

Getting started!

This documentation provides all the basic information you need to start using the “Lithuania post” API. It covers not only how to find streets, building, localities, addresses and buildings in Lithuania but as well how to create a shipping item, how to manage all the process of sending it from start to the end by calling the courier and creating required documents for the shipment.

We offer a TEST environment of “Lithuania Post” which mimics the PRODUCTION environment, so developers who integrate the public “Lithuania post” API to their application can experiment and test the product without any consequences.

Note: At this step you already have to have authentication to the TEST environment. We will provide you with the authentication to the PRODUCTION environment when you will be done with the integration in the TEST environment and all your needs have been reached ensuring that everything is working as it should.

All the communication with the API must be authenticated. Only authenticated request will be authorized. See **Authorization** page for detailed information about used security.

One of the most important function of this API is to let the user create, manage and send an item by creating it, adding required documents, calling the courier and tracking it. The processes of doing so are visualized bellow. As well find all the needed API methods and data structures for creating a request for the API.



Note: As well we are giving you a base requests collection for POSTMAN, which will help you with trying out the test environment of the API. All the needed actions to do are:

1. Use(POST): Authorization(Without client secret) script. In the **params** tab add your credentials.
2. You will get a response which will contain access token. Add it to your collection, select: View more actions -> Edit ->Authorization->Token

Contents Table

Getting started!	1
1. Shipping item creation and management process	6
1.1. Shipping item creation process	6
1.2. Shipping item additional services management	7
1.3. Delete shipping item	8
1.4. Get shipping item details	9
1.5. Get details for a list of shipping items	9
1.6. Get shipping item documents	10
1.7. Get courier manifest	10
1.8. Call a courier	11
2. List of API methods of the Lithuanian Post	12
2.1. findCountries (GET) method	15
2.2. getCountry (GET) method	15
2.3. findLocalities (GET) method	16
2.4. getLocality (GET) method	16
2.5. findLocalityPostalCodes (GET) method	17
2.6. findStreets (GET) method	17
2.7. getStreet (GET) method	18
2.8. findBuildings (GET) method	18
2.9. getBuilding (GET) method	19
2.10. findStreetPostalCodes (GET) method	19
2.11. findCountryPostalCodes (GET) method	20
2.12. getCountryPostalCodePostOffice (GET) method	20
2.13. findPostBoxes (GET) method	21
2.14. findPostOffices (GET) method	21
2.15. findPostOfficePostBoxes (GET) method	22
2.16. getDefaultValues (GET) method	22
2.17. findTerminals (GET) method	23
2.18. verify (POST) method	23
2.19. createShippingItem (POST) method	23
2.20. getShippingItems (GET) method	24
2.21. getShippingItem (GET) method	25
2.22. updateShippingItem (PUT) method	25
2.23. deleteShippingItem (DELETE) method	26
2.24. initiateShipping (POST) method	26
2.25. getShippingItemTemplates (OPTIONS) method	27
2.26. template (OPTIONS) method	27
2.27. getShippingItemOptions (OPTIONS) method	27

2.28.	callCourier (POST) method	28
2.29.	getSticker (GET) method	28
2.30.	getManifest (GET) method	29
2.31.	getManifestByBarcode (GET) method	29
2.32.	getDocumentCN22Report (GET) method	29
2.33.	getDocumentCN23Report (GET) method	30
2.34.	getTrackingBarcode (GET) method	30
2.35.	getTrackingItem (GET) method	31
3.	Global API data type structures and lists	32
3.1.	CountryType	32
3.2.	LocalityType	32
3.3.	StreetType	32
3.5.	BuildingType	33
3.6.	PostalCodeType	33
3.7.	PostOfficeType	33
3.8.	PostBoxType	33
3.9.	PostBoxRangeType	34
3.10.	TemplateType	34
3.11.	CountryType	34
3.12.	TerminalType	35
3.13.	SizeConst	35
3.14.	DocumentsType	35
3.15.	SizeConst	35
3.16.	ParcelTypeConstLPItems	36
3.17.	ParcelTypeConstLPEXPRESSItems	36
3.18.	StickerPrintLayoutEnum	36
3.19.	CN23failureInstructions	36
3.20.	TrackingItemStatus	36
3.21.	ReceiverType	39
3.22.	SenderType	39
3.23.	PostOfficeAddressType	39
3.24.	PostOfficeAddressBoxType	40
3.25.	LithuanianAddressType	40
3.26.	NOTLithuanianAddressType	40
3.27.	AddressRequestType	41
3.28.	ShippingItemType	41
3.29.	TemplateType	42
3.30.	AdditionalServicesType	42
3.31.	AdditionalServicesPropertiesType	42

3.32.	NoteProperties	43
3.33.	AmountProperties	43
3.34.	VariantProperties	43
3.35.	PriceType	43
3.36.	ShippingDocumentsType	44
3.37.	CN22FormType	44
3.38.	CN23FormType	44
3.39.	CNPartsType	45
3.40.	ShippingItemLabelType	45
3.41.	DocumentResponseType	45
3.42.	ManifestResponseType	45
4.	Template classification	46
5.	CN declarations	48
5.1.	Schema of required declarations for each shipping item template.....	48
5.2.	JSON request example of CN forms	49
5.3.	Troubleshooting of CN forms	49
6.	Additional Services	50
6.1.	Lithuanian Post and LPEXPRESS item type insurance differences	50
6.1.1.	LPEXPRESS type items	50
6.1.2.	Lithuanian Post type items.....	50
6.2.	Lithuanian Post item type additional services	51
6.2.3.	Priority shipping item	51
6.2.4.	Registered shipping item.....	51
6.2.5.	Save item for 10 days	51
6.2.6.	Cash on devlivery	51
6.3.	LPEXPRESS item type additional services.....	51
6.3.1.	Receiver pays the delivery fee	51
6.3.2.	Delivery on demanded time	51
6.3.3.	Parcel delivered with cover letter	51
6.3.4.	Print the sticker at the terminal	51
6.3.5.	Sticker will be delivered by the courier	52
6.3.6.	Cash on devlivery	52
7.	Security.....	52
7.1.	Authorization.....	52
7.2.	Addresses of method access points	53
7.3.	Authorization.....	55
7.4.	Limitation to the number of access point requests	56
7.5.	Potential errors	56
8.4.1.	Error codes	56

8.4.2.	VerificationError	56
8.4.3.	FieldValidationError	56
8.4.4.	ValueValidationError	57
8.4.5.	Error.....	57
7.6.	Request examples	57
8.5.1.	Example of create shipping item element	57
8.5.2.	Shipping item field validation error	59
8.5.3.	Shipping item field value validation error	59
8.5.4.	Initiate shipping item request	60
8.5.5.	Get stickers request	60
8.5.6.	Sending a priority shipping item	61
8.5.7.	Sending a registered shipping item	62
8.5.8.	Sending item with an unstructured receivers address	63
7.6.5.	Sending item with a structured receivers address	64

1. Shipping item creation and management process

1.1. Shipping item creation process

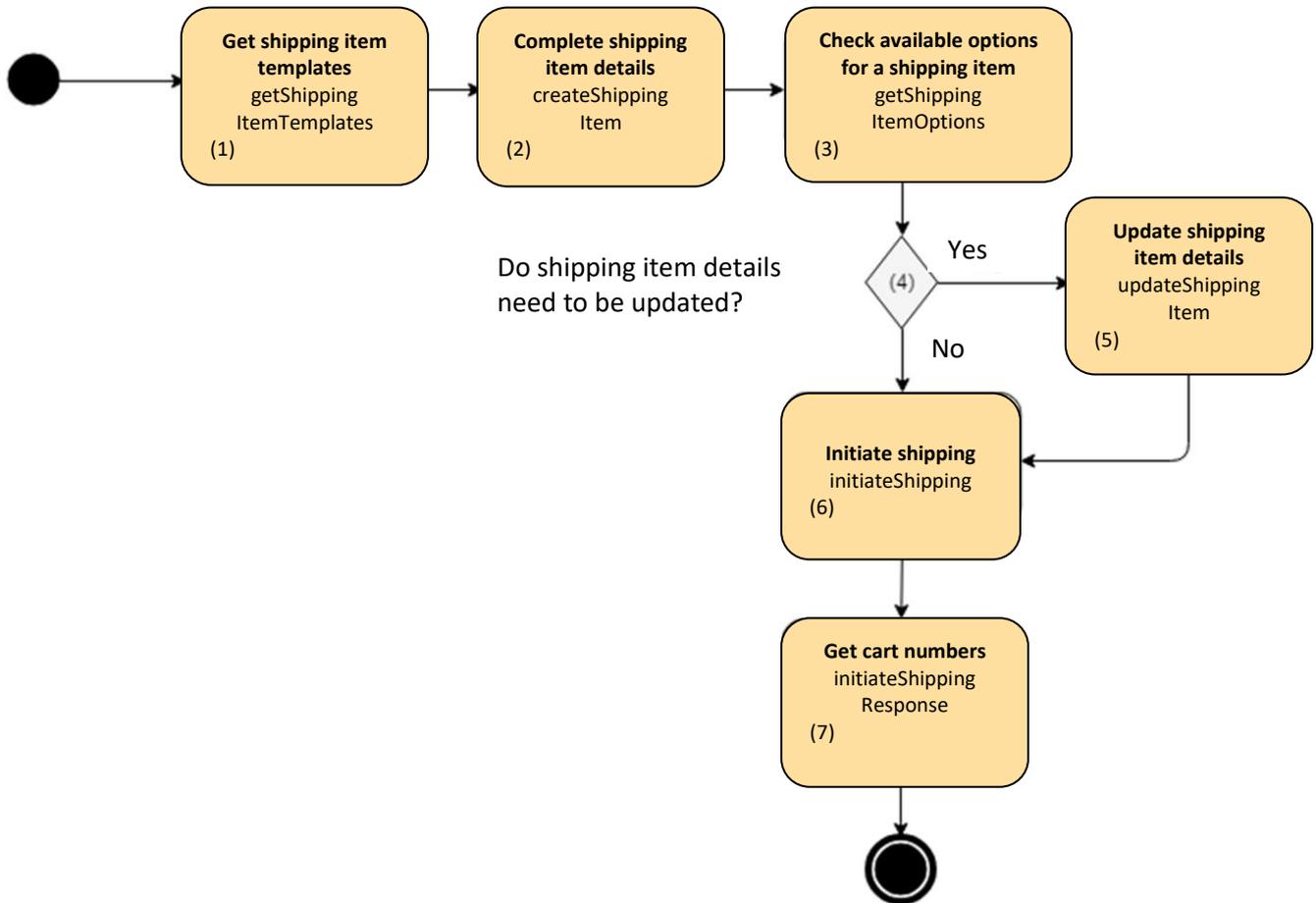


Figure 1. Shipping item creation and management process diagram

Table 1. Shipping item creation and management process description

No	Process	Description
1.	Get shipping item templates	To create a new shipping item, it is necessary to specify the default shipping item attributes (parameters) first. For this purpose, the system uses shipping item templates, i.e. sets of default parameters which are used to create a shipping item. A list of shipping item templates and their description may be obtained using the getShippingItemTemplates method. Note: the method is called whenever appropriate (when it is necessary to update the templates), but not every time to initiate a shipping item to be created.
2.	Create shipping item	To create a shipping item, it is necessary to enter the shipping item details by calling the createShippingItem method. To create a shipping item, it is necessary to specify the shipping item template number and the recipient's details. Note: if the sender's address is not provided, the system will automatically detect the sender's address using the account details. This method may have (not mandatory) the additional services regarding the shipping item and declaration details entered.

3.	Check available options for a shipping item	When a shipping item is created, it is possible to check the available options for it. The getShippingItemOptions method presents the available countries of the shipping item recipient, a list of additional services, the required declarations to be completed and the shipping item weight and size ranges. Note: the method is called whenever appropriate (when it is necessary to update the available shipping item properties). If the shipping item parameters are known and do not change, this step is not mandatory for a shipping item to be created and dispatched.
5.	Update shipping item details	When a shipping item is created, shipping item details may be updated, where necessary, using updateShippingItem . Note: the method is called whenever appropriate.
6.	Initiate shipping	When a shipping item or a list of shipping items is/are created, initiate shipping. The call of the initiateShipping method and specification of the shipping item identifiers is followed by a confirmation of the shipping item dispatch order.
7.	Receive cart identification numbers	When a shipping item order is successfully confirmed, the call of the initiateShipping returns cart identifiers.

1.2. Shipping item additional services management

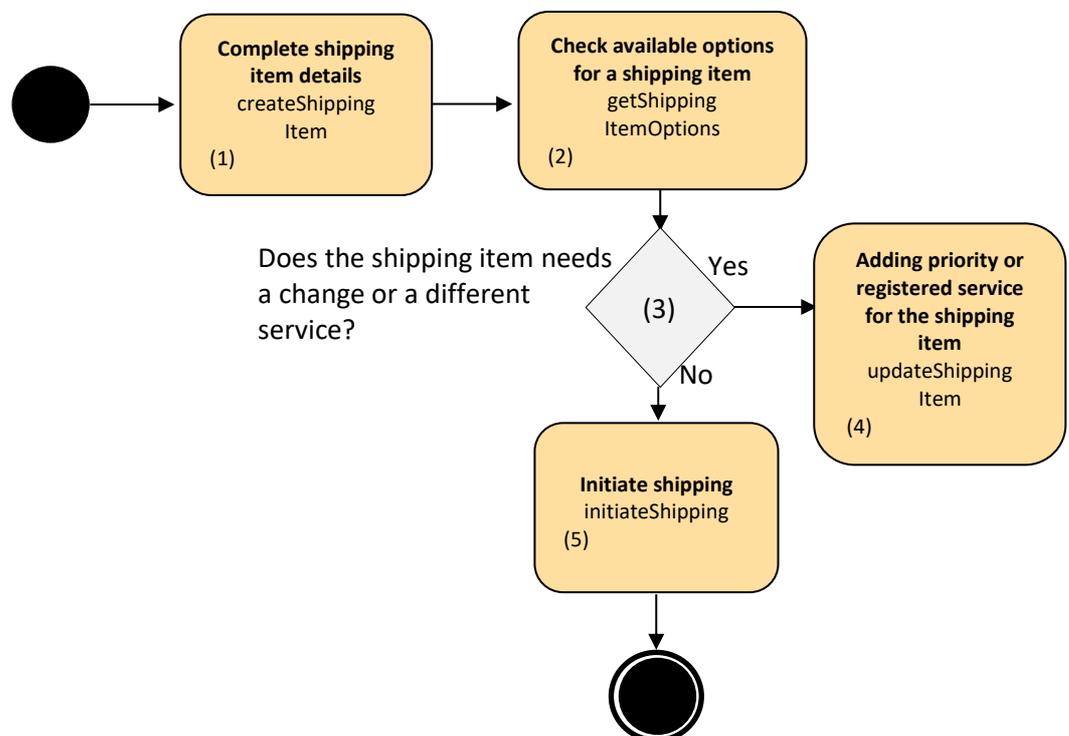


Figure 2. Shipping item additional services management process diagram

Table 2. Shipping item additional services management process description

No	Process	Description
1.	Create shipping item	To create a shipping item, it is necessary to enter the shipping item details by calling the <i>createShippingItem</i> method. To create a shipping item, it is necessary to specify the shipping item template number and the recipient's details. It is possible to add the additional services in the beginning of creating the item. If the required additional service is registered service. The form requires to input both additional services (priority and registered). Note: if the sender's address is not provided, the system will automatically detect the sender's address using the account details. This method may have (not mandatory) the additional services regarding the shipping item and declaration details entered.
2.	Check available options for a shipping item	When a shipping item is created, it is possible to check the available options for it. The <i>getShippingItemOptions</i> method presents the available countries of the shipping item recipient, a list of additional services, the required declarations to be completed and the shipping item weight and size ranges. Note: the method is called whenever appropriate (when it is necessary to update the available shipping item properties). If the shipping item parameters are known and do not change, this step is not mandatory for a shipping item to be created and dispatched.
3.	Does the shipping item needs a change or a different service?	When a shipping item is created, shipping item details may be updated, where necessary, using <i>updateShippingItem</i> . Note: the method is called whenever appropriate.
4.	Adding registered or priority service for a shipping item	In available options it will be seen that there is additional services: priority, which gives the item an additional service of priority and reaches the customer faster, and registered, which gives the item an additional service of registered item, and the item will be presented to the receiver directly. If the required additional service is registered service. The form requires to input both additional services (priority and registered). For priority form requires only priority . Using <i>updateShippingItem</i> adding an id of the additionalServices from the response of <i>getShippingItemOptions</i> . Note: these actions are available only before initiating the item, as well it is possible to add the services while creating an item.item.
5.	Initiate shipping	When a shipping item or a list of shipping items is/are created, initiate shipping. The call of the <i>initiateShipping</i> method and specification of the shipping item identifiers is followed by a confirmation of the shipping item dispatch order. Note: After initiating the shipping item or items it is impossible to add any services or update the items/items.

