

# Beacon API documentation

## Overview

Beacon API allows you to programmatically create and track shipments and orders with Beacon, for example updating container ETAs in your ERP or TMS. This guide details what data can be exchanged with Beacon via our API endpoints to help you implement this in your system. If you have feedback or questions please let us know at [support@beacon.com](mailto:support@beacon.com).

### [Overview](#)

#### [1. Getting started](#)

##### [1.1. Login](#)

##### [1.2. Refresh token](#)

#### [2. Ocean containers](#)

##### [2.1. Start tracking](#)

##### [2.2. Get the latest tracking data](#)

##### [2.3. Status codes & error messaging](#)

##### [2.4. Supported carrier code list](#)

#### [3. Air waybills \(AWBs\)](#)

##### [3.1 Start tracking](#)

##### [3.2 Get latest tracking data](#)

##### [3.3. Status codes & error messaging](#)

##### [3.4. Supported carrier code list](#)

#### [4. Order Management](#)

##### [4.1 Create / Update Order](#)

##### [4.2 Link Order to Shipment](#)

#### [5. Status codes & error messaging](#)

## 1. Getting started

Follow the steps below to start the implementation. If you are not a Beacon customer, please get in touch with us at [support@beacon.com](mailto:support@beacon.com) first. If you are a Beacon customer, your username and password can be used to access both the platform and the API. Your customer success manager can grant access to the API on a user level.

Our API responses use JSON API schema and our APIs can be used with any HTTP client, for example Postman. You can find the collection of requests [here](#).

**Important** Beacon API has a rate limit of 120 requests per minute for non-login calls and 20 requests per minute for login calls.

## 1.1. Login

Use the following endpoint to authenticate the user and login to the API.

**POST** <https://api.beacon.com/v1/login>

### 1.1.1. Request example

```
{
  "username": "user123@gmail.com",
  "password": "password123!"
}
```

Field	Required?	Type	Description
username	Yes	String	Beacon username
password	Yes	String	Beacon password

### 1.1.2. Response example

```
{
  "access_token": "eyJ...",
  "refresh_token": "v1...",
  "token_type": "Bearer",
  "expires_in": 300
}
```

Field	Always included?	Type	Description
access_token	Yes	String	Access token
refresh_token	Yes	String	Refresh token
token_type	Yes	String	Token type
expires_in	Yes	Int	Time to expiration in seconds

## 1.2. Refresh token

Use the following endpoint to refresh your token if it has expired.

**POST** <https://api.beacon.com/v1/login/token>

### 1.2.1. Request example

```
{
  "refresh_token": "v1..."
}
```

Field	Required?	Type	Description
refresh_token	Yes	String	Beacon token

### 1.2.2. Response example

```
{
  "access_token": "eyJ...",
  "refresh_token": "v1...",
  "token_type": "Bearer",
  "expires_in": 300
}
```

Field	Always included?	Type	Description
access_token	Yes	String	Access token
refresh_token	Yes	String	Refresh token
token_type	Yes	String	Token type
expires_in	Yes	Int	Time to expiration in seconds

## 2. Ocean containers

### 2.1. Start tracking

Use the following endpoint to start tracking **one or more containers** with Beacon.

**POST** <https://api.beacon.com/v1/containers>

Once the request is sent to Beacon, we will validate the syntax of the container number. If successful, the container will be visible in the Beacon platform and you will receive a successful response. Beacon will then attempt to fetch tracking data for the container's current journey. Once the tracking data is received, it will be available in the Beacon platform and can be fetched using the endpoint in the next section.

**Important** Please note that your contracted usage will be consumed if the request is successful and tracking data is found.

**Reminder** Please ensure you are sending **only containers that you are actively tracking** (i.e. your shipment in the container is about to be or currently in-transit). Beacon will always fetch tracking data that relates to the current journey of the container and not to journeys that have ended.

If a container sent to Beacon is a container you are no longer tracking, we will still fetch data relating to the container's current journey. If we are able to return data to you, your contracted usage will be consumed.

### 2.1.1. Request example

**Important** Custom fields must first be created in the platform before they can be utilised via the API. A customer may create up to 10 unique custom fields all of which are type String.

