



SPEDISCI E RITIRA NEL MODO PIÙ IN

InPost Italy

Client Integration Guide



InPost in Italy: Courier Services



This guide has the intent to provide the mapping between the logistic processes and how Client and InPost systems have to communicate with each other to support them.

This document describes the standard integration process in Italy, supported by the ShipX API. It's just a summary, and does not substitute the official tech documents.

From the **System Integration** point of view, the **Clients** are considered as the entities the InPost systems integrate with.

Our Clients are mainly

- eTailers that implement their business as Shop Online
- Sending Platforms that sell the logistic services to other Consumer Final Customers or other Business Customers

In Italy available services are:

- InPost Point to Point: For Clients that offer c2c services
- Client Hub/Warehouse to InPost Point
- InPost Point to Client Hub: Usually used for the return requests

InPost point can be a Parcel Locker or a PUDO (PickUP/Drop Off Point)

As InPost Italy we provide continuous support services for both **Integrated Clients** and for their **Final Customers** as well, from the Integration Design to the activities in Production when the integration is up&running

B2C

Hub to InPost Point



Sender create the label



Parcel is collected from Client's HUB



InPost delivers parcel to the Parcel locker/PUDO



Receiver collects the parcel from the Parcel locker/PUDO

Return with label



Customer requests the label from the sender and prints it



Customer delivers parcel to the Parcel locker/PUDO



InPost collects the parcel



InPost delivers it to the sender warehouse

C2C

Point to Point



Sender creates the label and prints it



Sender drop off the parcel to the Parcel locker/PUDO



InPost collects the parcel



InPost delivers parcel to the destination Parcel locker/PUDO



Receiver collects the parcel

InPost Paper-less Logistics Services (new)



B2C

Return without label



Customer starts the paper-less return process from the client's site



Customer delivers parcel to the Parcel locker/PUDO



InPost collects the parcel



InPost delivers it to the sender warehouse

C2C

Point to Point without label



Sender creates the shipment request without label



Sender drop off the parcel to the Parcel locker/PUDO



InPost collects the parcel



InPost delivers parcel to the destination Parcel locker/PUDO



Receiver collects the parcel

B2C

Click&Collect



Sender create the label on Client's site and chooses to pickup in the locker hosted in the shop



Client's Shop Staff checks the capacity availability of the specific locker



Client's staff delivers parcel to the Parcel locker, with the courier credentials