1.3. Delete shipping item

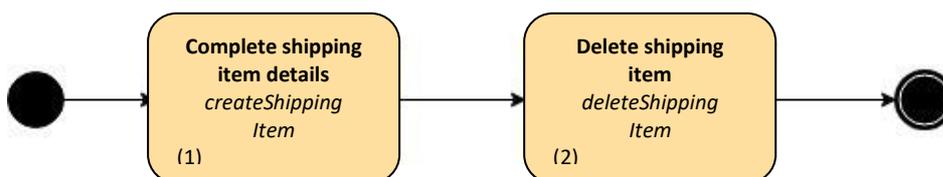


Figure 3. Diagram for deletion of a shipping item in progress

Table 3. Description of shipping item deletion

No	Process	Description
1.	Create shipping item	See Table 1– Shipping item creation and management process description
2.	Delete shipping item	A shipping item in progress may be deleted by calling the <i>deleteShippingItem ()</i> method. Note: if the shipping item has been dispatched, it cannot be deleted.

1.4. Get shipping item details

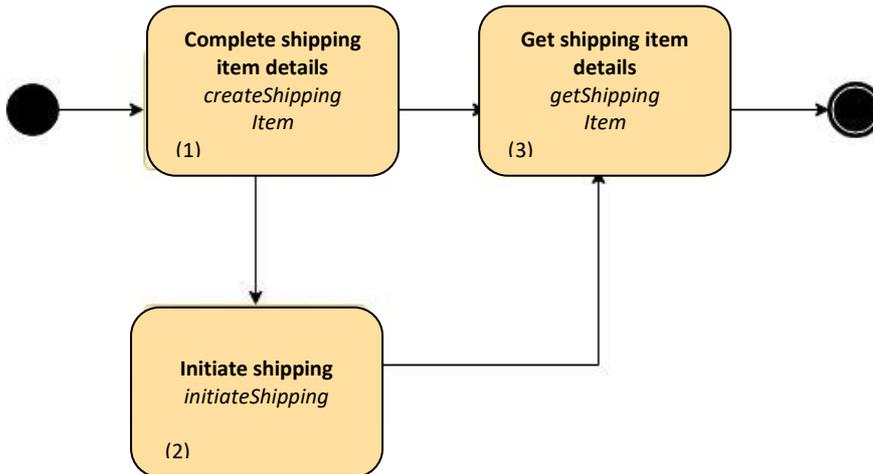


Figure 4. Diagram for shipping item

Table 4. Description of getting shipping item details

No	Process	Description
1.	Create shipping item	See Table 1– Shipping item creation and management process diagram
2.	Initiate shipping	See Table 1– Shipping item creation and management process diagram
3.	Get shipping item details	The details (e.g. status) of both a shipping item in progress and a dispatched shipping item may be received by calling the <i>getShippingItem</i> method. Note: certain shipping item details (e.g. barcode) may be obtained only for dispatched shipping items.

1.5. Get details for a list of shipping items

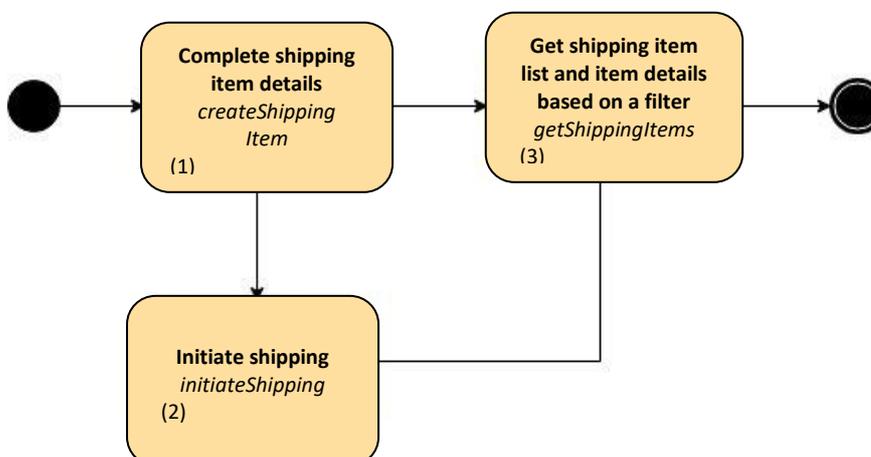


Figure 5. Diagram for getting details of a list of shipping items

Table 5. Description of getting details for a list of shipping items

No	Process	Description
1.	Create shipping item	See Table 1– Shipping item creation and management process description
2.	Initiate shipping	See Table 1– Shipping item creation and management process description
3.	Get a list of shipping items and item details based on a filter	The details (e.g. status) of both the shipping items in progress and the dispatched shipping items may be received by calling the getShippingItems method. The following filters may be used to get a list of shipping items <i>limit, offset, createdAfter, createdBefore</i> .

1.6. Get shipping item documents

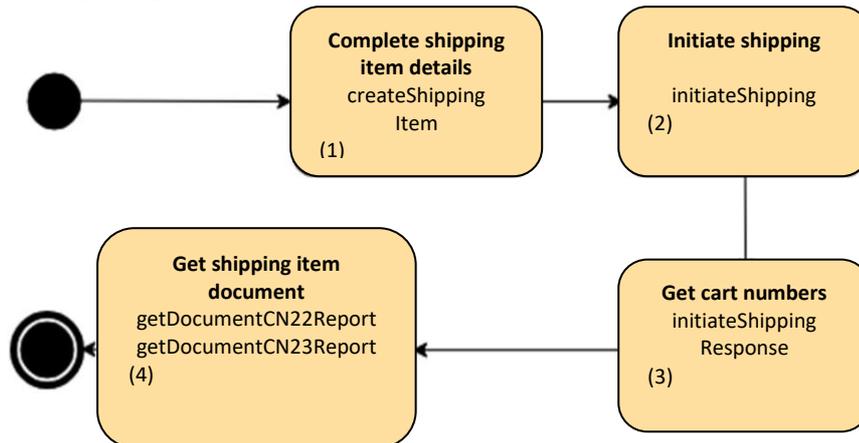


Figure 6. Diagram for getting a document

Table 6. Description of getting document details

No	Process	Description
1.	Create shipping item	See Table 1– Shipping item creation and management process description
2.	Initiate shipping	See Table 1– Shipping item creation and management process description
3.	Get cart numbers	When a shipping item or a list of shipping items is/are created, initiate shipping. The call of the initiateShipping method and specification of the shipping item identifiers is followed by a confirmation of the shipping item dispatch order. The call of the initiateShipping method returns the identifiers of created carts.
4.	Get shipping item document	When a shipping item is created, the required documents may be obtained, where necessary, using one of the available methods: getDocumentCN22Report, getDocumentCN23Report . Note: Documents are generated only for the successfully dispatched shipping items.

1.7. Get courier manifest

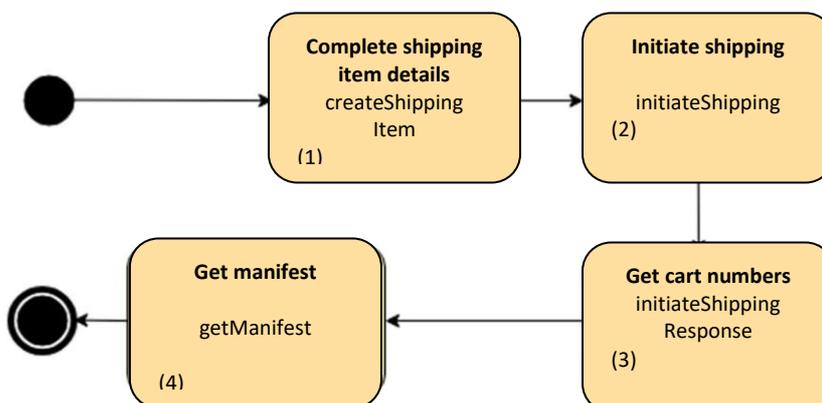


Figure 7. Get courier manifest process diagram

Table 7. Description of the get courier manifest process

No	Process	Description
1.	Create shipping item	See Table 1– Shipping item creation and management process description
2.	Initiate shipping	See Table 1– Shipping item creation and management process description
3.	Get cart numbers	When a shipping item or a list of shipping items is/are created, initiate shipping. The call of the <i>initiateShipping</i> method and specification of the shipping item identifiers is followed by a confirmation of the shipping item dispatch order. The call of the <i>initiateShipping</i> method returns the identifiers of created carts.
4.	Get courier manifests	The use of the <i>getManifest</i> method returns a courier manifest for the shipping item cart. Note: A manifest may only be obtained for initiated and dispatched shipping items. A shipping item manifest is automatically generated for every shipping item unless your contract provides for a possibility to call a courier.

1.8. Call a courier

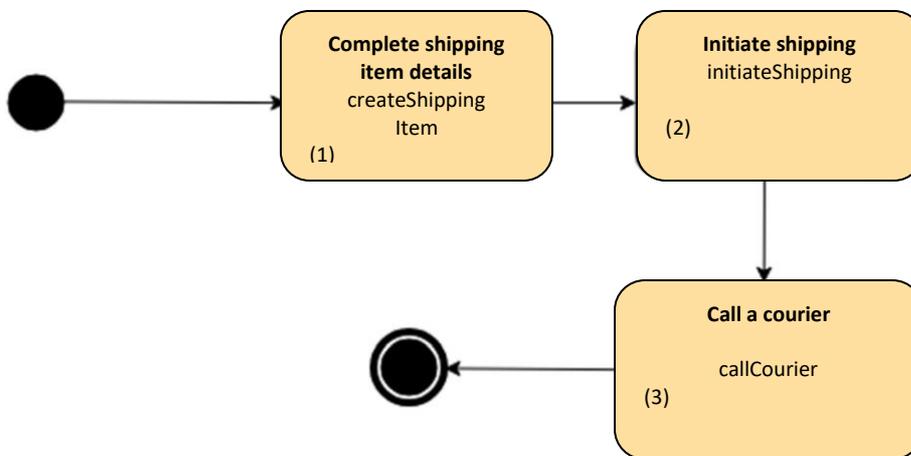


Figure 8. Diagram for calling a courier

Table 8. Description of the call a courier process

No	Process	Description
1.	Create shipping item	See Table 1– Shipping item creation and management process description
2.	Initiate shipping	See Table 1– Shipping item creation and management process description
3.	Call a courier	Initiates the call of a courier. (See 2.28 callCourier (POST) method)

2. List of API methods of the Lithuanian Post

To meet the needs of the business customers of the Lithuanian Post, 4 widgets have been created. Their summary has been provided in the table below. Detailed descriptions of the methods, including call and response parameters, are presented in Sections 2.1 – 2.26.

Table 9. List of API methods

API methods					
Category	Method type	Method name	Request	Response	Objective
Getting addresses	GET	<i>findCountries</i> (GET) method	[keyword, page, size]	[CountryType]	To find a list of available countries.
	GET	<i>getCountry</i> (GET) method	[countryId]	[CountryType]	To get a country.
	GET	<i>findLocalities</i> (GET) method	[countryId, keyword, page, postalCode, size]	[LocalityType]	To find a list of available localities.
	GET	<i>getLocality</i> (GET) method	[countryId, localityId]	[LocalityType]	To get a locality.
	GET	<i>findLocalityPostalCodes</i> (GET) method	[countryId, localityId, page, postalCode, size, street]	[PostalCodeType]	To find locality postal codes.
	GET	<i>findStreets</i> (GET) method	[countryId, keyword, localityId, page, postalCode, size]	[StreetType]	To find a list of available streets.
	GET	<i>getStreet</i> (GET) method	[countryId, localityId, streetId]	[StreetType]	To get a street.
	GET	<i>findBuildings</i> (GET) method	[countryId, keyword, localityId, page, postalCode, size, streetId]	[BuildingType]	To find buildings.
	GET	<i>getBuilding</i> (GET) method	[buildingId, countryId, localityId, streetId]	[BuildingType]	To get a building.
	GET	<i>findStreetPostalCodes</i> (GET) method	[countryId, keyword, localityId, page, postalCode, size, streetId]	[PostalCodeType]	To find street postal codes.
	GET	<i>findCountryPostalCodes</i> (GET) method	[countryId, keyword, page, size]	[PostalCodeType]	To find country's postal codes.
	GET	<i>getCountryPostalCodePostOffice</i> (GET) method	[countryId, postalCode]	[PostOfficeType]	To get country's post offices.
	GET	<i>findPostBoxes</i> (GET) method	[countryId, keyword, page, size]	[PostBoxType]	To find a list of PO boxes.
	GET	<i>findPostOffices</i> (GET) method	[countryId, keyword, page, size]	[PostOfficeType]	To find a list of post offices.
	GET	<i>findPostOfficePostBoxes</i> (GET) method	[countryId, keyword, postOfficeId]	[PostBoxType]	To find post office's PO boxes.
	GET	<i>getDefaultValues</i> (GET) method	-	[DefaultResponse]	To get a default country.
GET	<i>findTerminals</i> (GET) method	[page, size, templateId]	[TerminalType]	To find a list of available terminals.	

Address verification	POST	<i>verify</i> (POST) method	[addressRequestType]	[AddressRequestType]	To check the validity of address.
Shipping item creation and management	POST	<i>createShippingItem</i> (POST) method	[external id, sender, receiver, weight, price, additional services, CN22, CN23]	[shipping item]	To create a shipping item.
	PUT	<i>updateShippingItem</i> (PUT) method	[external id, sender, receiver, weight, price, additional services, CN22, CN23]	[shipping item]	To update and edit shipping item details.
	DELETE	<i>deleteShippingItem</i> (DELETE) method	[shipping item id]	[]	To delete a shipping item.
	POST	<i>initiateShipping</i> (POST) method	[shipping item ID's]	[cart IDs]	To initiate dispatch of shipping items. The response returns the identification numbers of shipping item carts.
	GET	<i>getShippingItems</i> (GET) method	[offset, limit, createdBefore, createdAfter]	[external id, shipping item id, sender, receiver, size, weight, price, shipping date, additional services, CN22, CN23]	To get details of dispatched shipping items.
	GET	<i>getShippingItem</i> (GET) method	[shipping item id]	[external id, shipping item id, sender, receiver, size, weight, price, shipping date, code, additional services, CN22, CN23]	To get details of a single shipping item.
	OPTIONS	<i>getShippingItem Templates</i> (OPTIONS) method	[]	[templates]	To get descriptions of shipping item templates.

