

Reference Manual

MyDHL Express Comprehensive Shipping for Magento 2.x — Extmag_MydhIShipping

2026-05-21

- 1 Module layout
- 2 DHL Express REST API endpoints
 - 2.1 Authentication
 - 2.2 Error envelopes
- 3 Carrier registration
- 4 DHL service codes
 - 4.1 Doc services (intra-EU, document-class)
 - 4.2 Non-Doc services (international, non-document)
- 5 DI plugin map
- 6 Configuration scoping
- 7 Configuration field matrix
- 8 Cron jobs
- 9 Source models
- 10 Per-product attributes
- 11 Landed Cost data flow
 - 11.1 EstimatedDutiesStorage internals
- 12 Estimated duties checkout total
- 13 Reference token macros
- 14 Plugin sort-order interleaving
- 15 Custom admin controller
- 16 Known limitations
- 17 Diagnostics



logo

1 Module layout

Extmag_MydhLShipping is a carrier extension on top of the Extmag_Shiplab base shipping framework. It plugs DHL Express into Shiplab through DI plugins — there are very few standalone services here.

```
Extmag/MydhLShipping/
├─ Controller/Adminhtml/ShipperAccount/MyDhlSignatureUpload.php
├─ Helper/
│   └─ DataPrepare.php
│   └─ Request/ {MydhL, Rate, Ship, Track, PickupCreation, PickupCancel, LandedCost}.php
│   └─ Response/ {MydhL, Rate, Ship, Track, PickupCreation, PickupCancel, LandedCost}.php
├─ Model/
│   └─ EstimatedDutiesStorage.php
│   └─ Source/ (19 source models – see "Source models" below)
│   └─ Total/Quote/EstimatedDuties.php
├─ Plugin/
│   └─ Controller/Adminhtml/{Label,ShipperAccount/Save,ShippingMethods/Save}.php
│   └─ Helper/{Connector, Request, DataPrepare/ProductHandler}.php
│   └─ Model/{Carrier, Config, Label, ShipperAccount/DataProvider, Source/CarrierCodes}.php
│   └─ ViewModel/Order/Email/Shipment/Tracking.php
├─ Setup/Patch/Data/EAVAttributesCategory.php
├─ etc/
│   └─ adminhtml/{di,menu,routes,shiplab,shiplab_config,system}.xml
│   └─ adminhtml/system/{general,international_invoice,notification,other,payment,pickup,printing,reference,refund}.xml
│   └─ config.xml, di.xml, email_templates.xml, module.xml, sales.xml
└─ view/{adminhtml,frontend}/...
```

2 DHL Express REST API endpoints

The module talks to DHL Express through the official MyDHL Express REST API. Endpoint mapping per request type lives in `Plugin\Model\Source\CarrierCodes::beforeGetCarriers`.

Request type	Sandbox URL (Live Account = No)	Production URL (Live Account = Yes)
rate	https://express.api.dhl.com/mydhlapitest/rates	https://express.api.dhl.com/mydhlapirates
ship	https://express.api.dhl.com/mydhlapitest/shipments	https://express.api.dhl.com/mydhlapishipments
track	https://express.api.dhl.com/mydhlapitest/tracking?shipmentTrackingNumber={trackingNumber}&levelOfDetail=ALL_CHECKPOINTS	https://express.api.dhl.com/mydhlapitracking?shipmentTrackingNumber={trackingNumber}&levelOfDetail=ALL_CHECKPOINTS
pickupcreation	https://express.api.dhl.com/mydhlapitest/pickups	https://express.api.dhl.com/mydhlapipickups
pickupcancel	https://express.api.dhl.com/mydhlapitest/pickups/{id}	https://express.api.dhl.com/mydhlapipickups/{
landedcost	https://express.api.dhl.com/mydhlapitest/landed-cost	https://express.api.dhl.com/mydhlapilanded-co

URL placeholders ({id} , {trackingNumber}) are substituted by `Plugin\Helper\Connector::afterGetAPIUrl` from `request['AdditionalData']['UrlReplacements']`.

2.1 Authentication

HTTP Basic — the Authorization header is built in `Helper\Request\Mydhl::getHttpHeaders()` :

```
Authorization: Basic base64(account_user_id:account_password)
Content-Type: application/json
Accept: application/json
```

`account_user_id` = Site ID, `account_password` = Password (both from the Shipper Account record).