When sending custom field information via the POST request the name of the custom field must match the name pre-defined in the platform. For example a custom field called Reference Number should be posted as "name": "Reference Number"

```
[
  {
    "container_number": "CSQU3054383",
    "carrier_code": "MAEU",
    "destination_warehouse_name": "London Warehouse",
    "custom_fields": [
      {
        "name": "invoice_number",
        "value": "IV123",
      },
      {
        "name": "reference_number",
        "value": "123245",
      }
    ]
  }
]
```

Field	Required?	Type	Description
container_number	Yes	String	Container number
carrier_code	No	String	4 character SCAC code unique to carrier
destination_warehouse_name	No	String	Name of warehouse created in the address book
custom_fields	No	Array	Array of up to 10 custom fields pre-defined in platform. All custom fields are strings

**Important** Custom fields are case sensitive

### 2.1.2. Response example

The response will either be a successful status code, or an error status code along with a relevant error messaging shown in section 4.

## 2.2. Get the latest tracking data

**Important** We recommend scheduling your system to fetch tracking data from Beacon no more frequently than every 2 hours.

### 2.2.1. Get updates for one container

Use the following endpoint to get the latest tracking data for **one container** you have started tracking with Beacon.

**GET** `https://api.beacon.com/v1/containers/{containerNumber}`

#### 2.2.1.1. Request example

All you need to pass to Beacon is the above endpoint, with the container number included.

Field	Required?	Type	Description
containerNumber	Yes	String	Container number

#### 2.2.1.2. Response example

**Important** Note that for all date fields the value of `actual_date` and `estimated_date` is dependent on the available data. Date values are usually provided as a date and time with the relevant offset for the location where the event takes place. If no zone information is available, it may be a UTC date time ("2021-06-07T02:25:17Z"). If no time information is available, it will be a date with no time information ("2021-06-07")

```
{
  "container_number": "CSQU3054383",
  "port_of_loading": {
    "name": "Yantian",
    "un_location_code": "CNYTN"
  },
  "port_of_discharge": {
    "name": "Felixstowe",
    "un_location_code": "GBFXT"
  },
  "gated_out_empty_dates": {
    "actual_date": "2023-02-28T23:00:00+08:00"
  },
  "gated_in_full_dates": {
    "actual_date": "2023-03-01T12:00:00+08:00"
  }
}
```

```

    },
    "loaded_dates": {
      "actual_date": "2023-03-02T11:00:00+08:00"
    },
    },
    "vessel_departure_dates": {
      "estimated_date": "2023-03-03T10:00:00+08:00",
      "actual_date": "2023-03-03T10:00:00+08:00"
    },
    },
    "vessel_arrival_dates": {
      "estimated_date": "2023-03-03T10:00:00+08:00",
      "actual_date": "2023-03-22T18:00:00z"
    },
    "discharged_dates": {
      "actual_date": "2023-03-23T20:00:00z"
    },
    },
    "gated_out_full_dates": {
      "actual_date": "2023-03-24T09:00:00z"
    },
    },
    "vessel": {
      "name": "CMA CGM LYRA",
      "imo_number": "9410806"
    },
    "carrier": {
      "name": "Maersk",
      "code": "MAEU"
    },
    "status": "GATED_OUT_FUIL",
    "custom_fields": [
      {
        "name": "invoice_number",
        "value": "IV123",
      },
      {
        "name": "reference_number",
        "value": "123245",
      }
    ],
    "purchase_orders": [
      {
        "number": "123",
      },
      {
        "number": "1234",
      }
    ]
  }
}

```

Field	Subfield	Always included?	Type	Description
container_number		Yes	String	Container number
port_of_loading	name	No	String	Name of Port of Loading

	un_location_code	No	String	Location code of Port of Loading
port_of_discharge	name	No	String	Name of Port of Discharge
	un_location_code	No	String	Location code of Port of Discharge
gated_out_empty_dates	actual_date	No	String	Actual Gated Out Empty datetime with an offset
gated_in_full_dates	actual_date	No	String	Actual Gated In Full datetime with an offset
loaded_dates	actual_date	No	String	Actual Loaded datetime with an offset
vessel_departure_dates	estimated_date	No	String	ETD in offset datetime format
	actual_date	No	String	ATD in offset datetime format
vessel_arrival_dates	estimated_date	No	String	ETA in offset datetime format
	actual_date	No	String	ATA in offset datetime format
discharged_dates	actual_date	No	String	Actual Discharged datetime with an offset
	actual_date	No	String	Actual Gated Out Full datetime with an offset
vessel	name	No	String	Vessel name
	imo_number	No	String	Vessel IMO number
carrier	name	No	String	Carrier name
	code	No	String	Carrier code
status		Yes	String	Status of the container (GATED_OUT_EMPTY, GATED_IN_FULL, LOADED_AT_POL, IN_TRANSIT, ARRIVED_AT_POD, DISCHARGED_AT_POD, GATED_OUT_FULL, PROCESSING)
custom_fields		No	Array	List of custom fields
	name	No	String	Custom field name
	value	No	String	Custom field value

purchase_orders		No	Array	List of purchase order numbers
	number	No	String	Purchase order number

**Important** Custom fields are case sensitive

## 2.2.2 Get updates for more than one container

Use the following endpoint to get the latest tracking data for **more than one container** you have started tracking with Beacon.