	OPTIONS	<i>getShippingItemOptions (OPTIONS) method</i>	[shipping item id]	[countries, additional services, required documents, size, weight, terminals]	To get available options for a shipping item. The result returns available recipient's countries, additional services, required documents to be completed and size and weight ranges.
	POST	<i>callCourier (POST) method</i>	[item ID's]	[callCourier]	To call a courier for individual shipping items.
	OPTIONS	<i>template (OPTIONS) method</i>	[template id]	[countries, terminals]	To get template variables
Receive documents and stickers for completed shipping items	GET	<i>getSticker (GET) method</i>	[shipping item ID's, StickerPrintLayoutEnum]	[labels]	To get stickers for initiated shipping items.
	GET	<i>getManifest (GET) method</i>	[cart ID]	[document]	To get a manifest for a completed cart.
	GET	<i>getManifestByBarcode (GET) method</i>	[item barcode]	[manifest]	To get a manifest/manifests for completed shipping item/items based on barcode.
	GET	<i>getDocumentCn22Report (GET) method</i>	[cart ID]	[document]	To get a CN22 document for a cart.
	GET	<i>getDocumentCn23Report (GET) method</i>	[cart ID]	[document]	To get a CN23 document for a cart.

2.1. findCountries (GET) method

The method is designed to get a list of all countries available in the open application programming interface of the Lithuanian Post.

Table 10. Parameters for a call to the *findCountries* method.

Name	Type	Mandatory nature	Description
<i>keyword</i>	string	Optional, [0..1]	Keyword for the countries to be returned, which is a search criterion for the countries.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the countries to be returned, which determines the number of countries to be skipped in the list of selected countries before a list of countries is returned.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of countries to be returned, which determines the maximum number of countries to be generated on the list.

Table 11. Result of successful call of *findCountries*.

Name	Type	Mandatory nature	Description
<i>countries</i>	array CountryType (see Table 77)	Optional, [0..*]	The method returns a list of countries or an empty array if the country is not found based on the given parameters.

2.2. getCountry (GET) method

The method is designed to get a single country's details based on the country identifier.

Table 12. Parameters for a call to the *getCountry* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	The identifier of the country to be returned in the open application programming interface of the Lithuanian Post.

If the method is called successfully, it returns the country description or, where the country could not be found, an empty request.

Table 13. Result of a successful call to the *getCountry* method

Name	Type	Mandatory nature	Description
<i>country</i>	CountryType (see Table 77)	Optional, [0..*]	Returns a country. If the country is not found, it returns an empty array.

2.3. findLocalities (GET) method

The method is designed to get the localities of a country used in the open application programming interface of the Lithuanian Post.

Table 14. Parameters for a call to the *findLocalities* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>keyword</i>	string	Required, [0..1]	Keyword for locality search.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the localities to be returned, which determines the number of localities to be skipped in the list of selected localities before a list of localities is returned.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of localities to be returned, which determines the maximum number of localities present on the list.
<i>postalCode</i>	string	Optional, [0..1]	Postal code for locality search.

Table 15. Result of a successful call to the *findLocalities* method

Name	Type	Mandatory nature	Description
<i>localities</i>	LocalityType (see Table 78)	Optional, [0..*]	Locality search result. If the locality is not found, it returns an empty array.

2.4. getLocality (GET) method

The method is designed to get the details of a single locality.

Table 16. Parameters for a call to the *getLocality* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post for locality search.
<i>localityId</i>	int	Required, [1..1]	Locality identification number.

Table 17. Result of a successful call to the *getLocality* method

Name	Type	Mandatory nature	Description
<i>locality</i>	LocalityType (see Table 78)	Optional, [0..*]	Returned result of the locality search. This result may be empty if the system could not find a match.

2.5. findLocalityPostalCodes (GET) method

The method is designed to find the postal codes for a locality.

Table 18. Parameters for a call to the *findLocalityPostalCodes* method.

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post for postal code search.
<i>localityId</i>	int	Required, [1..1]	Locality identification number in the open application programming interface of the Lithuanian Post.
<i>street</i>	string	Optional, [0..1]	Keyword for street search.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the postal codes to be returned, which determines the number of postal codes to be skipped in the list of selected postal codes before a list of postal codes is returned.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of postal codes to be returned, which determines the maximum number of postal codes to be generated on the list.
<i>postalCode</i>	string	Optional, [0..1]	Postal code keyword used, in full or in part, for locality search.

Table 19. Result of a successful call to the *findLocalityPostalCodes* method

Name	Type	Mandatory nature	Description
<i>postalCodes</i>	PostalCodeType (see Table 81)	Optional, [0..*]	Result of postal code search. This result may not be returned, if the request could not find matches on the system.

2.6. findStreets (GET) method

The method is designed to find streets available for the locality.

Table 20. Parameters for a call to the *findStreets* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post for searching the streets of the locality.
<i>localityId</i>	int	Required, [1..1]	Locality identification number in the open application programming interface of the Lithuanian Post.
<i>keyword</i>	string	Optional, [0..1]	Keyword for street search.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the streets to be returned, which determines the number of streets to be skipped in the list of selected streets before a list of streets is returned.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of streets to be returned, which determines the maximum number of streets to be generated on the list.

<i>postalCode</i>	string	Optional, [0..1]	Postal code keyword used, in full or in part, for street search.
-------------------	--------	------------------	--

Table 21. Result of a successful call to the *findStreets* method

Name	Type	Mandatory nature	Description
<i>streets</i>	StreetType (See Table 79)	Optional, [0..*]	Street search result. This result may be empty if the system could not find any matches for the request.

2.7. *getStreet* (GET) method

The method is designed to get the details of a street.

Table 22. Parameters for a call to the *getStreet* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Searched street identifier in the open application programming interface of the Lithuanian Post.
<i>localityId</i>	int	Required, [1..1]	Locality identification number in the open application programming interface of the Lithuanian Post.
<i>streetId</i>	int	Required, [1..1]	Identified of the street to be returned in the open application programming interface of the Lithuanian Post.

Table 23. Result of a successful call to the *getStreet* method

Name	Type	Mandatory nature	Description
<i>street</i>	StreetType (See Table 79)	Optional, [0..1]	Search result of the returned street. This result may be empty if the system could not find any matches for the request.

2.8. *findBuildings* (GET) method

The method is designed to find the buildings on a street.

Table 24. Parameters for a call to the *findBuildings* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>localityId</i>	int	Required, [1..1]	Locality identification number in the open application programming interface of the Lithuanian Post.
<i>streetId</i>	int	Required, [1..1]	Street identifier in the open application programming interface of the Lithuanian Post.
<i>keyword</i>	string	Optional, [0..1]	Keyword for building search.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the buildings to be returned, which determines the number of buildings to be skipped in the list of selected buildings before a list of buildings is returned.

<i>size</i>	int	Optional, [0..1]	A limitation to the number of buildings to be returned, which determines the maximum number of buildings to be generated on the list.
<i>postalCode</i>	string	Optional, [0..1]	Postal code for building search.

Table 25. Result of a successful call to the *getBuilding* method

Name	Type	Mandatory nature	Description
<i>buildings</i>	BuildingType (see Table 80)	Optional, [0..*]	Building search result. This result may be empty if the system could not find any matches for the request.

2.9. *getBuilding* (GET) method

The method is designed to get the building on a street.

Table 26. Parameters for a call to the *getBuilding* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>localityId</i>	int	Required, [1..1]	Locality identification number in the open application programming interface of the Lithuanian Post.
<i>streetId</i>	int	Required, [1..1]	Street identifier in the open application programming interface of the Lithuanian Post.
<i>buildingId</i>	int	Required, [1..1]	Building identifier in the open application programming interface of the Lithuanian Post.

Table 27. Result of a successful call to the *getBuilding* method

Name	Type	Mandatory nature	Description
<i>building</i>	BuildingType (see Table 80)	Optional, [0..1]	Search result for a returned building. This result may be empty if the system could not find any matches for the request.

2.10. *findStreetPostalCodes* (GET) method

The method is designed to get the postal codes for a street.

Table 28. Parameters for a call to the *findStreetPostalCodes* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>localityId</i>	int	Required, [0..1]	Locality identification number in the open application programming interface of the Lithuanian Post.
<i>streetId</i>	int	Required, [1..1]	Street identifier in the open application programming interface of the Lithuanian Post.

<i>page</i>	int	Optional, [0..1]	A parameter used for paging the postal codes to be returned, which determines the number of postal codes to be skipped in the list of selected postal codes before a list of postal codes is returned.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of postal codes to be returned, which determines the maximum number of postal codes to be generated on the list.
<i>postalCode</i>	string	Optional, [0..1]	Postal code for locality search.
<i>keyword</i>	String	Optional, [0..1]	Search keyword.

Table 29. Result of a successful call to the *findStreetPostalCodes* method

Name	Type	Mandatory nature	Description
<i>postalCodes</i>	PostalCodeType (see Table 81)	Optional, [0..*]	Result of postal code search. This result may be empty if the system could not find any matches for the request.

2.11. *findCountryPostalCodes* (GET) method

The method is designed to get the postal codes for a country.

Table 30. Parameters for a call to the *findCountryPostalCodes* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>keyword</i>	string	Optional, [0..1]	Search keyword.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the postal codes to be returned, which determines the number of postal codes to be skipped in the list of selected shipping items before a list of postal codes is returned.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of postal codes to be returned, which determines the maximum number of postal codes to be generated on the list.

Table 31. Result of a successful call to the *findCountryPostalCodes* method

Name	Type	Mandatory nature	Description
<i>postalCodes</i>	PostalCodeType (see Table 81)	Optional, [0..*]	Result of postal code search. This result may be empty if the system could not find any matches for the request.

2.12. *getCountryPostalCodePostOffice* (GET) method

The method is designed to get a post office based on the postal code.

Table 32. Parameters for a call to the *getCountryPostalCodePostOffice* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>postalCode</i>	string	Required, [1..1]	Postal code in the open application programming interface of the Lithuanian Post for the post office to be returned.

Table 33. Result of a successful call to the *getCountryPostalCodePostOffice* method

Name	Type	Mandatory nature	Description
<i>postOffice</i>	PostOfficeType (see Table 82)	Optional, [0..*]	Result of post office search. This result may be empty if the system could not find any matches for the request.

2.13. findPostBoxes (GET) method

The method is designed to find post boxes by country.

Table 34. Parameters for a call to the *findPostBoxes* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [0..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>keyword</i>	string	Optional, [0..1]	Search keyword.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the post boxes to be returned, which determines the number of post boxes to be skipped in the list of selected post boxes before a list of post boxes is returned.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of post boxes to be returned, which determines the maximum number of post boxes to be generated on the list.

Table 35. Result of a successful call to the *findPostBoxes* method

Name	Type	Mandatory nature	Description
<i>postBoxes</i>	PostBoxType (see Table 83)	Optional, [0..*]	Result of post box search. This result may be empty if the system could not find any matches for the request.

2.14. findPostOffices (GET) method

The method is designed to find post offices by country.

Table 36. Parameters for a call to the *findPostOffices* method

Name	Type	Mandatory nature	Description
------	------	------------------	-------------

<i>countryId</i>	int	Required, [0..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>keyword</i>	string	Optional, [0..1]	Search keyword.
<i>size</i>	int	Optional, [0..1]	A limitation to the number of post offices to be returned, which determines the maximum number of post offices to be generated on the list.
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the post offices to be returned, which determines the number of post offices to be skipped in the list of selected post offices before a list of post offices is returned.

Table 37. Result of a successful call to the *findPostOffices* method

Name	Type	Mandatory nature	Description
<i>postOffices</i>	array PostOfficeType (see Table 82)	Optional, [0..*]	Result of post office search. This result may be empty if the system could not find any matches for the request.

2.15. findPostOfficePostBoxes (GET) method

The method is designed to get the post boxes in the post offices.

Table 38. Parameters for a call to the *findPostOfficePostBoxes* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [0..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>postOfficeId</i>	int	Required, [0..1]	Locality identification number.
<i>keyword</i>	string	Optional, [0..1]	Search keyword.

Table 39. Result of a successful call to the *findPostOfficePostBoxes* method

Name	Type	Mandatory nature	Description
<i>countryId</i>	int	Required, [0..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>postOfficeId</i>	int	Required, [0..1]	Locality identification number.
<i>keyword</i>	string	Optional, [0..1]	Search keyword.

2.16. getDefaultValues (GET) method

The method is designed to get the default country value. A call to this method does not require any parameters.

Table 40. Result of a successful call to the *getDefaultValues* method

Name	Type	Mandatory nature	Description
------	------	------------------	-------------

<i>country</i>	CountryType (see Table 77)	Optional, [0..1]	Returns the default country value in the open application programming interface of the Lithuanian Post.
----------------	----------------------------	------------------	---

2.17. findTerminals (GET) method

The method is designed to find the available terminals by template identifier. Terminals automatically refreshes every 48 hours.

Table 41. Parameters for a call to the *findTerminals* method

Name	Type	Mandatory nature	Description
<i>page</i>	int	Optional, [0..1]	A parameter used for paging the terminals to be returned, which determines the number of terminals to be skipped in the list of selected terminals before a list of terminals is returned.
<i>size</i>	int	Required, [0..1]	A limitation to the number of terminals to be returned, which determines the maximum number of terminals to be generated on the list.
<i>templateId</i>	string	Optional, [0..1]	Template identifier, which requires a list of terminals.

Table 42. Result of a successful call to the *findTerminals* method

Name	Type	Mandatory nature	Description
<i>terminals</i>	TerminalType (see Table 87)	Optional, [0..1]	A list of terminals to be returned and available for template purposes. If the template is not specified, it returns all active terminals available in the open application programming interface of the Lithuanian Post.

2.18. verify (POST) method

The method is designed to verify the validity of address in the open application programming interface of the Lithuanian Post. The method checks whether the postal code is known in the default country in the open application programming interface of the Lithuanian Post. In case of redirection to other countries, it validates the address structure.

Table 43. Parameters for a call to the *verify* method

Name	Type	Mandatory nature	Description
<i>address</i>	AddressRequestType (see Table 104)	Required, [1..1]	Parameters of address to be validated.

Table 44. Result of a successful call to the *verify* method

Name	Type	Mandatory nature	Description
<i>address</i>	AddressRequestType (see Table 104)	Optional, [0..1]	Returns the same address structure if the address is valid.

2.19. createShippingItem (POST) method

The method is designed to transfer the details of a new shipping item into the open application programming interface of the Lithuanian Post, thereby creating a shipping item. Pending dispatch, the shipping item belongs to the list of shipping items in progress and may have its details edited or updated and it may be deleted. A shipping item template must

be used to create a shipping item. It specifies the location which the shipping item must be collected from and delivered to (home, LP EXPRESS terminal, LP post office), the shipping item type and weight ranges.

Table 45. Parameters for a call to the *createShippingItem* method

Name	Type	Mandatory nature	Description
<i>externalId</i>	string	Optional, [0..1]	Unique identifier assigned to the shipping item on the user's system.
<i>template</i>	string	Required, [1..1]	Unique identifier of a shipping item template.
<i>receiver</i>	ReceiverType (see Table 96)	Required, [1..1]	Shipping item recipient's details
<i>sender</i>	SenderType (see Table 97)	Optional, [0..1]	Shipping item sender's details. If not provided, the default details available in the user profile are given.
<i>partCount</i>	int	Required, [1..1]	Number of parts of the shipping item.
<i>weight</i>	number (double)	Optional, [0..1]	Shipping item weight (g). Required if the selected shipping item template has default shipping item weights.
<i>additionalServices</i>	AdditionalServicesType (see Table 107)	Optional, [0..1]	Additional services for the shipping item.
<i>documents</i>	ShippingDocumentsType (see Table 113)	Optional, [0..1]	A list of declarations to accompany the shipping item.

If the API method *createShippingItem* is called successfully, the system returns the *createShippingItem* result.

Table 46. Result returned for the *createShippingItem* method

Name	Type	Mandatory nature	Description
<i>shippingItem</i>	ShippingItemType (see Table 105)	Required, [1..1]	Shipping item data structure

2.20. *getShippingItems* (GET) method

The method is designed to get a list of shipping items in progress and dispatched shipping items including full details of the shipping item. The method call parameters are used as filters to sort out the shipping items for which details are sought. Available filters: offset, count, createdBefore, createdAfter.