2.2 Error envelopes

`Helper\Response\Mydhl::parse()` recognises three error shapes and routes them to `parseFault()` :

- `reasons[]` non-empty — preferred (newest API revision).
- `title` non-empty — short error envelope.
- `status >= 400` — HTTP-style error code.

`parseFault` walks `reasons[]` first, then `additionalDetails[]`, then the top-level `detail / title`. Each error becomes a `{Code, Description}` pair. `Unauthorized / 401 / malformed JSON` shortcut to `unauthorized()`.

3 Carrier registration

Key	Value
<code>code</code>	<code>mydhl</code>
<code>name / long_name</code>	<i>MyDHL Express</i>
<code>max_weight</code>	300 KG
<code>customer_printing_type</code>	PDF
<code>data_prepare_class</code>	<code>Extmag\MydhlShipping\Helper\DataPrepare</code>
<code>request_classes</code>	<code>per rate / ship / track / pickupcreation / pickupcancel / landedcost</code>
<code>response_classes</code>	mirrors the request map
<code>service_codes_class</code>	<code>Extmag\MydhlShipping\Model\Source\ServiceCodes</code>

4 DHL service codes

Source model: `Model\Source\ServiceCodes`. Origin selection by `getOriginName()` — `Doc` when both shipper and consignee are in the EU, `Non-Doc` otherwise.

4.1 `Doc` services (intra-EU, document-class)

Code	Service
1	DOMESTIC EXPRESS 12:00
2	B2C
5	SPRINTLINE
7	EXPRESS EASY
9	EUROPACK
B	BREAKBULK EXPRESS
C	MEDICAL EXPRESS
D	EXPRESS WORLDWIDE
G	DOMESTIC ECONOMY SELECT
I	DOMESTIC EXPRESS 9:00
K	EXPRESS 9:00

Code	Service
L	EXPRESS 10:30
N	DOMESTIC EXPRESS
O	DOMESTIC EXPRESS 10:30
R	GLOBALMAIL BUSINESS
S	SAME DAY
T	EXPRESS 12:00
U	EXPRESS WORLDWIDE
W	ECONOMY SELECT
X	EXPRESS ENVELOPE

4.2 Non-Doc services (international, non-document)

Code	Service
3	B2C
4	JETLINE
8	EXPRESS EASY
E	EXPRESS 9:00
F	FREIGHT WORLDWIDE
H	ECONOMY SELECT
J	JUMBO BOX
M	EXPRESS 10:30
P	EXPRESS WORLDWIDE
Q	MEDICAL EXPRESS
V	EUROPACK
Y	EXPRESS 12:00

5 DI plugin map

etc/di.xml — every plugin extends a Shiplab type with DHL-specific behaviour. Sort orders interleave with other carrier modules using the same base.

Shiplab target	Plugin	Role	Sort
Helper\Connector	Plugin\Helper\Connector	JSON-encodes payloads for mydhl ; rewrites URL placeholders (track / pickup-cancel).	30
Helper\Request	Plugin\Helper\Request	Merges Landed-Cost duties into Rate service_codes ; DDP uplift on shipping price.	30
Model\Carrier	Plugin\Model\Carrier	quoteId on duties storage before rate collection; restore per-service duties.	30
Model\Source\CarrierCodes	Plugin\Model\Source\CarrierCodes	Registers the mydhl carrier (URLs, request/response classes, service codes).	30
Model\Config	Plugin\Model\Config	Per-carrier configuration lookups.	30

Shiplab target	Plugin	Role	Sort
Model\Label	Plugin\Model\Label	DHL label-creation hooks (autocreate, mass, manual).	40
Controller\Adminhtml\Label	Plugin\Controller\Adminhtml\Label	Admin label controller hooks (save / preview).	40
Controller\Adminhtml\ShippingMethods\Save	Plugin\Controller\Adminhtml\ShippingMethods\Save	Method save hook (extra DHL field validation).	20
ViewModel\Order>Email\Shipment\Tracking	Plugin\ViewModel\Order>Email\Shipment\Tracking	Tracking link rendering in shipment emails.	40
Helper\DataPrepare\ProductHandler	Plugin\Helper\DataPrepare\ProductHandler	DHL-specific product attribute extraction (HS code, country of origin, etc.).	30

In addition, `Extmag\MydhlShipping\Model\EstimatedDutiesStorage` is registered with `shared="true"` so all consumers see the same instance per request.