**GET** <https://api.beacon.com/v1/containers>

### 2.2.2.1. Request example

With this endpoint you can either get all your tracked containers or select specific containers by using the following parameters.

- For a specific number of containers that have the freshest tracking data use these optional URL parameters:

```
"size": 40,
"page": 0
```

If you do not specify the number of containers to return, all containers that have previously been added to Beacon will be returned. This endpoint uses pagination - you can *limit* the number of containers returned, for example get the 40 containers with the freshest tracking data and specify the page of the container count.

- For specific containers use this URL parameter:

```
"container_numbers": ["CSQU3054383", "CSQU3023455"]
```

You can pass up to 50 `container_numbers`.

Field	Required?	Type	Description
container_numbers	No	Array [String]	Container numbers. Maximum 50 container_numbers allowed in one request.
limit	No	Int	Limit the number of results returned, maximum 50
start	No	Int	Start index of the results returned, defaults to 0



### 2.2.2.2. Response

The response body will be a list of the container objects shown in section 3.1.2.

Response header	Always included?	Type	Description
total_results	No	Int	Total number of results available
next_page	No	Int	Next page of the remaining results to return. Will be omitted if there are no more results to return.

## 2.3. Status codes & error messaging

Each successful response from Beacon will have a status code in the 2xx range, unless there was an error. Please refer to section 5 for a description of the error responses.

## 2.4. Supported carrier code list

You can find the list of ocean carriers [here](#).

# 3. Air waybills (AWBs)

## 3.1 Start tracking

Use the following endpoint to start tracking **one or more AWB's** with Beacon.

**POST** <https://api.beacon.com/v1/tracking/awbs>

Once the request is sent to Beacon, we will validate the syntax of the AWB number. If successful, the AWB will be visible in the Beacon platform.

**Important** Please note that your contracted usage will be consumed if the request is successful and tracking data is found.

### 3.1.1. Request example

```
[
  {
    "awb_number": "000-12345678"
  },
  {
    "awb_number": "000-87654321"
  }
]
```

Field	Required?	Type	Description
awb_number	Yes	String	AWB number

### 3.1.2. Response example

The response will either be a successful status code, or an error status code along with a relevant error messaging shown in section 5.3.

## 3.2 Get latest tracking data

Tracking data can be consumed using the Beacon Webhook, reach out to [support@beacon.com](mailto:support@beacon.com) for more information.

## 3.3. Status codes & error messaging

Each successful response from Beacon will have a status code in the 2xx range, unless there was an error. Please refer to section 5 for a description of the error responses.

## 3.4. Supported carrier code list

You can find the list of Air carriers [here](#).

# 4. Order Management

## 4.1 Create / Update Order

Use the following endpoint to either create or update an order with Beacon.

**POST** <https://api.beacon.com/v1/orders>

If successful, the order will be visible on the Beacon platform.

**Important** The critical piece of information inside the order data is the order number itself. This is a unique identifier that your organisation uses to identify the order. Beacon will use this identifier to determine if the order already exists. If it doesn't then a new order will be created, otherwise the existing order will be updated.

**Reminder** Please ensure you are consistent with spelling and capitalisation, especially when it comes to supplier and buyer names. Small differences could lead to separate suppliers being created which could affect your ability to search for orders.

In addition, spelling of origin and destination locations must match entries in the address book exactly, or they will be ignored.

#### 4.1.1. Request example

```
{
  "order_number": "PO12345",
  "supplier": "ABC Manufacturing",
  "buyer": "ABC Retail",
  "forwarder": "ABC Freight",
  "origin_location": "Warehouse A",
  "destination_location": "Warehouse B",
  "dates": {
    "order_placed_date": "2026-02-02",
    "order_confirmed_date": "2026-02-02",
    "cargo_ready_date": "2026-02-02",
    "delivery_by_date": "2026-02-02"
  },
  "labels": [
    "string"
  ],
  "skus": [
    {
      "sku_number": "100-2891-B",
      "description": "Fizzy Drinks (24 pack)",
      "quantity": 100,
      "unit_price": {
        "number": 99.95,
        "currency_code": "GBP"
      }
    }
  ]
}
```