Table 47. Parameters for a call to the *getShippingItems* method

Name	Type	Mandatory nature	Description
<i>offset</i>	string	Optional, [0..1]	A parameter used for paging the shipping items to be returned, which determines the number of shipping items to be skipped in the list of selected shipping items before a list of shipping items is returned.
<i>count</i>	int	Optional, [0..1]	A limitation to the number of shipping items to be returned, which determines the maximum number of shipping items to be generated on the list.

<i>createdBefore</i>	int	Optional, [0..1]	A limitation to the creation time of the shipping items to be returned, which instructs the selection of shipping items created before a specific time period. The parameter must have a time stamp entered in seconds in Unix format (https://www.unixtimestamp.com/).
<i>createdAfter</i>	int	Optional, [0..1]	A limitation to the creation time of the shipping items to be returned, which instructs the selection of shipping items created after a specific time period. The parameter must have a time stamp entered in seconds in Unix format (https://www.unixtimestamp.com/).

If the API service method **getShippingItems** is called successfully, the system returns the **getShippingItems** result.

Table 48. Result returned for the **getShippingItems** method

Name	Type	Mandatory nature	Description
<i>shippingItem</i>	ShippingItemType (see Table 105)	Optional, [0..*]	A list of selected shipping items

2.21. getShippingItem (GET) method

The method is designed to get the details of a specific shipping item whether already created or dispatched. If the unique identifier is given for a shipping item, depending on the shipping item status, the method presents full details of the shipping item.

Table 49. Parameters for a call to the **getShippingItem** method

Name	Type	Mandatory nature	Description
<i>itemId</i>	string	Required, [1..1]	Unique identifier of the shipping item on “Mano siuntos” system.

If the API service method **getShippingItem** is called successfully, the system returns the **getShippingItem** result.

Table 50. Result returned for the **getShippingItem** method

Name	Type	Mandatory nature	Description
<i>shippingItem</i>	ShippingItemType (see Table 105)	Required, [1..1]	Shipping item data structure (see 3.10 ShippingItemType)

2.22. updateShippingItem (PUT) method

The method is designed to edit and update a shipping item in progress. All details provided in this method are re-written. Therefore, to update the list of additional services, for example, it is necessary to provide all details of the shipping item. To make it easier to update the shipping item details, it is advisable to call the **getShippingItem (GET)** first and obtain all up-to-date details of the shipping item thereby with the updates left to follow.

Table 51. Parameters for a call to the **updateShippingItem** method

Name	Type	Mandatory nature	Description
<i>itemId</i>	string	Optional, [0..1]	Unique identifier of the shipping item.
<i>receiver</i>	ReceiverType (see Table 96)	Required, [1..1]	Shipping item recipient’s details (see 3.4 ReceiverType).

<i>sender</i>	SenderType (see Table 97)	Optional, [0..1]	Shipping item sender's details (see 3.5 SenderType). If not provided, the default details available in the user profile are given.
<i>partCount</i>	int	Optional, [0..1]	Number of parts of the shipping item.
<i>weight</i>	int	Optional, [0..1]	Shipping item weight (g).
<i>additionalServices</i>	AdditionalServicesType (see Table 107)	Required, [0..1]	Additional services for the shipping item.
<i>ShippingDocuments</i>	ShippingDocumentsType (see Table 113)	Optional, [0..1]	A list of declarations to accompany the shipping item.

If the API service method **updateShippingItem** is called successfully, the system returns **updateShippingItem** result.

Table 52. Result returned for the **updateShippingItem** method

Name	Type	Mandatory nature	Description
<i>shippingItem</i>	ShippingItemType (see Table 105)	Required, [1..1]	Details of an updated or edited shipping item.

2.23. deleteShippingItem (DELETE) method

The method is designed to delete a shipping item in progress or being edited from the system. If a unique identifier of the shipping item is provided, the method checks that the shipping item has not been dispatched and, if so, deletes it.

Table 53. Parameters for a call to the **deleteShippingItem** method

Name	Type	Mandatory nature	Description
<i>itemId</i>	string	Required, [1..1]	Unique identifier of the shipping item on "Mano siuntos" system.

If the API service method **deleteShippingItem** is called successfully, the system returns the **deleteShippingItem** result, i.e. a success message (boolean value).

2.24. initiateShipping (POST) method

The method is designed to confirm the details of created shipping items and initiate their shipment. If the list of shipping item identifiers is provided, the method checks whether the relevant shipping items have all required details and creates an order for the shipping items. If the order is confirmed successfully, the method returns the identification numbers for the created carts. This operation has shipping item identifiers provided by the POST request.

Table 54. Parameters for a call to the **initiateShipping** method

Name	Type	Mandatory nature	Description
<i>itemIds</i>	array, string	Required, [1..1]	A list of shipping items sought to be dispatched.

If the API service method **initiateShipping** is called successfully, the system returns the **initiateShipping** result.

Table 55. Result returned for the **initiateShipping** method

Name	Type	Mandatory nature	Description
<i>carts</i>	array, string	Required, [1..1]	Identification numbers for shipping item carts.

2.25. `getShippingItemTemplates (OPTIONS)` method

The method is designed to get detailed descriptions of shipping item templates including their unique identifiers on the system. To create a shipping item, it is necessary to specify the identifier for the shipping item template in the open application programming interface of the Lithuanian Post. Hence, this method may be used to select the correct template.

A call to this method does not require any parameters to be specified. If the API service method `getShippingItemTemplates` is called successfully, the system returns the `getShippingItemTemplates` result.

Table 56. Result returned for the `getShippingItemTemplates` method

Name	Type	Mandatory nature	Description
<code>template</code>	TemplateType (see Table 85)	Required, [1..*]	Shipping item template data structure.

2.26. `template (OPTIONS)` method

The method is designed to check the available options for a shipping item template. The response to the method provides information about the countries and/or list of terminals available for the template.

Table 57. Parameters for a call to the `template` method

Name	Type	Mandatory nature	Description
<code>template</code>	string	Required, [1..1]	Unique identifier of a shipping item template.

If the API service method `template` is called successfully, the system returns the `getShippingItemOptions` result.

Table 58. Result returned for the `template` method

Name	Type	Mandatory nature	Description
<code>countries</code>	CountryType(see Table 86)	Required, [1..*]	A list of countries where the shipping item concerned may be sent to.
<code>terminals</code>	TerminalType (see Table 87)	Required, [1..*]	If the shipping item originates from/ is destined for a terminal, it provides a list of terminals available to choose from.

2.27. `getShippingItemOptions (OPTIONS)` method

The method is designed to check the available options for an already created shipping item, which has not been dispatched yet. Based on the entered details (recipient'/sender's address), the system selects the available recipient's countries and provides a list of additional services and necessary declarations. If the shipping item to be dispatched is between terminals, it returns a list of available terminals. The response to the method also provides the permitted shipping item size and weight ranges. Having received the above details, the user may edit the shipping item accordingly by updating or changing its details to make the shipping item compatible for shipping purposes.

Table 59. Parameters for a call to the `getShippingItemOptions` method

Name	Type	Mandatory nature	Description
<code>itemId</code>	string	Required, [1..1]	Unique identifier of the shipping item.

If the API service method **getShippingItemOptions** is called successfully, the system returns the **getShippingItemOptions** result.

Table 60. Result returned for the **getShippingItemOptions** method

Name	Type	Mandatory nature	Description
<i>countries</i>	CountryType (see Table 86)	Required, [1..*]	A list of countries where the shipping item concerned may be sent to.
<i>itemId</i>	string	Required, [1..1]	Unique identifier of the shipping item.
<i>availableServices</i>	AdditionalServicesType (see Table 107)	Required, [1..*]	A list of additional services available for the selected shipping item.
<i>requiredDocuments</i>	DocumentsType (see Table 89)	Required, [1..*]	A list of required documents to send the shipping item abroad.
<i>weights</i>	WeightType (see Table 88)	Required, [1..*]	Permitted shipping item weight.
<i>terminals</i>	TerminalType (see Table 87)	Required, [1..*]	If the shipping item originates from/ is destined for a terminal, it provides a list of terminals available to choose from.

2.28. callCourier (POST) method

The method is designed to call a courier for an already created and initiated shipping item. Calling a courier is only available to customers that are allowed to call a courier manually on the basis of their contract.

Table 61. Parameters for a call to the **callCourier** method

Name	Type	Mandatory nature	Description
<i>itemId</i>	string	Required, [1..*]	Unique identifier of the shipping item.

Table 62. Result of a call to the **callCourier** method

Name	Type	Mandatory nature	Description
<i>barcode</i>	string	Required, [1..1]	Shipping item unique barcode
<i>manifestId</i>	string	Required, [1..1]	Shipping item unique manifest id.

2.29. getSticker (GET) method

The method designed to get stickers for initiated shipping items. This method may return an empty array if the stickers have not been generated for a shipping item. **Note:** Be sure that you are using an **itemId**.

Table 63. Parameters for a call to the **getSticker** method

Name	Type	Mandatory nature	Description
<i>itemId</i>	array string	Required, [1..*]	Unique list of identifiers of a shipping item.
<i>layout</i>	StickerPrintLayoutEnum (see Table 93)	Required, [1..1]	The format of shipping item sticker layout across the page.

Table 64. Result of a successful call to the *getSticker* method

Name	Type	Mandatory nature	Description
<i>labels</i>	array ShippingItemLabelType (see Table 117)	Optional, [0..*]	Shipping item sticker documents where the stickers have been generated in the open application programming interface of the Lithuanian Post.

2.30. *getManifest* (GET) method

The method is designed to get the courier manifests of initiated shipping items for a shipping item cart. This method may return an empty array if the courier manifest has not been generated for a shipping item.

Table 65. Parameters for a call to the *getManifest* method

Name	Type	Mandatory nature	Description
<i>cartId</i>	string	Required, [1..1]	Shipping item cart identifier.

Table 66. Result of a successful call to the *getManifest* method

Name	Type	Mandatory nature	Description
<i>manifest</i>	ManifestResponseType (see Table 119)	Optional, [0..1]	Document of the shipping item courier manifest.

2.31. *getManifestByBarcode* (GET) method

The method is designed to get a courier manifest for an initiated shipping item. This method returns one courier manifest where the courier has been called for the shipping item manually.

Table 67. Parameters for a call to the *getManifestByBarcode* method

Name	Type	Mandatory nature	Description
<i>barcode</i>	string	Required, [1..1]	Shipping item barcode identifier.

Table 68. Result of a successful call to the *getManifestByBarcode* method

Name	Type	Mandatory nature	Description
<i>manifest</i>	ManifestResponseType (see Table 119)	Optional, [0..1]	Document of the shipping item courier manifest.

2.32. *getDocumentCN22Report* (GET) method

The method is designed to get the CN22 documents for an initiated shipping item cart. **Note:** Be sure that you are using a **cartId**, **NOT** an **itemId**. **NOT** applicable for template 66&67

Table 69. Parameters for a call to the *getDocumentCN22Report* method

Name	Type	Mandatory nature	Description
<i>cartId</i>	string	Required, [1..1]	Shipping item cart identifier.

Table 70. Result of a successful call to the *getDocumentCN22Report* method

Name	Type	Mandatory nature	Description
<i>document</i>	DocumentReponseType (see Table 118)	Optional, [0..1]	Document generated from a CN 22 template.

2.33. *getDocumentCN23Report* (GET) method

The method is designed to get the generated CN23 documents for an initiated shipping item cart. **Note:** Be sure that you are using a **cartId**, **NOT** an **itemId**.

Table 71. Parameters for a call to the *getDocumentCN23Report* method

Name	Type	Mandatory nature	Description
cartId	string	Required, [1..1]	Shipping item cart identifier.

Table 72. Result of a successful call to the *getDocument23Report* method

Name	Type	Mandatory nature	Description
<i>document</i>	DocumentReponseType (see Table 118)	Optional, [0..1]	Document generated from a CN 23 template.

2.34. *getTrackingBarcode* (GET) method

The method gathers tracking information about shipping item by barcode.

Table 73. Parameters for a call to the *getTrackingBarcode* method

Name	Type	Mandatory nature	Description
<i>barcode</i>	string	Required, [1..1]	Shipping item unique barcode

If the API service method *getTrackingBarcode* is called successfully, the system returns the *getTrackingBarcode* result.

Table 74. Result returned for the *getTrackingItems* method

Name	Type	Mandatory nature	Description
<i>barcode</i>	TrackingItemStatus (see Table 95)	Required, [1..1]	Tracking shipments and checking shipment delivery status

2.35. getTrackingItem (GET) method

The method gathers tracking information about shipping item by item id.

Table 75. Parameters for a call to the *getTrackingItem* method

Name	Type	Mandatory nature	Description
<i>itemId</i>	string	Required, [1..1]	Unique identifier of the shipping item on “Mano siuntos” system.

If the API service method *getTrackingItem* is called successfully, the system returns the *getTrackingItem* result.

Table 76. Result returned for the *getTrackingItem* method

Name	Type	Mandatory nature	Description
<i>TrackingItem</i>	TrackingItemStatus (see Table 95)	Required, [1..1]	Tracking shipments and checking shipment delivery status

3. Global API data type structures and lists

This section provides the data structures used globally on the API and their detailed descriptions.

3.1. CountryType

Table 77. *CountryType* data structure

Name	Type	Mandatory nature	Description
<i>country</i>	string	Required, [1..1]	Country name.
<i>default</i>	boolean	Required, [1..1]	The value is set when the returned country is the default country on the system.
<i>id</i>	int	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.

3.2. LocalityType

Table 78. *LocalityType* data structure

Name	Type	Mandatory nature	Description
<i>country</i>	CountryType (see Table 77)	Optional, [0..1]	Country of the locality to be returned.
<i>district</i>	string	Required, [1..1]	District of the locality to be returned.
<i>id</i>	int	Required, [1..1]	Identifier of the locality to be returned in the open application programming interface of the Lithuanian Post.
<i>municipality</i>	string	Required, [1..1]	Municipality of the locality to be returned.
<i>locality</i>	string	Required, [1..1]	Name of the locality to be returned.

3.3. StreetType

Table 79. *StreetType* data structure

Name	Type	Mandatory nature	Description
<i>id</i>	int	Required, [1..1]	Street identifier in the open application programming interface of the Lithuanian Post.
<i>locality</i>	LocalityType (See Table 78)	Optional, [0..1]	Locality details in the open application programming interface of the Lithuanian Post.
<i>street</i>	string	Required, [1..1]	Street name in the open application programming interface of the Lithuanian Post.

3.5. BuildingType

Table 80. *BuildingType* data structure

Name	Type	Mandatory nature	Description
<i>building</i>	string	Required, [1..1]	Building identifier in the open application programming interface of the Lithuanian Post.
<i>id</i>	int	Required, [1..1]	Locality identification number in the open application programming interface of the Lithuanian Post.
<i>street</i>	StreetType (see Table 79)	Required, [1..1]	Street identifier in the open application programming interface of the Lithuanian Post.
<i>postalcode</i>	string	Required, [1..1]	Postal code of the building.

3.6. PostalCodeType

Table 81. *PostalCodeType* data structure

Name	Type	Mandatory nature	Description
<i>postalCode</i>	string	Required, [1..1]	Postal code.
<i>id</i>	int	Required, [1..1]	Postal code identifier in the open application programming interface of the Lithuanian Post.
<i>country</i>	CountryType (See Table 77)	Optional, [0..1]	Country of the postal code.

3.7. PostOfficeType

Table 82. *PostOfficeType* data structure

Name	Type	Mandatory nature	Description
<i>id</i>	int	Required, [1..1]	Post office identifier in the open application programming interface of the Lithuanian Post.
<i>locality</i>	LocalityType (see Table 78)	Optional, [0..1]	Postal locality.
<i>name</i>	string	Required, [1..1]	Post office name.
<i>postalcode</i>	string	Required, [1..1]	Postal code of the post office.

