries numbers you want to have on the protocol. The optional `deliveries` container is used for this. Requested deliveries, however, must be closed, undelivered, and not on another collection log, otherwise an error message will be returned.

HTTP request example (REQUEST)

Endpoint

```
POST: https://rest.foxdeli.com/v4/collection-protocols
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Ukázka těla

```
    "deliveries": [
        15023456,
        15023457
    ]
}
```

Response description(RESPONSE)

Header `Location` contains URI to created protocol, nevertheless the data is also included in this response. Response contains key `data` where you will find all the data including PDF file.

Example of successfull response (RESPONSE)

Label data is just an example and doesn't contain actual label data.

Ukázka hlavičky

```
HTTP/1.1 201
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
Location => /v4/collection-protocols?collectionProtocolId=1425
```

Ukázka těla

```
{
    "code": 201,
    "status": "success",
    "message": "Collection protocol successfully created!",
    "data": {
        "collectionProtocolId": 1425,
        "agent": "PPL",
        "collectionPlace": "sokolovska-21",
        "protocol": "JVBERi0xLg0KdHJhaWxlcjw8L1Jvb3Q8PC9QYWdlczw8L0tpZHNbPDwvTWVkaWFCb3hbMCAwIDMgM10+Pl0+Pj4+Pj4=",
        "created": "2020-09-25T08:07:45+02:00",
        "deliveries": [
            15023456,
            15023457,
            15023458
        ]
    }
}
```

Response description (RESPONSE)

Field	Data type	Description
`collectionProtocolId`	string	ID fo colleciton protocol, used in URL to get collection protocol detail
`agent`	string	Carrier abbreviation, that was used to generate protocol
`protocol`	string	Content of PDF encoded as base64 MIME
`created`	string	Date of protocol creation
`deliveries`	string	array of shipment IDs that are included in protocol

Validation and error codes

Collection protocol can be generated only if given colleciton place has any unclosed shipments of given carrier without collection protocol.

All validation errors return code **422** including array of `errors` with standard structure and identical error codes as fo shipment creation.

For general error codes (like non existent shipments for protocol) `errors` is unavailable. It is only possilbe to generate 1 protocol per request.

collection-protocols GET

Obtain infromation about colleciton protocol including PDF file content.

Request description (REQUEST)

JAs a query paramater `collectionProtocolId` use value of `collectionProtocolId` received in response of collection protocol creation. It is possible to get only 1 protocol at a time.

HTTP request example (REQUEST)

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/collection-protocols?collectionProtocolId=1425
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
```

Response description (RESPONSE)

Response in `data` key is identical to response of collection protocol creatipn. See description above.

Successful response example (RESPONSE)

Data of label in example is just example and doesn't contain actual label.

Ukázka hlavičky

```
HTTP/1.1 200
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
```

Ukázka těla

```
{
    "code": 200,
    "status": "success",
    "message": "Protocol successfully fetched!",
    "data": {
        "collectionProtocolId": 1425,
        "agent": "PPL",
        "collectionPlace": "sokolovska-21",
        "protocol": "JVBERi0xLg0KdHJhaWxlcjw8L1Jvb3Q8PC9QYWdlczw8L0tpZHNbPDwvTWVkaWFCb3hbMCAwIDMgM10+Pl0+Pj4+Pj4=",
        "created": "2020-09-25T08:07:45+02:00",
        "deliveries": [
            15023456,
            15023457,
            15023458
        ]
    }
}
```

Validaction and error codes

If collection protocol doesn't exist **404** is returned or **403** unauthenticated.

collection-places GET

Obtain information about all collection places.