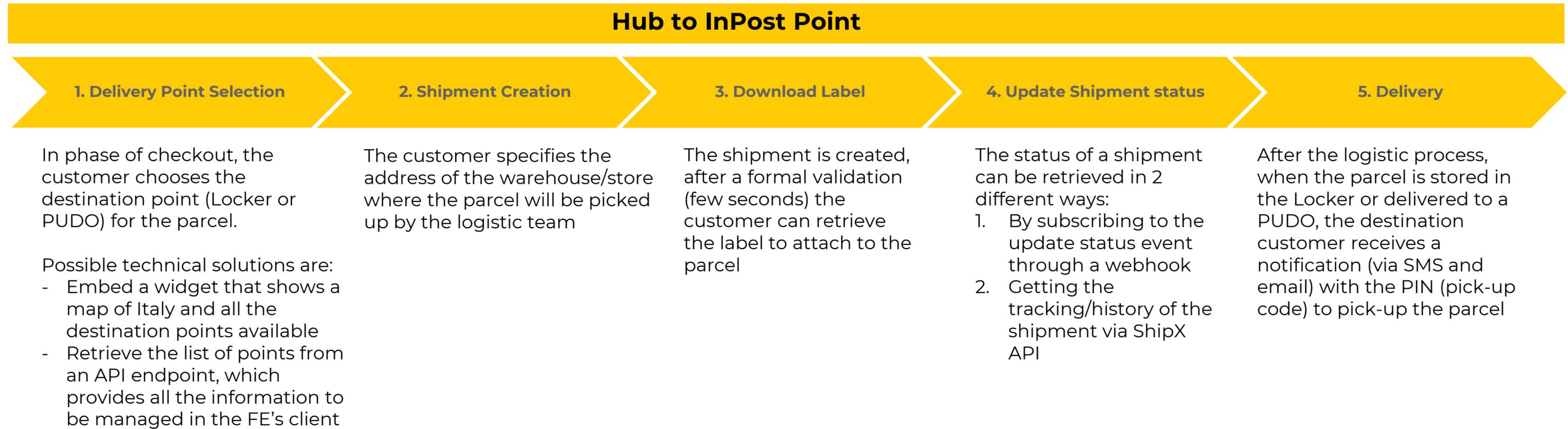


Receiver collects the parcel from the Parcel locker

Main principles/configurations supporting this process:

- InPost Logistics is not involved in the process; All the logistic actions are supposed to be done by the Client's employees
- Only lockers closed to Client's shops will be enabled to the service. The service is not foreseen for PUDOs
- The account configuration is different than the standard outbound (H2L) account, even if the Client is the same
- Labels creation will be through ShipX APIs, the capacity report (if needed) will be sent via ftp integration
- Expired parcels management (after the max time in the locker) will be in charge to Client's employees

Delivery experience B2C



Delivery experience B2C

Return with Label

1. Delivery Point Selection

In phase of return, the system can retrieve the return destination address in the integration via API or it can be assigned by the system itself. The destination address is mandatory when a client have more than one warehouses or want to have return in store.

2. Shipment Creation

The customer doesn't need to specify the origin of the shipment; he can drop off the parcel in any point of the InPost Logistics Network

3. Download Label

The return shipment is created and after a formal validation (few seconds), the customer can retrieve the label to attach to the parcel

4. Update Shipment status

The status of a shipment can be retrieved in 2 different ways:

1. By subscribing to the update status event through a webhook
2. Getting the tracking/history of the shipment via ShipX API

5. Delivery

The parcel is delivered by the InPost Logistic Team to the defined address, as for standard process

Delivery experience B2C

Return Paper-less

1. Delivery Point Selection

In phase of return, the system can retrieve the return destination address in the integration via API or it can be assigned by the system itself. The destination address is mandatory when a client have more than one warehouses or want to have return in store.

2. Shipment Creation

The customer doesn't need to specify the origin of the shipment; he can drop off the parcel in any point of the InPost Logistics Network

3. Download Label

The return shipment is created
The customer receives a notification via email and sms with all the information to drop-off the parcel

4. Update Shipment status

The status of a shipment can be retrieved in 2 different ways:

1. By subscribing to the update status event through a webhook
2. Getting the tracking/history of the shipment via ShipX API

5. Delivery

The parcel is delivered by the InPost Logistic Team to the defined address, as for standard process

Delivery experience C2C

Point to Point

1. Delivery Point Selection

In phase of checkout, the customer chooses the destination point (Locker or PUDO) for the parcel.

Possible technical solutions are:

- Embed a widget that shows a map of Italy and all the destination points available
- Retrieve the list of points from an API endpoint, which provides all the information to be managed in the FE's

2. Shipment Creation

The customer doesn't need to specify the origin of the shipment; they can drop off the parcel in any point of the InPost Logistic Network

3. Download Label

The shipment is created, after a formal validation (few seconds) the customer can retrieve the label to attach to the parcel

4. Update Shipment status

The status of a shipment can be retrieved in 2 different ways:

1. By subscribing the update status event through a webhook
2. Getting the tracking/history of the shipment via ShipX API

5. Delivery

After the logistic process, when the parcel is stored in the Locker or delivered to a PUDO, the destination customer receives a notification via SMS and email with the PIN (pick-up code) to pick-up the parcel

Delivery experience C2C

Point to Point Paper-less

1. Delivery Point Selection

In phase of checkout, the customer chooses the destination point (Locker or PUDO) for the parcel.