3.8. PostBoxType

Table 83. *PostBoxType* data structure

Name	Type	Mandatory nature	Description
<i>boxes</i>	array, PostBoxRangeType (see Table 84)	Optional, [0..*]	Result of post box search. This result may be empty if the system could not find any matches for the request.
<i>postOffice</i>	PostOfficeType (see Table 82)	Required, [1..1]	Post office details.

3.9. PostBoxRangeType

Table 84. *PostBoxRangeType* data structure

Name	Type	Mandatory nature	Description
<i>fromNumber</i>	Int	Required, [1..1]	Lower limit of the post box number range.
<i>toNumber</i>	int	Required, [1..1]	Upper limit of the post box number range.

3.10. TemplateType

Table 85. *TemplateType* shipping item template data structure

Name	Type	Mandatory nature	Description
<i>id</i>	int	Required, [1..1]	Unique identifier of a shipping item template.
<i>title</i>	string	Required, [1..1]	Shipping item template name.
<i>summary</i>	string	Required, [1..1]	Shipping item template summary.
<i>description</i>	string	Required, [1..1]	Shipping item template description.
<i>weight</i>	int	Optional, [0..1]	Shipping item weight in grams.
<i>size</i>	string, SizeConst (see Table 90)	Optional, [0..1]	Shipping item size.
<i>partCount</i>	int	Required, [1..1]	Number of parts of the shipping item.
<i>type</i>	string	Required, [1..1]	Shipping item type.
<i>sender.address.country</i>	string	Required, [1..1]	Shipping item sender's country code.
<i>receiver.address.country</i>	string	Required, [1..1]	Shipping item recipient's country code.
<i>receiver.address.isTerminalReceiverAddressRequired</i>	boolean	Optional, [1..0]	To be set if the template requires a terminal id to be entered at the recipient's address.
<i>additionalServices</i>	AdditionalServicesPropertiesType (see Error! Reference source not found.)	Required, [1..1]	Data structure for additional services of a shipping item.

3.11. CountryType

Table 86. *CountryType* data structure

Name	Type	Mandatory nature	Description
<i>code</i>	string	Required, [1..1]	ISO-3166-alpha2 country code in the open application programming interface of the Lithuanian Post.

<i>id</i>	string	Required, [1..1]	Country identifier in the open application programming interface of the Lithuanian Post.
<i>name</i>	string	Required, [1..1]	Country name in the open application programming interface of the Lithuanian Post.

3.12. TerminalType

Table 87. *TerminalType* data structure

Name	Type	Mandatory nature	Description
<i>address</i>	string	Required, [1..1]	Terminal address.
<i>boxes</i>	array string, SizeConst (see Table 90)	Required, [1..*]	Available box sizes for the terminal.
<i>city</i>	string	Required, [1..1]	City of the terminal.
<i>id</i>	string	Required, [1..1]	Terminal identifier in the open application programming interface of the Lithuanian Post.
<i>name</i>	string	Required, [1..1]	Terminal name.
<i>postalCode</i>	string	Required, [1..1]	Postal code of the terminal.

3.13. SizeConst

Table 88. *WeightType* data structure

Name	Type	Mandatory nature	Description
<i>weightMax</i>	number (double)	Optional, [1..1]	Maximum permitted weight of the shipping item.
<i>weightMin</i>	number (double)	Optional, [1..1]	Minimum permitted weight of the shipping item.

3.14. DocumentsType

Table 89. *DocumentsType* data structure

Name	Type	Mandatory nature	Description
cn22Document Required	boolean	Required, [1..1]	Attribute indicating whether the shipping item requires a CN22 declaration form.
cn23Document Required	boolean	Required, [1..1]	Attribute indicating whether the shipping item requires a CN23 declaration form.

3.15. SizeConst

Table 90. *SizeConst* types

Name	Description
XSmall	a very small shipping item. Magazines, documents and jewellery.
Small	a small shipping item. Small electronics, toys and toiletries
Medium	a medium shipping item. Gifts, shoes and clothes.
Large	a large shipping item. Bulky items and large appliances.

<i>XLarge</i>	a very large shipping item. Bulky items and large appliances.
---------------	---

3.16. ParcelTypeConstLPItems

Table 91. Lithuanian Post Parcel Types

Name	Description
Gift	a Gift
<i>Document</i>	a Document
<i>Sample</i>	a Sample of an item
<i>Sell</i>	Merchandise
<i>Return</i>	Goods to be returned
<i>Other</i>	Other. If "Other" is selected, the parcelTypeNote parameter including the item description is required.

3.17. ParcelTypeConstLPEXPRESSItems

Table 92. LPEXPRESS Parcel Types

Name	Description
Gift	a Gift
<i>Document</i>	a Document
<i>Sample</i>	a Sample of an item
<i>Sell</i>	Merchandise
<i>Return</i>	Goods to be returned

3.18. StickerPrintLayoutEnum

Table 93. Stickers print Layout

Name	Description
LAYOUT_MAX	the maximum number of stickers per sheet
LAYOUT_A4	one sticker on an A4 sheet
LAYOUT_10x15	letter-sized stickers

3.19. CN23failureInstructions

Table 94 CN23failureInstruction

Name	Description
RETURN_TO_SENDER_NON_PRIORITY	The parcel will be returned to the sender without priority. (Default)
RETURN_TO_SENDER_PRIORITY	The parcel will be returned to the sender with priority.
TREAT_AS_ABANDONED	Treat with the parcel as abandoned

3.20. TrackingItemStatus

Table 95. Tracking Item status

Name	Description
ACCEPTED	Accepted
BP_TERMINAL_REQUEST_ACCEPTED	LP express terminal request accepted

BP_TERMINAL_REQUEST_FAILED	LP express terminal request failed
BP_TERMINAL_REQUEST_REJECTEDACCEPTED	LP express terminal request rejected
BP_TERMINAL_REQUEST_SENT	BP terminal request submitted
CANCELLED	Canceled
DA_ACCEPTED_LP	The parcel was taken from the Lithuanian Post Office
DA_ACCEPTED	Parcel accepted from sender
DA_DELIVERED_LP	The parcel was delivered to the Lithuanian Post Office
DA_DELIVERED	The shipment has been delivered to the receiver
DA_DELIVERY_FAILED	Delivery failed
DA_EXPORTED	Item shipped from Lithuania
DA_PASSED_FOR_DELIVERY	The parcel has been handed over to the courier for delivery
DA_RETURNED	Shipment returned
DA_RETURNING	The parcel is being returned
DEAD	The itemt was delivered for destruction
DELIVERED	Delivered
DEP_RECEIVED	Item accepted at distribution center
DEP_SENT	The item shall be transported to another distribution center
DESTROYED	Destroyed
DISAPPEARED	It's gone
EDA	The shipment was detained at the distribution center of the recipient country
EDB	The item has been presented to the customs authorities of the country of destination
EDC	The consignment is subject to customs controls in the country of destination
EDD	Consignment at the distribution center in the country of destination
EDE	The shipment was sent from a distribution center in the recipient country
EDF	The shipment is on hold in the recipient's post office
EDG	The shipment has been delivered for delivery
EDH	The shipment was delivered to the place of collection
EMA	Consignment accepted from sender
EMB	Consignment at distribution center
EMC	Consignment shipped from Lithuania
EMD	Consignment in the country of destination
EME	Consignment at the customs office of destination
EMF	The shipment was sent to the recipient's post office
EMG	Parcel at the recipient's post office
EMH	Attempt to deliver failed
EMI	The shipment has been delivered to the consignee
EXA	The consignment has been presented to the customs authorities of the country of departure

EXB	The consignment was detained at the office of departure
EXC	The consignment has been checked at the customs office of dispatch
EXD	The item is detained at the dispatch center of the country of dispatch
EXPORTED	Exported
EXX	The shipment has been canceled from the sender's country
FETCHCODE	The shipment was delivered to the parcel self-service terminal
HANDED_IN_BKIS	Served (BKIS)
HANDED_IN_POST	Served at the post office
HANDED_TO_GOVERNMENT	Transferred to the State
IMPLICATED	Included
INFORMED	Receipt message
LABEL_CANCELLED	Delivery tag canceled
LABEL_CREATED	Delivery tag created
LP_DELIVERY_FAILED	Delivery failed
LP_RECEIVED	The parcel was received at the Lithuanian Post Office
NOT_INCLUDED	Not included
NOT_SET	Unknown
ON_HOLD	Detained
PARCEL_DELIVERED	The shipment was delivered to the parcel self-service terminal
PARCEL_DEMAND	Secure on demand
PARCEL_DETAINED	Detained
PARCEL_DROPPED	The shipment is placed in the parcel self-service terminal for shipment
PARCEL_LOST	The shipment is gone
PARCEL_PICKED_UP_AT_LP	The shipment has been delivered to the receiver
PARCEL_PICKED_UP_BY_DELIVERYAGENT	The parcel is taken by courier from the parcel self-service terminal
PARCEL_PICKED_UP_BY_RECIPIENT	The shipment has been withdrawn by the receiver
RECEIVED_FROM_ANY_POST	Received
RECEIVED	Received
REDIRECTED_AT_HOME	Forwarded
REDIRECTED_IN_POST	Forwarded in post office
REDIRECTED	Forwarded-Served
REDIRECTING	Forwarding started
REDIRECTING	Forwarding started - Forwarding
REFUND_AT_HOME	Refunded
REFUNDED_IN_POST	Returned to post office
REFUNDED	Refunded
REFUNDING	Return started
SENT	Sent
STORING	Transferred to storage
STORING	Transferred to storage-Secure
TRANSFERRED_FOR_DELIVERY	Passed on for deliver

UNDELIVERED	Not delivered
UNSUCCESSFUL_DELIVERY	Delivery failed

3.21. ReceiverType

Table 96. ReceiverType data structure

Name	Type	Mandatory nature	Description
<i>name</i>	string	Required, [1..1]	Please provide the recipient's forename and surname. Note: Either a <i>name</i> or <i>company</i> field is required
<i>companyName</i>	string	Required, [1...1]	Please provide the company name. Note: Either a <i>name</i> or <i>company</i> name is required
<i>address</i>	AddressType (see Table 100)	Required, [1..1]	Shipping item receiver's details. Required if the unique identifier for the sender address is not provided. Note: For NOT Lithuanian receiver use NOTLithuanianAddressType
<i>phone</i>	string	Required, [1..1] (45-63 templates) Optional, [1..1] (40-44&66-67 templates)	Recipient's mobile number. A mobile phone number is strongly recommended, and is required when delivery choice notification at destination is by SMS
<i>email</i>	string	Required, [1..1]	Recipient's email. Mandatory for at least one of addressee's-telephone and addressee's-email must be provided
<i>terminalId</i>	string	Optional, [0..1]	If the destination is an LP Express terminal, this parameter must have the unique terminal identifier provided.
<i>postOfficeAddress</i>	PostOfficeAddressType (see Table 98)	Optional, [0..1]	If the destination is a post office, this parameter must have the unique post office and/or PO box identifier provided.

3.22. SenderType

Table 97. SenderType data structure

Name	Type	Mandatory nature	Description
<i>senderId</i>	string	Optional, [0..1]	Please provide the unique identifier of the sender's address. Required if the <i>address</i> data structure is not provided.
<i>address</i>	AddressType (see Table 100)	Optional, [0..1]	Shipping item sender's details. Required if the unique identifier of the sender's address is not provided.
<i>name</i>	string	Optional, [0..1]	Please specify the company name and/or the sender's forename and surname. Required if the unique identifier of the sender's address is not provided.
<i>phone</i>	string	Required, [1..1]	Sender's mobile number.
<i>Email</i>	string	Optional, [0..1]	Sender's email.

3.23. PostOfficeAddressType

Table 98. PostOfficeAddressType data structure

Name	Type	Mandatory nature	Description
------	------	------------------	-------------

<i>postOfficeId</i>	string	Required, [1..1]	Please provide the post office identifier.
<i>postOfficeBoxAddress</i>	PostOfficeBoxAddressType (see Table 99)	Optional, [0..1]	Provided if the shipping item is to be sent to a PO box.

3.24. PostOfficeAddressBoxType

Table 99. *PostOfficeAddressBoxType* data structure

Name	Type	Mandatory nature	Description
<i>postOfficeBoxId</i>	string	Required, [1..1]	Please provide the PO box identifier.

3.25. AddressTypes

Table 100. *AddressTypes* data structure

Name	Type	Mandatory nature	Description
<i>Structured Address</i>	Structured Address Type (see Table 101 StructuredAddressType data structureTable 101)	Required, [1..1]	Structured address to be provided where it is possible to structure the address.
<i>Unstructured Address</i>	Unstructured Address Type (see Table 102)	Optional, [0..1]	Unstructured address to be provided where it is not possible to structure the address. For an unstructured address submit a structured address is optional. NOTE: NOT applicable for SenderType and 70-73 templates
<i>NOT Lithuanian address</i>	NOT Lithuanian Address Type (see Table 103)	Optional, [0..1]	Address which is meant not for Lithuanian receiver

3.26. StructuredAddressType

Table 101 *StructuredAddressType* data structure

Name	Type	Mandatory nature	Description
<i>locality</i>	string	Required, [1..1]	Locality name (e.g. city/town, village name).
<i>street</i>	string	Required, [1..1]	Street name.
<i>building</i>	string	Required, [1..1]	House number.
<i>apartment</i>	string	Optional, [0..1]	Apartment number.
<i>postalCode</i>	string	Required, [1..1]	Postal code.
<i>country</i>	string	Required, [1..1]	Country code.

3.27. UnstructuredAddressType

Table 102 *UnstructuredAddressType* data structure

Name	Type	Mandatory nature	Description
------	------	------------------	-------------

<i>locality</i>	string	Required, [1..1]	Locality name (e.g. city/town, village name).
<i>postalCode</i>	string	Required, [1..1]	Postal code.
<i>country</i>	string	Required, [1..1]	Country code.
<i>freiformAddress</i>	string	Optional, [0..1]	Unstructured address to be provided where it is not possible to structure the address. For an unstructured address submit a structured address is optional. NOTE: NOT applicable for SenderType.

3.28. NOTLithuanianAddressType

Table 103. NOTLithuanianAddressType data structure

Name	Type	Mandatory nature	Description
<i>locality</i>	string	Required, [1..1]	Locality name (e.g. city/town, village name).
<i>postalCode</i>	string	Required, [1..1]	Postal code.
<i>country</i>	string	Required, [1..1]	Country code.
<i>address1</i>	string	Required, [1..1]	Address of the receiver
<i>address2</i>	string	Optional, [0..1]	Address of the receiver

3.29. AddressRequestType

Table 104. AddressRequestType data structure

Name	Type	Mandatory nature	Description
<i>locality</i>	string	Required, [1..1]	Locality name (e.g. city/town, village name).
<i>street</i>	string	Required, [1..1]	Street name.
<i>building</i>	string	Required, [1..1]	House number.
<i>apartment</i>	string	Optional, [0..1]	Apartment number.
<i>postalCode</i>	string	Required, [1..1]	Postal code.
<i>Country</i>	string	Required, [1..1]	Country code.

3.30. ShippingItemType

Table 105. ShippingItemType data structure

Name	Type	Mandatory nature	Description
<i>id</i>	string	Required, [1..1]	Shipping item unique identifier.
<i>template</i>	string	Required, [1..1]	Shipping item template data structure.
<i>type</i>	string	Optional, [0..1]	Shipping item type.
<i>title</i>	string	Optional, [0..1]	Shipping item name.
<i>description</i>	string	Optional, [0..1]	Shipping item description.
<i>summary</i>	string	Optional, [0..1]	Shipping item summary.