HTTP request example (REQUEST)

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/collection-places
```

Ukázka hlavičky

```
Host => rest.foxdeli.com
Authorization => Basic b66f6b694032e4271710deb21d7a7d703c974854a9a09a72825115bce69cb369
Accept => application/json
Content-type => application/json
x-http-method-override => GET
```

Response description (RESPONSE)

Field `data` contains information about collection places with following data.

Successful response example (RESPONSE)

Label data in repsonse doesn't contain actual label.

Ukázka hlavičky

```
HTTP/1.1 200
X-Frame-Options => SAMEORIGIN
Vary => X-Requested-With
X-Powered-By => Lean Mapper
X-Robots-Tag => noindex
Content-Type => application/json; charset=UTF-8
Cache-Control => max-age=0
Expires => Thu, 26 Sep 2020 13:15:01 GMT
ETag => "8887cca6c93d143c689ced8bc5080651"
```

Ukázka těla

```
    "message": "Active collection places successfully fetched",
    "data": [
        {
            "name": "Stará 251",
            "identificator": "stara-251",
            "email": "obchod@mujeshopik.cz",
            "phone": "+420702358586",
            "contactPerson": null,
            "state": "CZ",
            "city": "Bohumín",
            "street": "Stará 251",
            "postalCode": "73552"
        },
        {
            "name": "Sokolovská 21, Praha",
            "identificator": "sokolovska-21",
            "email": "obchod@mujeshopik.cz",
            "phone": "+420702358586",
            "contactPerson": null,
            "state": "CZ",
            "city": "Praha",
            "street": "Sokolovská 51",
            "postalCode": "18000"
        }
    ]
}
```

Description of response fields (RESPONSE)

Array	Data type	Description
`identificator`	string	Unique ID of collection place, used to print colleciton protocols
`name`	string	Name of collection place
`email`	string	email of contact person
`phone`	string	phone of contact person
`contactPerson`	string	conrtact person
`state`	string	State in format [ISO 3166-2](), for example CZ
`city`	string	City
`street`	string	Street and number
`postalCode`	string	Postal code

Enumerations

You can also get comprehensive and current enumerations of supported carriers `agent` , optional `extra-service` , delivery states `state` A supported label formats for individual carriers.

Root endpoint **https://rest.foxdeli.com/v4/list** works as a signpost for individual enumerations.

Getting the enumerations is public and does not require authentication.

Please note that the default response format is XML (and is used when accessing the browser).If you query through your system you can get an answer in JSON using header `Accept: application/json`

For the purpose of API documentation, we are showing only in JSON format.

ACtual enumerations with values can be found in table [in this part of documentation]().

list/agents GET

Personalized carrier list

To receive carriers and services for your account just change endpoint to **https://rest.foxdeli.com/v4/list/agents/account-only**. Only the carriers you use will be available at this address. Endpoint **account-only** requires authentication.

HTTP request example(REQUEST)

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/list/agents
```

Example of successful response(RESPONSE)

The example contains only the example part of the answer.

Ukázka těla

```
{
    "code": 200,
    "status": "success",
    "message": "Foxdeli list of agents",
    "data": [
        {
            "abbr": "DPD",
            "fullname": "Direct Parcel Distribution CZ s. r. o.",
            "isActive": 1,
            "hasTicketPrint": 1,
            "hasProtocolPrint": 1,
            "deliveryTypes": [
                {
                    "abbr": "DQ",
                    "fullname": "DPD AirExpress",
                    "isActive": 1,
                    "isPickUpPlaceType": 0,
                    "isCargoType": 0,
                    "description": "Standardní balíková přeprava bez notifikací"
                },
                {
                    "abbr": "DJ",
                    "fullname": "DPD Classic",
                    "isActive": 1,
                    "isPickUpPlaceType": 0,
                    "isCargoType": 0,
                    "description": "Letecká přeprava"
                }
            ]
        }
    ]
}
```

Response field description (RESPONSE)

Field	Data type	Values	Description
`data`	array		List of carriers
↳ `abbr`	string		Abbreviation of the carrier in Foxdeli (Abbrevation).Used in API as `agent`.
↳ `fullname`	string		The entire carrier name
↳ `isActive`	tinyint	**0, 1** [1]	Indication whether the carrier can be used to create shipments.In case of termination of cooperation with the carrier, the value will be **0**.
↳ `hasTicketPrint`	tinyint	**0, 1** [1]	Indication whether Foxdeli allows you to print labels for a given carrier.
↳ `hasProtocolPrint`	tinyint	**0, 1** [1]	Indication whether Foxdeli allows for a given carrier to print a collection protocol.
↳ `deliveryTypes`	array		List of Supported Shipping Services of the carrier.
↳ `abbr`	string		Abbreviation of shipping service in Foxdeli (Abbrevation).Used in API as `deliveryType`.