Possible technical solutions are:

- Embed a widget that shows a map of Italy and all the destination points available
- Retrieve the list of points from an API endpoint, which provides all the information to be managed in the FE's

2. Shipment Creation

The customer doesn't need to specify the origin of the shipment; they can drop off the parcel in any point of the InPost Logistic Network

3. Download Label

The return shipment is created
The customer receives a notification via email and sms with all the information to drop-off the parcel

4. Update Shipment status

The status of a shipment can be retrieved in 2 different ways:

1. By subscribing the update status event through a webhook
2. Getting the tracking/history of the shipment via ShipX API

5. Delivery

After the logistic process, when the parcel is stored in the Locker or delivered to a PUDO, the destination customer receives a notification via SMS and email with the PIN (pick-up code) to pick-up the parcel

Delivery experience **Click&Collect**

Click&Collect

1. Delivery Point Selection

In phase of checkout, the customer selects Click&Collect as delivery method; and then the shop where he wants to collect the order. Only the list of Lockers related to the Client's Shops are enabled to the service

2. Shipment Creation

The customer specifies the address of the warehouse/store where the parcel will be collected by the Client's logistic team. The only mandatory requirement for the label creation is to use as target point one of the enabled lockers for the service

3. Download Label

The shipment is in the specific status that allows its immediate delivery to the destination point. After the label is attached to the parcel, in the Client's warehouse, it will be inserted in the locker by the Client's staff by scanning the label's barcode. **Label is standard Hub2Locker Label**

4. Update Shipment status

The status of a shipment can be retrieved in 2 different ways:

1. By subscribing the update status event through a webhook
2. Getting the tracking/history of the shipment via ShipX API

5. Delivery

When the parcel is stored in the Shop's Locker, the final customer receives a notification via SMS and email with the PIN (pick-up code) to pick-up the parcel

1. Delivery Point Selection

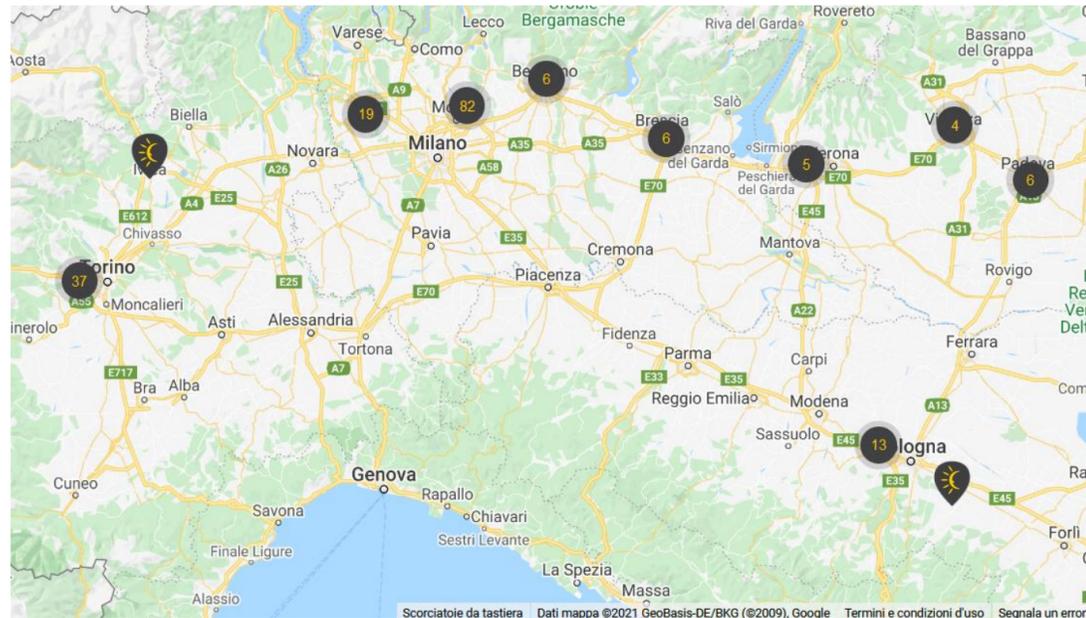


In case of delivery to an InPost Point:

- Points can be Automated Parcel Machines (Lockers) or PickUp DropOff (PUDO) Managed Point
- Only Points in status "Operating" can be selected

Geowidget

The map contains all the operating points within the InPost Logistics Coverage Area



Points API

<https://api-it-local-points.easypack24.net/v1/points>

- The list of points can be retrieved with the location data (longitude and latitude)
- Only points in **status** "Operating" can be selected
- The "**Location_247**" attribute indicates whether the location is always open or not. If "Location_247" = False, then the opening hours can be retrieved from the "operating_hours_extended" attribute
- **Types** of the points in use in Italy:
 - **Parcel locker** are the automated parcel machine that are pickup points (lockers) where the customer can pick the parcel without any interaction.
 - "**Pok**" are the PUDOs (pickup and drop-off points) inside a shop or small business, the customer in this case is supported by the shop staff
- For location name, the best solution would be:
 - Name = "**locker**" + "**Location_description**" for parcel_lockers
 - Name = "**InPost Point**" + "**location_description**" for PUDOs (Type 'pok')
- "**Location_description_1**" will provide optional further indications for the user to reach the specific point (for example, the floor where the locker is installed)

Points API Value information



Points API <https://api-it-local-points.easypack24.net/v1/points>

The API Points provide the most important information about the Points in the logistic network.

They can basically be 2 different **types**:

- PUDOs: Small business/shops that are used to pickup or drop-off parcels. In the API outcome are identified with type «pok»
- APMs (automated parcels machines): Automated lockers integrated in the network identified with the type «parcel_locker»

The **Statuses** that can adopt a point are:

- Created => Point configuration/deployment is in progress. Visible in the API return, but cannot be selected by the end user.
- Operating => Point ready to be used; Visible in the API result and is the only status selectable by end users
- Non Operating => Point temporarily out of order. Visible in the API return, but cannot be selected by the end user.
- Disabled => Point not retrieved by the API

A point can change its status at any moment, and depending on the change done, it can appear or disappear from the API output.

Other useful information:

- “location_description_1” will provide optional further indications for the user to reach the specific point
- “operating_hours_extended” includes the operating times for the point for everyday of the week, calculated in minutes after the midnight

How to display them?

From the end-user point of view, the best solution is to show them directly in a map, considering the area of interest of the customer

For location name, the best solution would be:

Name = “locker” (fixed string) + “location_description” for parcel_lockers

Name = “InPost Point” (fixed string) + “location_description” for PUDOs (Type ‘pok’)

Check-out InPost Standard



Out of home delivery selection needs a specific communication helping the end-user to understand what he can expect of this service. For that, based on our experience, we've defined the **Standard InPost** - it is a selected group of elements positioning InPost in the Merchant check out process and give a clear vision to the user of the advantages that he will get by selecting this delivery option.

As part of this standard, following points are suggestions than can be implemented on the Client's site check-out UX:

- 1. Map with APMs / PUDOs (geowidget):** When the Points are shown on a map, instead of a flat list, for a clear visibility of the points location
- 2. Search field on map (by zip code, city, street):** Provide the user more options to get to the list of preferred points, will provide the users with access to transparent information, enabling them to choose their preferred delivery location; For that the Points API provides different ways to be queried and/or some filters for the result list
- 3. Frequent updated APMs/PUDOs list** (daily at least), better if it's got online through the points API each time that needs to be displayed
- 4. Correct, full name for each Point** (as referred in the previous slide); with a clear differentiation between Parcel lockers and PUDOS; the 24/7 availability for lockers should be evident for the end user
5. As a general consideration; a **responsive web design** of e-shop; better accessibility of shop websites on mobile devices would be a significant incentive to make a purchase in a given shop

2. Shipment Creation



<https://developers.inpost.it/#tag/Shipments>

The Shipment will be created through the Shipment Shipx API. This API can be invoked in many ways, for the Italian market the Simplified Mode is the standard. Some simple information will be asked, such as Sender and Receiver, size of the parcel and the destination.