<i>externalId</i>	String	Optional, [0..1]	Shipping item identifier on the customer's system.
<i>createdOn</i>	DateTime	Required, [1..1]	Shipping item creation date.
<i>dateSent</i>	DateTime	Optional, [0..1]	Shipping item dispatch date.
<i>partCount</i>	int	Optional, [0..1]	Number of parts of the shipping item.
<i>status</i>	string	Required, [1..1]	Shipping item status.
<i>barcode</i>	string	Required, [1..1]	Shipping item barcode.
<i>price</i>	PriceType (see Table 112)	Optional, [0..1]	Shipping item price.
<i>size</i>	string, SizeConst (see Table 90)	Optional, [0..1]	Shipping item size.
<i>locality</i>	string	Required, [1..1]	Locality name (e.g. city/town, village name).
<i>street</i>	string	Required, [1..1]	Street name.
<i>building</i>	string	Required, [1..1]	House number.
<i>apartment</i>	string	Optional, [0..1]	Apartment number.
<i>postalCode</i>	string	Required, [1..1]	Postal code.
<i>country</i>	string	Required, [1..1]	Country code.

3.31. TemplateType

Table 106. *TemplateType* data structure

Name	Type	Mandatory nature	Description
<i>id</i>	string	Required, [1..1]	Unique number of the shipping item template on the open application programming interface of the Lithuanian Post.
<i>title</i>	string	Required, [1..1]	Shipping item template name
<i>summary</i>	string	Required, [1..1]	Shipping item template summary.
<i>description</i>	string	Required, [1..1]	Shipping item template description.

3.32. AdditionalServicesType

Table 107. *AdditionalServicesType* data structure

Name	Type	Mandatory nature	Description
<i>id</i>	int	Required, [1..1]	Unique code of an additional service.
<i>amount</i>	int	Optional, [0..1]	Attribute of the additional service, which determines, e.g., the value of the insurance cover for the shipping item.
<i>note</i>	string	Optional, [0..1]	Additional service summary.
<i>variant</i>	string	Optional, [0..1]	Attribute of an additional service variant.
<i>price</i>	PriceType (see Table 112)	Required, [1..1]	Additional service price.

3.33. AdditionalServicesPropertiesType

Table 108 AdditionalServicesPropertiesType data structure

Name	Type	Mandatory nature	Description
<i>id</i>	int	Required, [1..1]	Unique code of an additional service.
<i>amount</i>	number, double	Optional, [0..1]	Attribute of the additional service, which determines, e.g., the value of the insurance cover for the shipping item.
<i>note</i>	string	Optional, [0..1]	Additional service summary.
<i>variant</i>	string	Optional, [0..1]	Attribute of an additional service.
<i>price</i>	PriceType (see Table 112)	Required, [1..1]	Additional service price.
<i>NoteProperties</i>	NoteProperties (see Table 109)	Optional, [0..1]	Attribute of the appended summary of an additional service.
<i>amountProperties</i>	AmountProperties (see Table 110)	Optional, [0..1]	Ranges of the additional service amount.
<i>variantProperties</i>	VariantProperties (see Table 111)	Optional, [0..1]	Properties of the additional service variants.

3.34. NoteProperties

Table 109. NoteProperties data structure

Name	Type	Mandatory nature	Description
<i>note</i>	string	Required, [1..1]	Parameter indicating that the additional service requires a summary.

3.35. AmountProperties

Table 110. AmountProperties data structure

Name	Type	Mandatory nature	Description
<i>precision</i>	int	Required, [1..1]	Precision of the value to be entered.
<i>currency</i>	string	Required, [1..1]	Currency.
<i>min</i>	number, double	Required, [1..1]	Minimum limit of the permitted value.
<i>max</i>	number, double	Required, [1..1]	Maximum limit of the permitted value.

3.36. VariantProperties

Table 111. VariantProperties data structure

Name	Type	Mandatory nature	Description
<i>values</i>	string	Required, [1..1]	Available additional service variants.

3.37. PriceType

Table 112. *PriceType* data structure

Name	Type	Mandatory nature	Description
<i>amount</i>	number, double	Required, [1..1]	Price.
<i>currency</i>	string	Required, [1..1]	Currency.
<i>vat</i>	number, double	Optional, [0..1]	Value Added Tax.

3.38. ShippingDocumentsType

Table 113. *ShippingDocumentsType* data structure

Name	Type	Mandatory nature	Description
<i>cn22Form</i>	CN22FormType (see Table 114)	Optional, [0..1]	CN22 declaration details. Required where letter-post items are sent outside the EU countries. If the content value exceeds 325 EUR, it requires a CN23 declaration form to be completed.
<i>cn23Form</i>	CN23FormType (see Table 115)	Optional, [0..1]	CN23 declaration details. Required where parcels are sent outside the EU countries.

3.39. CN22FormType

Table 114. *cn22DocumentType* data structure

Name	Type	Mandatory nature	Description
<i>parcelType</i>	ParcelTypeConst (see Table 91 and Table 92)	Required, [1..1]	Shipping item available contents type depends on shipment item type if it is LP or LPEXPRESS.
<i>parcelTypeNotes</i>	string	Required, [1..1]	Shipping item contents. If "OTHER" is selected in the <i>parcelType</i> parameter, this field needs to detail "OTHER".
<i>description</i>	string	Required, [1..1]	Description of declared items.
<i>cnParts</i>	CNPartsType (see Table 116)	Required, [1..*]	Data structure of a declared item. Max 4 cnParts per declaration.

3.40. CN23FormType

Table 115. *cn23DocumentType* data structure

Name	Type	Mandatory nature	Description
<i>cnParts</i>	CNPartsType (see Table 116)	Required, [1..*]	Data structure of a declared item. Max 3 cnParts per declaration.
<i>exporterCustomsCode</i>	string	Optional, [0..1]	Sender's registration number with the customs authority (if any). Required for the CN23 declaration to be completed.
<i>parcelType</i>	ParcelTypeConst (see Table 91 and Table 92)	Required, [1..1]	Shipping item available contents type depends on shipment item type if it is LP or LPEXPRESS.
<i>parcelTypeNotes</i>	string	Optional, [0..1]	Shipping item contents. If "Other" is selected in the <i>parcelType</i> parameter, this field needs to have the item description provided.
<i>license</i>	string	Optional, [0..1]	Licence No of the item to be sent.
<i>certificate</i>	string	Optional, [0..1]	Certificate No of the item to be sent.

<i>invoice</i>	string	Optional, [0..1]	Invoice No of the item to be sent.
<i>notes</i>	string	Optional,[0..1]	Comment where the item is subject to restrictions.
<i>failureInstruction</i>	String(see Table 94)	Optional, [0..1]	Sender's instructions to follow in case of non-delivery. By Default: RETURN_TO_SENDER_NON_PRIORITY
<i>importerCode</i>	string	Optional, [0..1]	Importer's code.
<i>importerCustomsCode</i>	string	Optional, [0..1]	Importer's registration number with the customs authority (if any).
<i>importerEmail</i>	string	Optional, [0..1]	Importer's email.
<i>importerFax</i>	string	Optional, [0..1]	Importer's fax.
<i>importerPhone</i>	string	Optional, [0..1]	Importer's tel.
<i>importerTaxCode</i>	string	Optional, [0..1]	Tax payer's number.
<i>importerVatCode</i>	string	Optional, [0..1]	VAT identification number.
<i>description</i>	string	Optional, [0..1]	Description of declared documents.

3.41. CNPartsType

Table 116. *CNPartsType* data structure

Name	Type	Mandatory nature	Description
<i>amount</i>	int	Required, [1..1]	Item value. MAX value 999
<i>countryCode</i>	string	Required, [1..1]	Country of origin of the goods. It is possible to provide a country ID or a country code.
<i>currencyCode</i>	string	Required, [1..1]	Currency. Only EUR and USD
<i>hsCode</i>	int	Optional, [0..1]	HS item number. MAX 6 digits
<i>weight</i>	int	Required, [1..1]	Item weight in grams
<i>quantity</i>	int	Required, [1..1]	Item quantity in pieces.
<i>summary</i>	string	Required, [1..1]	Name of a declared item.

3.42. ShippingItemLabelType

Table 117. *ShippingItemLabelType* data structure

Name	Type	Mandatory nature	Description
<i>itemId</i>	string	Required, [1..1]	Unique shipping item identifier.
<i>contentType</i>	string	Required, [1..1]	MIME document type.
<i>label</i>	array byte	Required, [1..1]	Base64 label document.

3.43. DocumentResponseType

Table 118. *DocumentResponseType* data structure

Name	Type	Mandatory nature	Description
<i>cartId</i>	string	Required, [1..1]	Unique cart identifier.
<i>contentType</i>	string	Required, [1..1]	MIME document type.
<i>document</i>	array byte	Required, [1..1]	Base64 document.

3.44. ManifestResponseType

Table 119. Courier *ManifestResponseType* data structure

Name	Type	Mandatory nature	Description
<i>barcode</i>	string	Required, [1..1]	Unique barcode identifier of the cart element.
<i>contentType</i>	string	Required, [1..1]	MIME document type.
<i>document</i>	array byte	Required, [1..1]	Base64 document.

4. Template classification

All the templates are as a classification for all possible types of shipping items. Possibility to send some of the items by the template depends on which agreement you have with “Lithuanian Post”.

Table 120. *Template description*

Id	Type	Size	Title	Max weight(kg.)	Description
42	SMALL_CORESPONDENCE	S	Small correspondence shipment from post to receiver address/post.	0.5	Press, documents, prints etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 20mm b - Up to 381mm c - Up to 305mm.
43	BIG_CORESPONDENCE	M	(M) big correspondence shipment from post to receiver address/post.	2	Gifts, shoes, clothing etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 600mm b - Up to 600mm c - Up to 600mm.
44	PACKAGE	L	(L) large shipment from post to receiver address/post.	30	Bulky stuff, big appliances etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 1050mm b - Up to 1050mm c - Up to 1050mm. b + biggest dimension to any direction length <2000mm
45	EBIN	-	Shipment from home or office, delivered by LPEXPRESS courier.	30-1500	All sort of things: electronics, clothing etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 1500mm b - Up to 1500mm c - Up to 1500mm a + b + c < 34000 mm.
46	AB	-	Shipment from home or office, delivered to receivers post.	10	Press, documents, apparel etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 1050mm b - Up to 1050mm c - Up to 1050mm. b + biggest dimension to any direction length <2000mm.
49	CHCA	XS	Shipment from LPEXPRESS self-service terminal to receivers address.	30	Press, documents, apparel etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 185mm b - Up to 610mm c - Up to 80mm.
50	CHCA	S	Shipment from LPEXPRESS self-service terminal to receivers address.	30	Small electronics, toys, perfumery etc. Minimum item dimensions: 90 × 140 mm, ± 2 mm. a - 350 mm. b - 610 mm. c - 80 mm.
51	CHCA	M	Shipment from LPEXPRESS self-service terminal to receivers address.	30	Shoes, Clothes, Gifts etc. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 175mm.
52	CHCA	L	Shipment from LPEXPRESS self-service terminal to	30	Large appliances, bulky items. Minimum item dimensions: 90 ×

			receivers address.		140mm, ± 2mm. a - 350mm. b - 610mm. c - 365mm.
53	CHCA	XL	Shipment from LPEXPRESS self-service terminal to receivers address.	30	Large size items, bulky electronics. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 745mm.
54	HC	XS	Shipment from home or office, to LPEXPRESS to self-service terminal.	30	Press, documents, apparel etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 185mm b - Up to 610mm c - Up to 80mm.
55	HC	S	Shipment from home or office, to LPEXPRESS to self-service terminal.	30	Small electronics, toys, perfumery etc. Minimum item dimensions: 90 × 140 mm, ± 2 mm. a - 350 mm. b - 610 mm. c - 80 mm.
56	HC	M	Shipment from home or office, to LPEXPRESS to self-service terminal.	30	Shoes, Clothes, Gifts etc. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 175mm.
57	HC	L	Shipment from home or office, to LPEXPRESS to self-service terminal.	30	Large appliances, bulky items. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 365mm.
58	HC	XL	Shipment from home or office, to LPEXPRESS to self-service terminal.	30	Large size items, bulky electronics. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 745mm.
59	CC	XS	Shipment from and to LPEXPRESS self-service terminal.	30	Press, documents, apparel etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 185mm b - Up to 610mm c - Up to 80mm.
60	CC	S	Shipment from and to LPEXPRESS self-service terminal.	30	Small electronics, toys, perfumery etc. Minimum item dimensions: 90 × 140 mm, ± 2 mm. a - 350 mm. b - 610 mm. c - 80 mm.
61	CC	M	Shipment from and to LPEXPRESS self-service terminal.	30	Shoes, Clothes, Gifts etc. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 175mm.
62	CC	L	Shipment from and to LPEXPRESS self-service terminal.	30	Large appliances, bulky items. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 365mm.
63	CC	XL	Shipment from and to LPEXPRESS self-service terminal.	30	Large size items, bulky electronics. Minimum item dimensions: 90 × 140mm, ± 2mm. a - 350mm. b - 610mm. c - 745mm.
70	SMALL_COESPONDENCE_TRACKED	S	Small correspondence shipment from post to receiver address/post. Tracked. Only out of Lithuania	0.5	Press, documents, prints etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 20mm b - Up to 381mm c - Up to 305mm.
73	MEDIUM_COESPONDENCE_TRACKED	M	(M) big correspondence shipment from post to receiver address/post. Only out of Lithuania	2	Gifts, shoes, clothing etc. Minimum item dimensions: 90x140mm. ± 2mm. a - Up to 600mm b - Up to 600mm c - Up to 600mm.



5. CN declarations

for shipment out of Lithuania, different declarations are required depending on destination country, declared value and size.

5.1. Schema of required declarations for each shipping item template

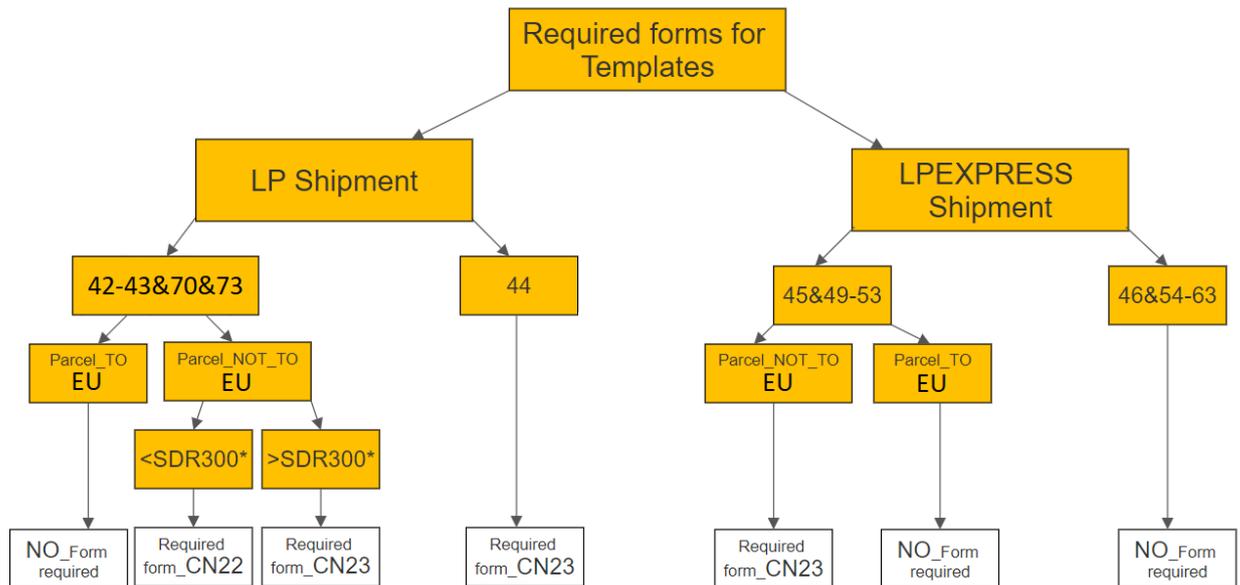


Figure 9. Schema of required CN declarations for each shipping item

First the required documents depends on which template you are going to use, the templates you are available to use depends on your rights and agreement with „Lithuanian post“.