↳ `isActive`	tinyint	**0, 1** [1]	Indication whether the transport service can be used to create shipments.In case of disabling of the service, the value**0**.
↳ `isPickUpPlaceType`	tinyint	**0, 1** [1]	Indications Whether the transport service goes with the pickup point.In most cases, the beneficiary identifier is also necessary in the recipient data `pickUpPlace` .
↳ `isCargoType`	tinyint	**0, 1** [1]	Indication if the transport service is designed for Cargo transport.Cargo services require sending parameters `containerCode` and `containerItems` in the definition of packages `packages` .
↳ `description`	string		Additional information on transport service

Information

[1] **1** - yes, **0** - no

list/extra-services GET

Getting a list of optional `extraservices` with details and support for individual carriers.

Example HTTP request (REQUEST)

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/list/extra-services
```

Succesful response example (RESPONSE)

The example contains only a demonstrative part of the answer.

Ukázka těla

```
    "message": "Foxdeli list of extra services",
    "data": [
        {
            "code": "insurance",
            "fullname": "Připojištění",
            "description": "Dodatečné jednorázové připojištění k jednotlivým balíkům.",
            "isActive": 1,
            "isImplicitOnly": 0,
            "supportedAgents": [
                {
                    "agentFullname": "General Logistics Systems Czech Republic s.r.o.",
                    "agentAbbr": "GLS",
                    "requiredArguments": []
                },
                {
                    "agentFullname": "TNT Express Worldwide spol. s r.o.",
                    "agentAbbr": "TNT",
                    "requiredArguments": []
                }
            ]
        },
        {
            "code": "authentication",
            "fullname": "Ověření totožnosti",
            "description": "Při doručení kurýr předá zásilku příjemci až po předložení originálu dokladu totožnosti – OP, ŘP nebo cestovní pas.",
            "isActive": 1,
            "isImplicitOnly": 0,
            "supportedAgents": [
                {
                    "agentFullname": "WEIDO CZ s.r.o.",
                    "agentAbbr": "IT",
                    "requiredArguments": {
                        "identifier": "note",
                        "name": "Poznámka",
                        "example": ""
                    }
                }
            ]
        }
    ]
}
```

Description of the response fields (RESPONSE)

Field	Data type	Value	Description
`data`	array		Seznam přepravců
↳ `code`	string		Abbreviation Additional services in Foxdeli (Abbrevation).Used in API as `extraService`.
↳ `fullname`	string		Full name of the Additional Service
↳ `description`	string		Brief description of additional services.
↳ `isActive` [1]	tinyint	**0, 1**	Indication whether the service is supported and can be used to create shipments.In case of termination of service support, the value will be **0**.
↳ `isImplicitOnly` [1]	tinyint	**0, 1**	Indication whether the service is created automatically based on the parameters of the shipment.You do not submit implicit services via API.Foxdeli is setting it automatically (such as cash on delivery `cod`).
↳ `supportedAgents`	array		Transporters who support the service.If the unsupported service is sent error will be returned.
↳ `agentFullname`	string		The entire carrier name.
↳ `agentAbbr`	string		The abbreviation of the carrier (abbrevation) used in Foxdeli as `agent`.
↳ `requiredArguments`	array		Mandatory parameters that must be sent if you use the service.Used in API as `extraServices.arguments`
↳ `identifier`	string		The identifier of the mandatory parameter.The identifier is the key and the assigned value is expected as string.Example of use is in the methoddeliveries/POST

↳ `example`	string	Example of possible values (such as phone number, etc..)

Information

[1] **1** - yes, **0** - no

list/delivery-states GET

Getting Foxdeli Track & Trace Status for Shipment tracking.

Example of HTTP request(REQUEST)

Endpoint otevřít ↗