6 Configuration scoping

Configuration values live in the Shiplab configuration tree, not in `core_config_data`. Storage is decomposed across:

Table	Scope
<code>extmag_configuration</code>	Root configuration row.
<code>extmag_configuration_scope</code>	Carrier × direction (domestic / international / return).
<code>extmag_configuration_store</code>	Per-store-view.
<code>extmag_configuration_country</code>	Per-country (or per-continent group).
<code>extmag_configuration_direction</code>	Domestic / outbound / inbound direction split.

Lookups go through `Extmag\Shiplab\Model\Config` (`Plugin\Model\Config` applies DHL-specific overrides). A single field can resolve through up to four nested fallbacks before defaulting.

7 Configuration field matrix

`Stores` → `Configuration` → `Sales` → `Shipping Methods` → `MyDHL Express Configuration` exposes the per-carrier tree. The XML for each tab lives under `etc/adminhtml/system/`:

Tab / XML file	Notable fields
<code>general.xml</code>	Account (Shipper Account ref), Shipper Address, Ship From Address, Destination Type, Shipping Description.
<code>international_invoice.xml</code>	Active, Paperless Trade (PLT), Type (Commercial / Proforma), Terms of Shipment, Reason for Export, Invoice Number, Comments, Additional Declaration Text, Place of Incoterm, Sold To Address, Shipper / Consignee Business Party Type, Receiver EORI Attribute. Sub-group: Product Attributes (Description, Country of Manufacture, Unit of Measurement, Harmonized (Commodity) Code, Brand, DHL Product Category). Sub-group: Landed Cost (Active, Estimated Tariff Rate Type, Show Duties & Taxes on Checkout).
<code>notification.xml</code>	Active, Message (delivery notification to recipient).
<code>other.xml</code>	Options: Saturday Pickup, Saturday Delivery, Send Tracking Number, GoGreen Climate Neutral, Delivery Confirmation, Order Status After Created Shipping Label. Sub-group: Dangerous Goods (Product Categories with Dangerous Goods).
<code>payment.xml</code>	Currency, Shipment Payer, Duty And Tax Payer, Declared Value Type, Insurance, COD (Cash on Delivery), Allowed Payment Method for COD, Allowed Specific Payment Method for COD.
<code>pickup.xml</code>	Earliest Pickup Time, Latest Pickup Time, Special Instructions, Schedule (allowed days of week, time window), Drop-off Type.
<code>printing.xml</code>	Type (label format), Template, Label Resolution (dpi) — 200 or 300, Archive Document Template, Mass Print.

Tab / XML file	Notable fields
reference.xml	Reference ID (with macro tokens: #order_id#, #customer_name#, #store_id#, #store_code#, #store_name#).
refund.xml	Creating Return Label With Shipping Label, Allow Print, Allow Create (per customer-account).

8 Cron jobs

Inherited from Extmag\Shiplab\etc\crontab.xml. All jobs live in the Magento default cron group.

Job	Class	Default schedule	Purpose
extmag_create_shipping_label	Extmag\Shiplab\Cron\AutoCreateLabel	every minute	Queue-driven automatic label creation.
extmag_create_pickup	Extmag\Shiplab\Cron\AutoCreatePickup	hourly / end-of-day	Pickup creation (per-label or batched).
extmag_get_track	Extmag\Shiplab\Cron\AutoGetTracking	every minute	Tracking polling. Stops on delivery success or error.
extmag_clean_db_logs	Extmag\Shiplab\Cron\DbLogCleaner	daily	Trim extmag_logs retention window.
extmag_apq_purge_obsolete	Extmag\Shiplab\Cron\AccessPointQuotePurge	daily	Drop obsolete access-point quotes.

Gating: extmag_create_shipping_label and extmag_create_pickup are no-ops when the carrier's Active flag is off across all shipping methods.

9 Source models

Drop-down sources for admin configuration and label-creation forms. Located under Model\Source\ :