If the parcel is NOT to EU the customs will require a CN22 or CN23 form**.

If the parcel is **42, 43, 70** or **73** template and sent **NOT** to **EU**, and is **LESS** than **300SDR*** the **CN22** form is **REQUIRED**, it has to be added to the parcel with the sticker.

If the parcel is **42, 43, 70** or **73** template and sent **NOT** to **EU**, and is **MORE** than **300SDR*** the **CN23** form is **REQUIRED**, it has to be added to the parcel with the sticker.

If the parcel is **44** template the **CN23** form is **REQUIRED** not depending on the value or the country, it has to be added to the parcel with the sticker.

If the parcel is **45&49-53** template and sent **NOT** to **EU**, the **CN23** form is **REQUIRED** not depending on the value, it has to be added to the parcel as it printed together with the sticker.

**More detailed information is in Lithuanian customs page: <https://www.lrmuitine.lt/web/guest/verslui/pastosiuntos/gabenimas#en>

*SDR-is the value of the parcel. The value of 1SDR is changing: https://www.imf.org/external/np/fin/data/rms_sdrv.aspx

5.2. JSON request example of CN forms

The fields for each form are required, there are more additional optional fields.

Note: For the forms to be generated first the parcel has to be initiated with an **initiate** method. **Then and only then the forms are generated** and the response is **cartId**, which has to be used to get all the generated forms **NOT itemId**. The **itemId** is used to get the **stickers**.

```

.....
"documents": {
  "cn23Form": {
    "parcelType": "DOCUMENT",
    "parcelTypeNotes": null,
    "description": "staff",
    "cnParts": [
      {
        "amount":376,
        "countryCode":"AF",
        "currencyCode":"EUR",
        "weight":2000,
        "quantity": 1,
        "summary": "test"
      }
    ]
  },
  "cn22Form": {
    "parcelType": "DOCUMENT",
    "parcelTypeNotes":"Confidential documents",
    "description": "Document for test",
    "cnParts": [
      {
        "amount":200,
        "countryCode":"AF",
        "currencyCode":"EUR",
        "weight":2000,
        "quantity": 5,
        "summary": "test"
      }
    ]
  }
}

```

5.3. Troubleshooting of CN forms

First make sure that you are using the correct **document form** for the right parcel **template**. Be sure if the parcel is going to **ES** or **NOT** to **ES**. Make sure that the declared value is **LESS** or **MORE 300SDR**. Check if **ALL REQUIRED** fields are input. Be sure you did **initiate** the parcel before trying to get all the required documents. Use only **cartId** for required **CN documents**. **CN22** declaration comes along sticker, the endpoint shall not return any documents.

6. Additional Services

For items to be sent via Lithuanian Post logistics chain, there are possibilities to add additional services while creating the orders.

6.1. Lithuanian Post and LPEXPRESS item type insurance differences

6.1.1. LPEXPRESS type items

All **LPEXPRESS type** shipments are automatically insured up to € 500.

The Contractor (for LPEXPRESS parcel) for the loss, damage of an unevaluated parcel shipped and delivered in Lithuania for his fault shall reimburse the value of the lost items (goods), but not more than EUR 500 per parcel, and its delivery costs. This applies only to LP EXPRESS shipments. No additional insurance is currently foreseen.

6.1.2. Lithuanian Post type items

But **Lithuanian Post type** items can be additionally insured.

In order to receive this service, it must be registered and priority. I.e. it is not enough to specify only the insurance service ID when setting up a shipment to additional services, you need to specify the IDs for all three services. Request should look like example below, whether it is a service added during a new creation or update method:

```
AdditionalServices: [  
  {  
    "Id": 2  
  },  
  {  
    "Id": 1  
  },  
  {  
    "Id": 4,  
    "Amount":50  
  }  
],
```

If you do not specify all 3 services during the update, the services will be overwritten and none will remain. Therefore, it is important to maintain the structure.

NOTE: IF the item is **42&43** template type and sent inside ES, only then they can be additionally insured. IF the item is **44** template type and is sent to the countries where it can be sent, then the item can be insured*.

* See a full detailed list of countries to which the item can be insured while sending:

https://www.post.lt/sites/default/files/Dokumentai/Tarptautini%C5%B3%20siunt%C5%B3%20duomen%C5%B3%20rikinys_2019_post.lt_Siuntiniai.pdf

6.2. Lithuanian Post item type additional services

6.2.3. Priority shipping item

Priority shipping item is an additional service provided by “Lithuania Post”. This service ensures that the item which is being shipped is delivered by the order of precedence. The item will have a special mark of „Prioritaire“ and will reach the receiver faster than the usual shipping item. It costs additional amount, which you can see by calling **options** for an item with the **/shipping/ {itemId}** which has been already **created** but **NOT** yet **initiated**. This service has to be ordered while **creating** the item or can be changed/added with an **update** function but just **before initiate** method, after the item has been initiated it is **impossible** to edit the item. All the process of ordering additional service you can see (Figure 2. Shipping item additional services management process diagram)

NOTE: Template 44 is by default priority and registered.

For more information please visit: <https://old.post.lt/en/private/postal-services/letter-post-items>

6.2.4. Registered shipping item

Registered shipping item is an additional service provided by “Lithuania Post”. This service ensures that the item which is being shipped, is delivered by the order of precedence and straight to the hands of the receiver, also there is given a code so you could track your order. It costs additional amount, which you can see by calling **options** for an item with the **/shipping/ {itemId}** which has been already **created** but **NOT** yet **initiated**. This additional service has to be made while **creating** the item or can be changed/added with an **update** function but just **before initiate** method, after the item has been initiated it is **impossible** to edit the item. **Note:** Very important. If the required additional service is a registered service, in the request form, in the field: “additionalServices” it is required to input **both priority** and **registered** services. (Unless the item is sent inside Lithuania, only then the item can contain only registered service). All the process of ordering additional service you can see (Figure 2. Shipping item additional services management process diagram)

NOTE: Template 44 is by default priority and registered.

For more information please visit: <https://old.post.lt/en/private/postal-services/letter-post-items>

6.2.5. Save item for 10 days

If the parcel fails to be delivered to the recipient, it is stored in the post office for 10 days and returned to the sender when the deadline expires. **Note:** Only available for items sent inside Lithuania.

6.2.6. Cash on delivery

The shipment is delivered to the recipient when he pays the service provider the amount of money indicated by the sender. The cost of the service is paid by the sender. The amount which has to be collected is paid to the sender to the bank account specified by the sender. **Note:** Only available for items sent inside Lithuania. The service is available if the item is **registered**.

6.3. LPEXPRESS item type additional services

6.3.1. Receiver pays the delivery fee

Shipping costs are paid by the recipient of the shipment on delivery. The parcel is given only after payment for services is done. **Note:** Only available for items sent inside Lithuania.

6.3.2. Delivery on demanded time

Shipments arrive by 10am. or from 6 p.m. **Note:** Only available for items sent inside Lithuanian cities: Vilnius, Kaunas, Klaipėda, Šiauliai, Panevėžys.

6.3.3. Parcel delivered with cover letter

To the parcel added documents signed by the receiver returnable to the sender. **Note:** Only available for items sent inside Lithuania.

6.3.4. Print the sticker at the terminal

Printing a parcel sticker at the LP EXPRESS parcel terminal.

6.3.5. Sticker will be delivered by the courier

If it is not possible to print the sticker, it will be prepared and glued on to the parcel by the courier on arrival to take the parcel. Once this service is ordered, the courier will come to take the shipment on the next working day.

6.3.6. Cash on delivery

The shipment is delivered to the recipient when he pays the service provider the amount of money indicated by the sender. The cost of the service is paid by the sender. The amount which has to be collected is paid to the sender to the bank account specified by the sender. **Note:** Only available for shipping items sent to Lithuania, Latvia and Estonia, and if parcels are of 45-th template type. **IF** parcels are of template type 49-53. Additional service is only available for parcels which are sent to Lithuania or Estonia.

7. Security

7.1. Authorization

Authorization of the users of the application programming interface uses the OAuth2 login standard. More information at <https://tools.ietf.org/html/rfc6749>

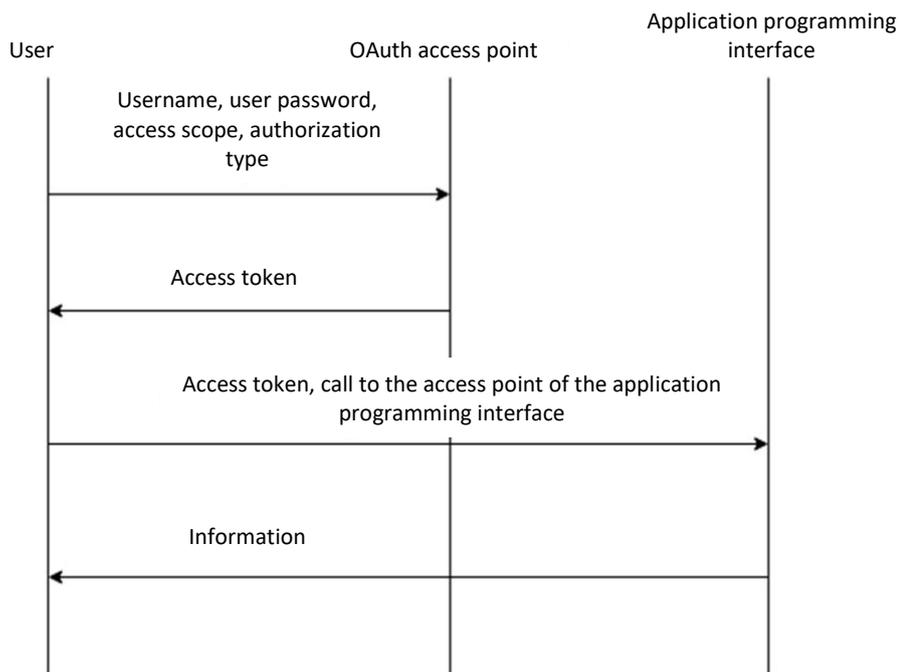


Figure 10. Authorization flow diagram

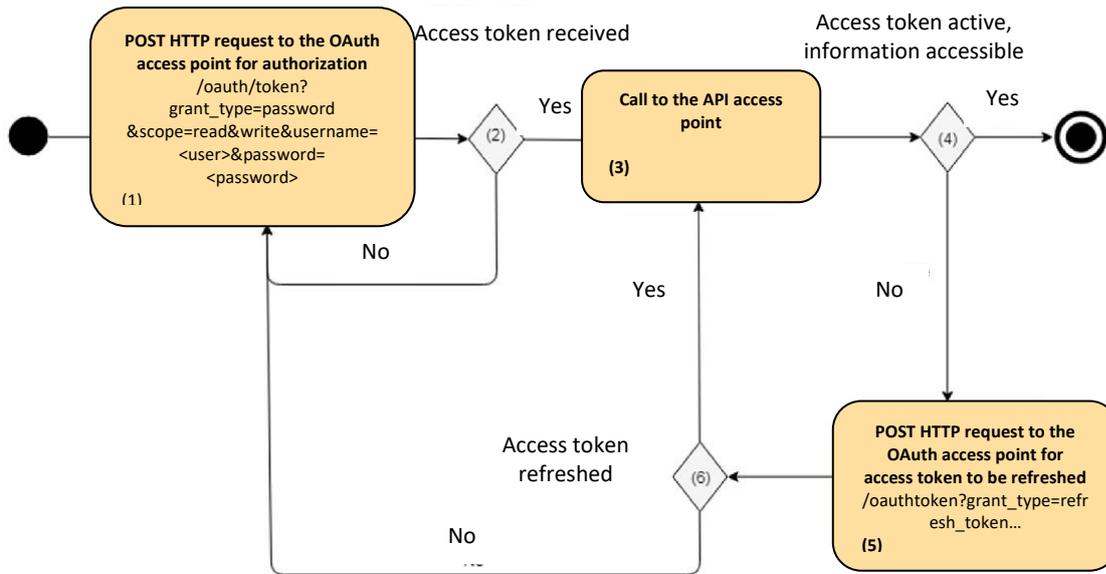


Figure 11. Authorization process diagram

Technical documentation

7.2. Addresses of method access points

Table 121. Addresses of method access points

Method type	Method name	Request parameters	Access point address
GET	<i>findCountries</i> (GET) method	[keyword, page, size]	/api/v1/address/country
GET	<i>getCountry</i> (GET) method	[countryId]	/api/v1/address/country/{countryId}
GET	<i>findLocalities</i> (GET) method	[countryId, keyword, page, postalCode, size]	/api/v1/address/country/{countryId}/locality
GET	<i>getLocality</i> (GET) method	[countryId, localityId]	/api/v1/address/country/{countryId}/locality/{localityId}
GET	<i>findLocalityPostalCodes</i> (GET) method	[countryId, localityId, page, postalCode, size, street]	/api/v1/address/country/{countryId}/locality/{localityId}/postalcode
GET	<i>findStreets</i> (GET) method	[countryId, keyword, localityId, page, postalCode, size]	/api/v1/address/country/{countryId}/locality/{localityId}/street
GET	<i>getStreet</i> (GET) method	[countryId, localityId, streetId]	/api/v1/address/country/{countryId}/locality/{localityId}/street/{streetId}
GET	<i>findBuildings</i> (GET) method	[countryId, keyword, localityId, page, postalCode, size, streetId]	/api/v1/address/country/{countryId}/locality/{localityId}/street/{streetId}/building
GET	<i>getBuilding</i> (GET) method	[buildingId, countryId, localityId, streetId]	/api/v1/address/country/{countryId}/locality/{localityId}/street/{streetId}/building/{buildingId}
GET	<i>findStreetPostalCodes</i> (GET) method	[countryId, keyword, localityId, page, postalCode, size, streetId]	/api/v1/address/country/{countryId}/locality/{localityId}/street/{streetId}/postalcode
GET	<i>findCountryPostalCodes</i> (GET) method	[countryId, keyword, page, size]	/api/v1/address/country/{countryId}/postalcode
GET	<i>getCountryPostalCodePostOffice</i> (GET) method	[countryId, postalCode]	/api/v1/address/country/{countryId}/postalcode/{postalCode}/postoffice

GET	<i>findPostBoxes (GET) method</i>	[countryId, keyword, page, size]	/api/v1/address/country/{countryId}/postbox
GET	<i>findPostOffices (GET) method</i>	[countryId, keyword, page, size]	/api/v1/address/country/{countryId}/postoffice
GET	<i>findPostOfficePostBoxes (GET) method</i>	[countryId, keyword, postOfficeId]	/api/v1/address/country/{countryId}/postoffice/{postOfficeId}/postbox
GET	<i>getDefaultValues (GET) method</i>	-	/api/v1/address/default
GET	<i>findTerminals (GET) method</i>	[page, size, templateId]	/api/v1/address/terminals
POST	<i>verify (POST) method</i>	[addressRequestType]	/api/v1/address/verification
POST	<i>createShippingItem (POST) method</i>	[external id, sender, receiver, weight, price, additional services, CN22, CN23]	/api/v1/shipping
PUT	<i>updateShippingItem (PUT) method</i>	[external id, sender, receiver, weight, price, additional services, CN22, CN23]	/api/v1/shipping/{itemId}
DELETE	<i>deleteShippingItem (DELETE) method</i>	[shipping item id]	/api/v1/shipping/{itemId}
POST	<i>initiateShipping (POST) method</i>	[shipping item ID's]	/api/v1/shipping/initiate
GET	<i>getShippingItems (GET) method</i>	[offset, limit, createdBefore, createdAfter]	/api/v1/shipping/F layout_
GET	<i>getShippingItem (GET) method</i>	[shipping item id]	/api/v1/shipping/{itemId}
OPTIONS	<i>getShippingItem Templates (OPTIONS) method</i>	-	/api/v1/shipping/shippingItemTemplates
OPTIONS	<i>getShippingItem Options (OPTIONS) method</i>	[shipping item id]	/api/v1/shipping/{itemId}
OPTIONS	<i>template (OPTIONS) method</i>	[template]	/api/v1/shipping/template/{template}
GET	<i>getSticker (GET) method</i>	[shipping item ID's, StickerPrintLayoutEnum]	/api/v1/documents/item/sticker/
GET	<i>getManifest (GET) method</i>	[cart ID]	/api/v1/documents/cart/{cartId}/manifest/
GET	<i>getManifestByBarcode (GET) method</i>	[item barcode]	/api/v1/documents/cart/{barcode}/manifest
GET	<i>getDocumentCn22Report (GET) method</i>	[cart ID]	/api/v1/documents/cart/{cartId}/cn22
GET	<i>getDocumentCn23Report (GET) method</i>	[cart ID]	/api/v1/documents/cart/{cartId}/cn23
POST	<i>callCourier (POST) method</i>	[item ID's]	/api/v1/shipping/courier/call
GET	<i>getTrackingByBarcode (GET) method</i>	[barcode]	/api/v1/tracking/byBarcode/{barcode}

POST	<i>getTrackingByItemIds</i> (POST) method	[item ID's]	/api/v1/tracking
------	--	-------------	------------------

7.3. Authorization

Login to the open application programming interface of the Lithuanian Post is effected by making a call to the **/oauth/token** access point with a **POST** request using the below request parameters:

Parameter	Permitted values of the parameter	Example of parameter	Parameter description
username	-	username=user@name.lt	User name.
password	-	password=Password	User password.
grant_type	password	grant_type=password	Type of access call.
clientSystem	PUBLIC	clientSystem=PUBLIC	Client subsystem constant.
scope	read+write	scope=read+write	Rights requested during the authorization in the application interface. The parameters are separated by a plus symbol.