Other info:

- In the Locker2Locker service the customer can drop the parcel in whatever Locker or PUDO point he prefers
- The service in Locker or PUDO (Pick Up and Drop-Off) point are totally equivalent, it just changes from the customer experience
- Technically, in the checkout phase a new Shipment is created considering:
 - The request will be created in simplified mode
 - The origin point is not needed but the destination is mandatory
 - Sender is optional, if it's not present in the shipment creation, the system will automatically take as sender the Client Configuration
 - The customer can send the reference to be printed in the label and also the customer reference ID for customer support service

2. Shipment Creation Details (1/5)

[POST /v1/organizations/:organization_id/shipments](https://api.inpost.pl/v1/organizations/:organization_id/shipments)

Hub2Locker

Parameter	Type	Description	Validation	Mandatory
receiver	Peer	Parcel recipient's data	The attribute is required.	Y
sender	Peer	Parcel sender's data.	The attribute is not required. • If no data are provided, the data of the organization under which the shipment is being created will be used by default (only in case of single warehouse).	N
parcels	Array[ParcelsSimpleForm]	Data of the parcels included in the shipment. Dimensions and weight are expected	only one parcel per shipment:	Y
custom_attributes	CustomAttributesForm	Additional shipment attributes, e.g. "custom_attributes": { "target_point": "KRA010" } The target point has to be in Operating Status	The attribute is required.	Y
reference	String	Additional shipment description, e.g. order number. This information will be printed in the label	The attribute is not required. • Minimum 3 characters, maximum 100 characters, possibility to provide an empty attribute.	N
service	String	Service selected by the client; in this scenario will be set as inpost_locker_standard .	The attribute is required	Y
external_customer_id	String	ID of the broker generating shipments within a different organization for customer service purpose	The attribute is not required.	N

2. Shipment Creation Details(2/5)

[POST /v1/organizations/:organization_id/shipments](https://api.inpost.pl/v1/organizations/:organization_id/shipments)

Locker2Locker

Parameter	Type	Description	Validation	Mandatory
receiver	Peer	Parcel recipient's data	The attribute is required.	Y
sender	Peer	Parcel sender's data.	The attribute is not required. • If no data are provided, the data of the organization under which the shipment is being created will be used by default.	N
parcels	Array[ParcelsSimpleForm]	Data of the parcels included in the shipment. Dimensions and weight are expected	only one parcel per shipment	Y
custom_attributes	CustomAttributesForm	Additional shipment attributes, e.g. "custom_attributes": { "sending_method": "parcel_locker", "target_point": "KRA010"}	Both attributes are required.	Y
reference	String	Additional shipment description, e.g. order number. This information will be printed in the label	The attribute is not required. • Minimum 3 characters, maximum 100 characters, possibility to provide an empty attribute.	N
service	String	Service selected by the client; in this scenario will be set as :inpost_locker_standard .	The attribute is required	Y
external_customer_id	String	ID of the broker generating shipments within a different organization for customer service purpose	The attribute is not required.	N

2. Shipment Creation Details(3/5)

[POST /v1/organizations/:organization_id/shipments](https://api.inpost.pl/v1/organizations/:organization_id/shipments)