Class	Purpose
Accounts	Shipper accounts available for the mydhl carrier.
ArchiveDocTemplate	ARCH_8X4_A4_002, ARCH_8X4, ARCH_6X4, ARCH_A4_RU_002.
BusinessPartyType	Shipper / Consignee business party type for UK Brexit / NI invoices.
DataSet	Underlying DHL code dictionary (see below).
DeliveryConfirmation	Signature, Adult Signature, etc.
DutyPayer	Customer (DAP) / Shipper (DDP) for the duty payer split.
EstimatedTariffRateType	Tariff rate calculation mode for Landed Cost.
Generic	Base class — subclasses pick a code key from DataSet.
InvoiceType	Commercial / Proforma.
LabelFormat	PDF / ZPL / EPL — also defines template sizes.
PackagingCodes	YP (Your Packaging) and DHL-defined containers.
Payer	Sender / Recipient / Third Party.
PrintType	Output channel for the label payload.
ProductUnitOfMeasurement	PCS, KG, LB, etc. — declared on invoice line items.
RatesResponded	List of services that responded to the last rate request (for Show Methods filter).
ReasonForExport	Permanent, Repair and Return, Personal Effects, Sample, Gift, Documents, Return to Origin, etc.
ResolutionDpi	200 or 300.
ServiceCodes	DHL service codes by origin (Doc / Non-Doc) — see above.
TermsOfTrade	EXW, FCA, FAS, FOB, CPT, CIP, CFR, CIF, DAP, DPU, DDP.

10 Per-product attributes

DHL-relevant product attributes consumed by the carrier:

Attribute	Used in	Notes
weight	Rate, Ship	Defaults to Magento <code>weight</code> — configurable via <i>Packaging</i> → <i>Attribute of Product Weight</i> .
Width / Length / Height	Rate, Ship	Mapped via <i>Packaging</i> → <i>Attribute of Product Width/Length/Height</i> .
Volumetric Weight	Rate, Ship	Optional attribute when product carries volumetric weight directly.
Country of Manufacture (<code>country_of_manufacture</code>)	International invoice, Landed Cost	Required for Landed Cost to compute origin-specific tariffs.
Harmonized Code (HS Code)	International invoice, Landed Cost	Required for Landed Cost; rejected by API when missing on international shipments.
Brand	International invoice	Optional brand declaration on line item.
<code>dhldg_category</code>	Other Options → Dangerous Goods	Category EAV attribute installed by <code>EAVAttributesCategory</code> data patch.
Pre-packed (Yes / No)	Packaging	When <code>Yes</code> , product ships as its own package without box-packing.

The attribute codes are not hard-coded — every one is selected via dropdown in the carrier *Packaging / International Invoice* → *Product Attributes* configuration tabs.

11 Landed Cost data flow

The Landed Cost API runs alongside the Rate API on every international rate request to surface estimated import duties, taxes, and fees at checkout.

```
beforeCollectRates(Carrier)
  ↳ EstimatedDutiesStorage::setQuoteId(quoteId)

Rate request (per service code, batched per package set)
  ↳ response['service_codes']['mydhl']['code']['landed_cost_duty|tax|fee']

afterParseResponse(Helper\Request) [Plugin\Helper\Request]
  ↳ setDisplayConfig(shipperPays, showOnCheckout)
  ↳ if shipperPays (DDP): merge duty+tax+fee into shipping price
  ↳ else (DAP): keep service_codes as-is, duties surface as totals line

afterProcessCollectedResponses(Carrier) [Plugin\Model\Carrier]
  ↳ setDisplayConfig(...) (idempotent)
  ↳ setServiceDuties(code, duty, tax, fee) per service
    (currency-converted via Carrier::convertCurrencyByCarrier)

EstimatedDutiesStorage
  ↳ serviceDuties: per-service-code map
  ↳ aggregatedDuties: fallback aggregate
  ↳ persistToCache() (key 'mydhl_duties_<quoteId>', TTL 3600s)

Model\Total\Quote\EstimatedDuties::fetch (sales.xml: quote.totals.mydhl_estimated_duties, sort 550)
  ↳ renders Magento checkout total when shouldShow() returns true
```

11.1 EstimatedDutiesStorage internals

Singleton (`shared="true"` in `etc/di.xml`). Three responsibilities:

- Per-service duty map** — `serviceDuties[serviceCode] = ['duty', 'tax', 'fee']`. Set via `setServiceDuties()` from rate-response post-processing.
- Aggregated fallback** — `aggregatedDuties` for shipments without a service-code split (legacy path).
- Cache persistence** — every `setServiceDuties()` call writes to Magento `CacheInterface` keyed `mydhl_duties_<quoteId>` (TTL 3600 s). `hasData()` and `shouldShow()` lazy-restore from cache on demand. This guard exists because Magento checkout fires

parallel HTTP requests (estimate-shipping-methods then totals-information) and the totals collector runs in a separate request than the rate collection.

Reset on cart change: tag-based cache invalidation is left to the Magento quote lifecycle. `EstimatedDutiesStorage::reset()` is provided but not called automatically — the cache entry expires after one hour or is overwritten by the next rate request on the same quote.