Example of authorization request using the request parameters:

```
/oauth/token?username=user@name.lt&password=Password&grant_type=password&scope=read+write&clientSystem=PUBLIC
```

Response to the authorization token request:

```
{"access_token": "5604f19b-39aa-436c-8a42-5a68be7a4ccc", "token_type": "bearer", "refresh_token": "6c1aa408-b604-4e2f-b556-25f6d0949424", "expires_in": 9920556, "scope": "read write"}
```

Example of response to authorization request:

Parameter	Parameter description
access_token	Authorization access token.
token_type	Authorization access token type.
refresh_token	Authorization refresh token.
scope	Rights granted to the authenticated user.
expires_in	Timestamp in Unix format, which determines how long the authorization access token is valid.

If the user is authenticated in the open application programming interface of the Lithuanian Post, the authorization access token and the access token type are inserted in the *Authorization* parameter of the HTTP header at every request.

Example of the request header:

```
> POST /api/v1/shipping HTTP/1.1
> Host: <host>
> User-Agent: curl/7.61.1
> accept: application/json
> Authorization: bearer 16a998cf-58b3-45d4-bf09-3fd25e3e6663
> Content-Type: application/json
> Content-Length: 381
```

The authorization token in the open application programming interface of the Lithuanian Post is refreshed by making a call to the **/oauth/token** access point with a **POST** request using the below request parameters:

Parameter	Parameter values	Example of parameter	Parameter description
refresh_token	-	refresh_token=6c1aa408-b604-4e2f-b556-25f6d0949424	Authorization refresh key.

clientSystem	PUBLIC	clientSystem=PUBLIC	Client subsystem constant.
grant_type	refresh_token	grant_type=refresh_token	Type of access call.
scope	read+write	scope=read+write	Rights requested during the authorization in the application interface. The parameters are separated by an ampersand (&).

Example of authorization token refresh request using request parameters
 /oauth/token?grant_type=refresh_token&scope=read+write&clientSystem=PUBLIC&refresh_token=6c1aa408-b604-4e2f-b556-25f6d0949424

Response to the authorization token refresh request:
 {"access_token":"b2a56854-37fb-48b5-9383-8e457ad6e9c4","token_type":"bearer","refresh_token":"6c1aa408-b604-4e2f-b556-25f6d0949424","expires_in":9999998,"scope":"read write"}

7.4. Limitation to the number of access point requests

Access points are subject to the limitation of the number of requests. The remaining number of requests is returned in the request response header as the below values:

Parameter	Parameter description
X-RateLimit-Limit	Permitted number of requests per hour.
X-RateLimit-Remaining	Remaining number of permitted requests.

7.5. Potential errors

8.4.1. Error codes

Error code	Error description
200	Request accepted successfully.
400	Incorrect user request.
403	Request invalid with the provided login details.
429	Exceeded the limit of permitted requests.
500	Internal application error.

8.4.2. VerificationError

This error object is returned in the event of an input error.

Name	Type	Mandatory nature	Description
<i>fieldValidationErrors</i>	array FieldValidationError	Required, [1..*]	Validation error of the entered fields.
<i>valueValidationErrors</i>	array ValueValidationError	Required, [1..*]	Validation error of the entered values.

8.4.3. FieldValidationError

This error object contains information about the input field error.

Name	Type	Mandatory nature	Description
<i>field</i>	string	Required, [1..1]	Field name.
<i>code</i>	string	Required, [1..1]	Identification code of the field validation error.

Possible field validation identification codes:

Field validation code	Description
<i>UNSUPPORTED</i>	The entered field is not supported.

<i>REQUIRED</i>	The field is mandatory.
-----------------	-------------------------

8.4.4. ValueValidationError

This error object contains information about the error of the input field value.

Name	Type	Mandatory nature	Description
<i>field</i>	string	Required, [1..1]	Field name of the field value.
<i>code</i>	string	Required, [1..1]	Validation code of the field value.
<i>message</i>	string	Optional, [0..1]	Validation message of the field value.

Possible field value validation codes:

Value validation code	Description
<i>OUT_OF_RANGE</i>	The entered value is out of permitted value range.
<i>NUMERIC</i>	The entered value must be numeric.
<i>UNSUPPORTED</i>	The entered value is not supported.
<i>INVALID_FORMAT</i>	The format of the entered value is invalid.

8.4.5. Error

This error object is returned in case of an error which is not documented.

Name	Type	Mandatory nature	Description
<i>message</i>	string	Required, [1..1]	Error message.

7.6. Request examples

8.5.1. Example of create shipping item element

Example of a sent request using *curl*:

```
curl -X POST "<hostname>/api/v1/shipping" -H "accept: application/json" -H "Authorization: <authorization access token>" -H "Content-Type: application/json" -d '{"receiver": {"email": "test@post.lt", "name": "Test1", "address": {"country": "LT"}, "phone": "37064155444", "terminalId": "0109"}, "sender": {"address": {"apartment": "5", "building": "20", "country": "LT", "locality": "Vilnius", "postalCode": "10200", "street": "Gedimino pr."}, "email": "test2@post.lt", "name": "Test2", "phone": "37064155444"}, "template": "54"}'
```

Payload to be sent by the request:

```
{
  "receiver": {
    "email": "test@post.lt",
    "name": "Forename Surname",
    "address": {
      "country": "LT"
    },
    "phone": "37064155444",
    "terminalId": "0109"
  },
  "sender": {
    "address": {
      "apartment": "5",
      "building": "20",
      "country": "LT",
      "locality": "Vilnius",
      "postalCode": "10200",
      "street": "Gedimino pr."
    },
    "email": "test2@post.lt",
    "name": "Forename Surname",
    "phone": "37064155444"
  },
  "template": "54"
}
```

Response received to the request:

```
{
  "weight": 0,
  "partCount": 1,
  "sender": {
    "id": "4896",
    "name": "Test2",
    "address": {
      "country": "LT",
      "postalCode": "10200",
      "locality": "Vilnius",
      "street": "Gedimino pr.",
      "building": "20",
      "apartment": "5"
    },
    "email": "test2@post.lt",
    "phone": "37064155444"
  },
  "receiver": {
    "id": "4894",
    "name": "Test1",
    "address": {
      "country": "LT",
      "postalCode": "09109",
      "locality": "Vilnius"
    },
    "email": "test@post.lt",
    "phone": "37064155444"
  },
  "additionalServices": [],
  "id": "3101040",
}
```

```

"createdOn": "2019-01-09 12:11:34",
"barcode": "PENDING",
"price": {
  "amount": 3.27,
  "vat": 0,
  "currency": "EUR"
},
"status": "PENDING",
"type": "HC",
"size": "XSmall",
"title": "(XS) Shipping item from home or office to LP EXPRESS self-service terminal.",
"summary": "up to 30 kg",
"description": "Magazines, documents and jewellery...\r\nMinimum shipping item measurements: 90x140 mm ± 2 mm.\r\na – 185 mm\r\nb – 610 mm\r\nc – 80 mm"
}

```

Example of a shipping item with an unstructured address:

```

{
  "receiver": {
    "address": {
      "country": "LT",
      "freeformAddress": "Kažkokia gatvė kažkokiam mieste",
      "postalCode": "19200"
    },
    "email": "string",
    "id": "string",
    "name": "string",
    "phone": "37064057408"
  },
  "template": "45",
  "weight": 1
}

```

8.5.2. Shipping item field validation error

Response to the shipping item element validation error:

```

{
  "fieldValidationErrors": [
    {
      "field": "receiver#terminalId",
      "code": "REQUIRED"
    }
  ],
  "valueValidationErrors": []
}

```

8.5.3. Shipping item field value validation error

Response to the validation error of the shipping item element field value.

```

{
  "fieldValidationErrors": [],
  "valueValidationErrors": [
    {
      "field": "sender#phone",
      "message": "Invalid format. Should contain 37000000000",
      "code": "INVALID_FORMAT"
    }
  ]
}

```

```
]
}
```

8.5.4. Initiate shipping item request

Example of a sent request using curl:

```
curl -X POST ""<hostname>/api/v1/shipping/initiate" -H "accept: application/json" -H "Authorization: bearer <authorization access token>" -H "Content-Type: application/json" -d ["5001266"]
```

Payload to be sent by the request:

```
[
  "5001266"
]
```

Response received to the request:

```
[
  "5000232"
]
```

8.5.5. Get stickers request

Example of a sent request using curl:

```
curl -X GET
"<hostname>/api/v1/documents/item/sticker/?itemId=5001266&itemId=5001267&layout=LAYOUT_MAX" -H
"accept: application/json" -H "Authorization: bearer <authorization access token>"
```

8.5.6. Sending a priority shipping item

```
{
  "externalId": "SP-2s5687",
  "template": 42,
  "receiver": {
    "address": {
      "country": "RU",
      "locality": "V",
      "street": "Миклухо-Маклая",
      "building": "55",
      "postalCode": "117279"
    },
    "name": "Max",
    "phone": "+37013123110",
    "email": "test@test.com"
  },
  "sender": {
    "address": {
      "country": "LT",
      "locality": "Vilnius",
      "postalCode": "03163",
      "building": "1456",
      "street": "Jasinskio, gatvė"
    },
    "email": null,
    "name": "Alex",
    "phone": "+37012312312"
  },
  "partCount": 1,
  "weight": 500,
  "additionalServices": [
    {
      "id": 2
    }
  ],
  "documents": {
    "cn22Form": {
      "parcelType": "DOCUMENT",
      "parcelTypeNotes": "Confidential documents",
      "description": "Document for test",
      "cnParts": [
        {
          "amount": 200,
          "countryCode": "RU",
          "currencyCode": "EUR",
          "weight": 500,
          "quantity": 5,
          "summary": "test"
        }
      ]
    }
  }
}
```

8.5.7. Sending a registered shipping item

```

{
  "externalId": "SP-2s5687",
  "template": 42,
  "receiver": {
    "address": {
      "country": "RU",
      "locality": "V",
      "street": "Миклухо-Маклая",
      "building": "55",
      "postalCode": "117279"
    },
    "name": "Max",
    "phone": "+37013123110",
    "email": "test@test.com"
  },
  "sender": {
    "address": {
      "country": "LT",
      "locality": "Vilnius",
      "postalCode": "03163",
      "building": "1456",
      "street": "Jasinskio, gatvė"
    },
    "email": null,
    "name": "Alex",
    "phone": "+37012312312"
  },
  "partCount": 1,
  "weight": 500,
  "additionalServices": [
    {
      "id": 1
    },
    {
      "id": 2
    }
  ],
  "documents": {
    "cn22Form": {
      "parcelType": "DOCUMENT",
      "parcelTypeNotes": "Confidential documents",
      "description": "Document for test",
      "cnParts": [
        {
          "amount": 200,
          "countryCode": "RU",
          "currencyCode": "EUR",
          "weight": 500,
          "quantity": 5,
          "summary": "test"
        }
      ]
    }
  }
}

```

//NOT required sending inside Lithuania

8.5.8. Sending item with an unstructured receivers address

```
{
  "template": 42,
  "receiver": {
    "address": {
      "country": "ES",
      "locality": "Valencia",
      "freeformAddress": "Plaza del Ayuntamiento 24, 46002 "
    },
    "name": "Max",
    "phone": "+37013123110",
    "email": "test@test.com"
  },
  "sender": {
    "address": {
      "country": "LT",
      "locality": "Vilnius",
      "postalCode": "03163",
      "building": "1456",
      "street": "Jasinskio, gatvė"
    },
    "email": null,
    "name": "Alex",
    "phone": "+37012312312"
  },
  "partCount": 1,
  "weight": 500
}
```

Note: Sending to Lithuania the **country** and **postalCode** is required, for sending to other counties, only the **country** code is required!

7.6.5. Sending item with a structured receiver's address

```
{
  "template": 42,
  "receiver": {
    "address": {
      "country": "LT",
      "locality": "Kaunas",
      "street": "Laisvės al.",
      "building": "102",
      "postalCode": "44001"
    },
    "name": "Max",
    "phone": "+37013123110",
    "email": "test@test.com"
  },
  "sender": {
    "address": {
      "country": "LT",
      "locality": "Vilnius",
      "postalCode": "03163",
      "building": "1456",
      "street": "Jasinskio, gatvė"
    },
    "email": null,
    "name": "Alex",
    "phone": "+37012312312"
  },
  "partCount": 1,
  "weight": 500
}
```

Date	Version	Comment
13/02/2019	V1.0.1	New method created. Template (OPTIONS). This is an information method, i.e. to get information about the template.
08/03/2019	V1.0.2	Validation logic has been rearranged in the createShippingItem method. BP shipping items have a telephone number or an email provided as mandatory.
03/04/2019	V1.0.3	The sticker constant has been amended. From LAYOUT_10x5 to LAYOUT_10x15. (This version has not yet been installed in the testing environment as at 09/04/2019)
09/04/2019	V1.0.4	The description of sample requests has been supplemented.
28/05/2019	V1.0.5	<p>An incorrect structure has been fixed (from title to name) (3.4 ReceiverType and 3.7 SenderType)</p> <p>Documentation for unstructured addresses added.</p> <p>The structure of receiver type has been supplemented.</p>
11/07/2019	V1.0.6	<p>Documentation for Tracking added.</p> <p>The structure of receiver mobile phone and e-mail supplemented.</p>
29/10/2019	V1.0.7	<p>CN and CP form filling and requirement explanations added.</p> <p>Restructure of tables "types".</p> <p>Getting started page created.</p> <p>Additional services explanation added.</p> <p>Request examples added.</p> <p>Tracking Type list added</p> <p>Templates description added</p>

02/01/2020	V1.0.8	Max cnParts number Contents Type Additional Services
13/01/2020	V1.0.9	CP form elimination, CN23 additional fields,
18/02/2020	V1.0.10	P1
24/09/2020	V1.1.1	Address structure changes and additional addresses