Locker2Locker label-less

Parameter	Type	Description	Validation	Mandatory
receiver	Peer	Parcel recipient's data	The attribute is required.	Y
sender	Peer	Parcel sender's data.	The attribute is not required. • If no data are provided, the data of the organization under which the shipment is being created will be used by default.	N
parcels	Array[ParcelsSimpleForm]	Data of the parcels included in the shipment. Dimensions and weight are expected	only one parcel per shipment	Y
custom_attributes	CustomAttributesForm	Additional shipment attributes, e.g. "custom_attributes": { "sending_method": "any_point", "target_point": "KRA010"} }	Both attributes are required.	Y
reference	String	Additional shipment description, e.g. order number. This information will be printed in the label	The attribute is not required. • Minimum 3 characters, maximum 100 characters, possibility to provide an empty attribute.	N
service	String	Service selected by the client; in this scenario will be set as :inpost_locker_standard .	The attribute is required	Y
additional_services	Structure	Additional service required "labelless"	The additional service is required	Y
external_customer_id	String	ID of the broker generating shipments within a different organization for customer service purpose	The attribute is not required.	N

2. Shipment Creation Details(4/5)

[POST /v1/organizations/:organization_id/shipments](https://api.inpost.com/v1/organizations/:organization_id/shipments)

Locker2Hub

Parameter	Type	Description	Validation	Mandatory
receiver	Peer	Parcel recipient's data	The attribute is required.	Y
sender	Peer	Parcel sender's data.	The attribute is not required. • If no data are provided, the data of the organization under which the shipment is being created will be used by default.	N
parcels	Array[ParcelsSimpleForm]	Data of the parcels included in the shipment. Dimensions and weight are expected	only one parcel per shipment	Y
custom_attributes	CustomAttributesForm	Additional shipment attributes, e.g. "custom_attributes": { "sending_method" : "parcel_locker"	The attribute is required.	Y
reference	String	Additional shipment description, e.g. order number. This information will be printed in the label	The attribute is not required. • Minimum 3 characters, maximum 100 characters, possibility to provide an empty attribute.	N
service	String	Service selected by the client; in this scenario will be set as inpost_courier_c2c .	The attribute is required	Y
external_customer_id	String	ID of the broker generating shipments within a different organization for customer service purpose	The attribute is not required.	N

2. Shipment Creation Details(5/5)

[POST /v1/organizations/:organization_id/shipments](https://api.inpost.pl/v1/organizations/:organization_id/shipments)

Locker2Hub label-less

Parameter	Type	Description	Validation	Mandatory
receiver	Peer	Parcel recipient's data	The attribute is required.	Y
sender	Peer	Parcel sender's data.	The attribute is not required. • If no data are provided, the data of the organization under which the shipment is being created will be used by default.	N
parcels	Array[ParcelsSimpleForm]	Data of the parcels included in the shipment. Dimensions and weight are expected	only one parcel per shipment	Y
custom_attributes	CustomAttributesForm	Additional shipment attributes, e.g. "custom_attributes": { "sending_method": "any_point"	The attribute is required.	Y
reference	String	Additional shipment description, e.g. order number. This information will be printed in the label	The attribute is not required. • Minimum 3 characters, maximum 100 characters, possibility to provide an empty attribute.	N
service	String	Service selected by the client; in this scenario will be set as inpost_courier_c2c .	The attribute is required	Y
additional_services	Structure	Additional service required "labelless"	The additional service is required	Y
external_customer_id	String	ID of the broker generating shipments within a different organization for customer service purpose	The attribute is not required.	N

3. Download Label



After the shipment confirmation the customer can download the label to be attached to the parcel by invoking the ShipX label API with the Shipment_id

- Possible formats are .pdf and .zpl
- The customer can also specify the dimension (normal, A6) on shipment basis