Field	Required?	Type	Description
order_number	Yes	String	Order number
supplier	No	String	Supplier name, if this is a purchase order
buyer	No	String	Buyer name, if this is a sales order
forwarder	No	String	Freight forwarder name
origin_location	No	String	Origin location, must correspond to an entry in your address book
destination_location	No	String	Destination location, must correspond to an entry in your address book

dates	order_placed_date		No	Local Date YYYY-MM-DD	When the order was placed / created
	order_confirmed_date		No	Local Date YYYY-MM-DD	When the order was confirmed
	cargo_ready_date		No	Local Date YYYY-MM-DD	The cargo ready date (CRD)
	delivery_by_date		No	Local Date YYYY-MM-DD	When the order must be delivered by
labels			No	Array	List of labels to attach to the order
sku	sku_number		No	String	A unique identifier for the SKU
	description		No	String	SKU description
	quantity		No	Integer	Quantity ordered
	unit_price	amount	No	Decimal	Price per item ordered
		currency_code	No	String	Three letter currency code that the amount is in, for example “USD”, “EUR”.

#### 4.1.2. Response example

The response will either be a successful status code (202), or an error status code along with a relevant error messaging shown in section 5.

## 4.2 Link Order to Shipment

Use the following endpoint to link an order to a shipment. The order must already exist, and the order number should be part of the URL path.

**POST** [https://api.beacon.com/v1/orders/{order\\_number}/link](https://api.beacon.com/v1/orders/{order_number}/link)

If successful, the order will be visible on the Beacon platform.

**Important** - Beacon will attempt to find an existing shipment using the details you provided. However if a shipment can't be found then tracking will be started- this will consume credits, the same way you would if directly requesting tracking using one of the other API endpoints.

Parameters:

Field	Required?	Type	Description
order_number (in URL path)	Yes	String	This is the unique order number you provided when originally creating the order. For example "PO-123". This parameter needs to be URL-encoded to prevent issues with the path.
shipment_type	Yes	String	One of the following: CONTAINER, AWB or BOL
tracking_number	Yes	String	The tracking number to identify the shipment. This will depend on the shipment_type you specified: <ul style="list-style-type: none"> <li>• If CONTAINER - this will be the 11 character container number</li> <li>• If BOL - this will be the bill of lading number provided by the carrier</li> <li>• If AWB - this will be the AWB number</li> </ul>
carrier_code	No	String	For ocean shipments only, this is the SCAC code for the carrier. It is required when the shipment_type is BOL, and optional otherwise.

#### 4.2.1. Request example - Ocean container

```
POST https://api.beacon.com/v1/orders/PO-123/link
{
  "shipment_type": "CONTAINER",
  "tracking_number": "AAAU1234567",
  "carrier_code": "MSCU"
}
```

#### 4.2.2. Request example - Ocean bill of lading

```
POST https://api.beacon.com/v1/orders/PO-123/link
{
  "shipment_type": "BOL",
  "tracking_number": "M-193827611231",
  "carrier_code": "MAEU"
}
```

#### 4.2.3. Request example - Air Waybill (AWB)

```
POST https://api.beacon.com/v1/orders/PO-123/link
{
  "shipment_type": "AWB",
  "tracking_number": "000-12345678"
}
```

#### 4.2.4. Response example

The response will either be a successful status code (202), or an error status code along with a relevant error messaging shown in section 5.

**Important** Please be aware of one specific error code 402 (Payment Required) which is used to indicate you do not have sufficient credits to track a new shipment.

## 5. Status codes & error messaging

Each response from Beacon will have a status code detailed below. If there is an error, the following error messaging will be provided.

Below is an example response for a validation error from one of the endpoints.

```
{
  "status_code": 400,
  "timestamp": "2023-02-20T17:35:35.916Z",
  "error": "Validation error",
  "sub_errors": [
    {
      "object": "ContainerResource",
      "field": "container_number",
      "error": "Must not be null"
    }
  ]
}
```

Field	Subfield	Always included?	Type	Description
status_code		Yes	Int	Success = 200, Bad request = 400, Unauthorised = 401, Forbidden = 403, Rate limit exceeded = 429
timestamp		No	String	Timestamp of error
error		No	String	Main error message
sub_errors		No	Array	A list of more descriptive errors
	object	Yes	String	The name of the object responsible for an error
	field	No	String	A field responsible for an error
	rejectedValue	No	String	The field value which caused an error (non-null values only)
	error	Yes	String	Sub error message