12 Estimated duties checkout total

Registered in `etc/sales.xml`:

```
<section name="quote">
  <group name="totals">
    <item name="mydhl_estimated_duties"
      instance="Extmag\MydhlShipping\Model\Total\Quote\EstimatedDuties"
      sort_order="550"/>
  </group>
</section>
```

`EstimatedDuties::collect()` is a no-op (the value is computed at rate-collection time). `fetch()` returns the duty / tax / fee aggregate from `EstimatedDutiesStorage` for the current `quoteId`. Rendering on the frontend is wired through `view/frontend/web/js/view/checkout/summary/...`.

13 Reference token macros

A few admin fields accept macro tokens that expand at request time:

Field	Tokens
Reference ID (<code>reference.xml</code>)	<code>#order_id#</code> , <code>#customer_name#</code> , <code>#store_id#</code> , <code>#store_code#</code> , <code>#store_name#</code>
Shipping Description (<code>general.xml</code>)	<code>#order_id#</code> , <code>#customer_name#</code>
Invoice Number (<code>international_invoice.xml</code>)	<code>#order_id#</code> , <code>#shipment_id#</code> , <code>#invoice_id#</code>
Comments (<code>international_invoice.xml</code>)	<code>#customer_name#</code> , <code>#order_id#</code> , <code>#shipment_id#</code> , <code>#invoice_id#</code> , <code>#store_id#</code> , <code>#store_code#</code> , <code>#store_name#</code>

14 Plugin sort-order interleaving

Multiple carrier modules (`Extmag_UpsShipping`, `Extmag_FedexShipping`, `Extmag_MydhlShipping`) plug into the same `Shiplab` types. Sort orders matter because some plugins return early when `carrier !== 'mydhl'`. The DHL plugins live at sort orders 20–40 — keep that range free if you write your own override.

`Plugin\Helper\Request::afterParseResponse` returns `unchanged` when `response['carrier'] !== 'mydhl'`. The same guard pattern is used in `Plugin\Helper\Connector::beforeFormatNativeData` (case-insensitive `strtolower($carrier) === 'mydhl'`).

15 Custom admin controller

Route	Class	Pur
<code>extmag_mydhlshipping/shipperaccount/mydhlsignatureupload</code>	<code>Controller\Adminhtml\ShipperAccount\MyDhlSignatureUpload</code>	Asy upk of tl PLT sigr ima the Shij Acc form

Allowed file types: PNG, GIF, JPG. Max size: 1 MB (enforced by the upload action; DHL itself rejects payloads larger than ~1 MB).

16 Known limitations

- **DHL service unavailability** — no automatic retry. Rate failures are surfaced inline at checkout; ship / pickup failures are queued back to `extmag_create_label_cron` only for transient HTTP errors. Logical errors (`reasons[]` , `title`) are persisted as failed log rows.
- **Landed Cost requires HS codes** — products without `country_of_manufacture` or Harmonized Code make the duty quote degrade silently to the aggregated fallback. Validate at import time, not at checkout time.
- **Currency conversion** — landed-cost amounts are converted via `Carrier::convertCurrencyByCarrier()` which uses Magento's `Magento\Directory\Model\CurrencyFactory`. Make sure base + display currency exchange rates are kept fresh.
- **Cache TTL on duty storage** — duty values persist for 1 hour per quote. A buyer who edits the cart after the TTL expires sees a fresh rate request (and a possibly different duty estimate).
- **No automatic re-pickup on cancel** — `pickupcancel` removes the pickup on DHL's side but does not re-queue. Operator must re-create manually.
- **typeCode fallback** — when no Package Box is matched and no dimensions are declared, the request degrades to `typeCode = 'YP'` (Your Packaging). DHL accepts this but the rate may be less accurate than dimensions would yield.

17 Diagnostics

- `var/log/extmag.log` — full request / response audit, written by `Extmag\Shiplab\WebService\Logger`. Trim with the `extmag_clean_db_logs` cron.
- `extmag_logs` table — database-level audit for the same flow. Browseable in the admin under `Extmag → Shipping Logs → Shipping Logs`.
- `Stores → Configuration → Sales → Shipping Methods → MyDHL Express Configuration` save action validates that the chosen Shipper Account exists and that the Shipper Address has the minimum required fields.
- `Stores → Configuration → Extmag → ...` controls global behaviour shared with the base Shiplab module (db log retention, etc.).