AREA	MICROAREA	InPost www.inpost24.it	CODICE HUB	SEDE OPERATIVA DESTINATARIA
BA ¹	2	3	4	5 BA
<div style="display: flex; align-items: center;"> 6 </div> <div style="display: flex; align-items: center; margin-top: 5px;"> 7 820074629270021015983032 </div>				
8 A 9 ADF00205P Estramurale Pozzo Zuccaro 22 Edicola di Acquaviva 70021 Acquaviva delle		DATA STAMPA ETICHETTA 12 2022.03.15		
		13 Consegna in qualsiasi locker		
MITTENTE	Name Surname calabiana 6 00171 Roma		14	ASSICURAZIONE
			15	CONTRASSEGNO
			16	DROP-OFF
DESTINATARIO	Name Surname calabiana 6 00171 Roma		17 NUMERO RIFERIMENTO CLIENTE	
			Reference For Label	
Condividi con noi il tuo giudizio sul servizio - myexperience@inpost24.it				

- Fields 1-7 are automatically calculated by the system
- Field 8 corrisponds to the template of the specific parcel

Template	Dimensions	Weight	Description
small	8 x 38 x 64 cm	do 25 kg	Size A
medium	19 x 38 x 64 cm	do 25 kg	Size B
large	41 x 38 x 64 cm	do 25 kg	Size C

- Field 9 is the target point (Locker or PUDO name and Hub)
- Field 10, 11 are Name and Adresses for sender and receiver
- Field 13 indicates if the parcel will be picked up by a courier or drop off by the customer
- Field 17 is the Reference indicated in the specific Shipment (if any)

Technical Specs <https://developers.inpost.it/#tag/Labels>

4. Update Shipment status



After the Shipment Creation, there're two alternative ways to keep updated about the status of a shipment: Configured a [Webhook](#) or invoke [Shipment Tracking API](#)

Webhook is used to **align near real-time** a client every time a significant change has occurred. The async events are sent by InPost Systems that can be subscribed by a client are:

- Shipment Confirmation
- Shipment Status Changed
- Error

For configuring the Webhook, the provided service must match 3 conditions.

- correct URL structure
- The service should be accessible from the Inpost application
- The service should respond with HTTP 200

For implementing an **on-demand alignment**, the best solution can be request of the tracking via **ShipX Tracking API** with the tracking number

Tracking resource is an object representing information about the current status of the shipment in the logistics system. It can be utilized to get information about the creation, status change and status history of the shipment

The tracking number is used for retrieving information about shipment current status and all its history , included all the status changes

STANDARD TRACKING FOR H2L *confirmed → adopted_at_source_branch → ready_to_pickup → [pickup_reminder_sent →] delivered*

STANDARD TRACKING FOR L2L

confirmed → dispatched_by_sender → taken_by_courier → adopted_at_source_branch → ready_to_pickup → [pickup_reminder_sent →] delivered

STANDARD TRACKING FOR L2H *confirmed → dispatched_by_sender → taken_by_courier → adopted_at_source_branch [→ out_for_delivery] → returned_to_sender*

5. Delivery



1. When the courier insert the parcel in the locker or deliver it in a PUDO Point (Shipment in status **ready_to_pickup**) the system send a notification, email and sms, to the customer.
2. The customer will have the PIN (pickup number) and QR that can use for retire the parcel from a locker or in a PUDO
3. After n days, a reminder will be sent to the customer, and the parcel status become **pickup_reminder_sent**
4. When the customer pickup the parcel from locker or PUDO it becomes in status **delivered**

If the parcel is not picked up before the expiration time (**pickup_time_expired**) then the parcel will be sent back to the depo or to the sender according to the specific agreement (**returned_to_sender**)



InPost Integrations General Considerations

InPost SHIPX APIs Considerations

- APIs detailed description available online. [Technical Docs](#)
- For some of the SHIPX APIs is required an authentication with a TOKEN provided by Inpost.
- TOKEN is different for each environment (test and production)
 - Staging (test) Environment Shipx url** <https://stage-api-shipx-it.easypack24.net>
 - Production Environment Shipx url** <https://api-shipx-it.easypack24.net>
- The Staging area is available for testing purposes, for all the duration of the integration project



InPost

out of the box