```
GET: https://rest.foxdeli.com/v4/list/delivery-states
```

Successful response example(RESPONSE)

The example contains only the demonstrative part of the answer.

Ukázka těla

```
    "message": "Foxdeli list of extra services",
    "data": [
        {
            "stateCategory": [
                {
                    "key": "in_progress",
                    "code": 1,
                    "name": "Rozpracované",
                    "color": "#ffffff"
                },
                {
                    "key": "ready_to_send",
                    "code": 2,
                    "name": "K odeslání",
                    "color": "#ffc83c"
                }
            ]
        },
        {
            "stateSubcategory": [
                {
                    "key": "in_progress",
                    "code": "1.0",
                    "name": "Rozpracované"
                },
                {
                    "key": "ready_to_send",
                    "code": "2.0",
                    "name": "K odeslání"
                }
            ]
        },
        {
            "state": [
                {
                    "key": "in_progress",
                    "code": "1.0.0",
                    "name": "Rozpracované",
                    "description": "Nově vytvořená zásilka (manuálně nebo metodou deliveries/POST), která ještě není uzavřená."
                },
                {
                    "key": "ready_to_send",
                    "code": "2.0.0",
                    "name": "K odeslání",
                    "description": "Uzavřené zásilky (manuálně nebo metodou deliveries/PATCH), které prozatím nebyly předány
kurýrovi."
                }
            ]
        }
    ]
}
```

Response field description (RESPONSE)

Field	Data type	Values	Description
`data`	array		List of states sorted into containers according to the detail of the states.
↳ `stateCategory|stateSubcategory|state`	array		See the container description below.
↳ `key`	string		Key.Unique Text Representation of Shipment Status (lower case).
↳ `code`	int|float|null		Unique number of shipment status code.Is contained in response methoddeliveries/traces/GET
↳ `name`	string		Status name (human readable).
↳ `description`	string		Shipment status information.

Information on containers and detail

- **stateCategory** - Container with main categories of shipments
- **stateSubcategory** - Container with Sub-categories shipment states

list/ticket-formats GET

Obtaining supported formats for printing shipping labels.

For more information, see the method description deliveries/traces/GET

Example HTTP request (REQUEST)

Endpoint otevřít

```
GET: https://rest.foxdeli.com/v4/list/zpl-tickets
```

Example of successful response (RESPONSE)

The example contains only a demonstrative part of the answer.

Ukázka těla

```
{
    "code": 200,
    "status": "success",
    "message": "Foxdeli list of ZPL tickets formats.",
    "data": [
        {
            "agentAbbr": "ČP",
            "size": "10x15",
            "dpi": "300",
            "printOrigin": "foxdeli",
            "orientation": "portrait"
        },
        {
            "agentAbbr": "GLS",
            "size": "10x5",
            "dpi": "300",
            "printOrigin": "foxdeli",
            "orientation": "letter"
        },
        {
            "agentAbbr": "GLS",
            "size": "10x15",
            "dpi": "300",
            "printOrigin": "foxdeli",
            "orientation": "letter"
        }
    ]
}
```

Response field description (RESPONSE)

Field	Data type	Field	Description
`data`	array		List all ZPL formats of the given carrier.
↳ `agentAbbr`	string	**GLS, PPL,...**	Carrier abbreviation
↳ `size`	string	**10x15, 10x5, 10x17,...**	Dimensions of the carrier label in cm.
↳ `code`	dpi	**300, 203**	Supported DPI.
↳ `name`	printOrigin	**foxdeli, agent**	Origin of label generation.In case of generation on the carrier (agent), you need to expect a longer time limit.
↳ `orientation`	string	**portrait, landscape**	Orientation of label.Information is suitable for preparing for automated printing.
↳ `isAgentDefault`		**1, 0**	**1** - The default label format of the carrier that will be returned in reply if there is no specific size requested.