Marketplace Documentation

Aug 14, 2019

Contents

CHAPTER 1

Markets

On the CDON web site, there is a concept known as *markets*, which roughly translates to which country the customer is visiting from, except in the case of b2b_se where it means that the product can be sold on our B2B-site. The identifiers for these markets, e.g. "se", follows the ISO 3166-1 alpha-2 country code definition. However, currently there are only five supported markets:

- se, Sweden
- b2b_se, B2B Sweden
- dk, Denmark
- no, Norway
- fi, Finland

The Marketplace APIs inherit this concept, which is why some product and order data is different between markets.

In some APIs, there is also a "default" market used as a fallback value. However, localized content is always preferred as it provides a better user experience.

CHAPTER 2

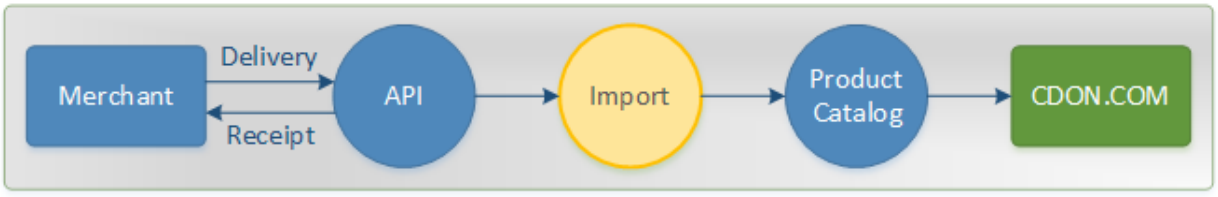
Frequently Asked Questions

CHAPTER 3

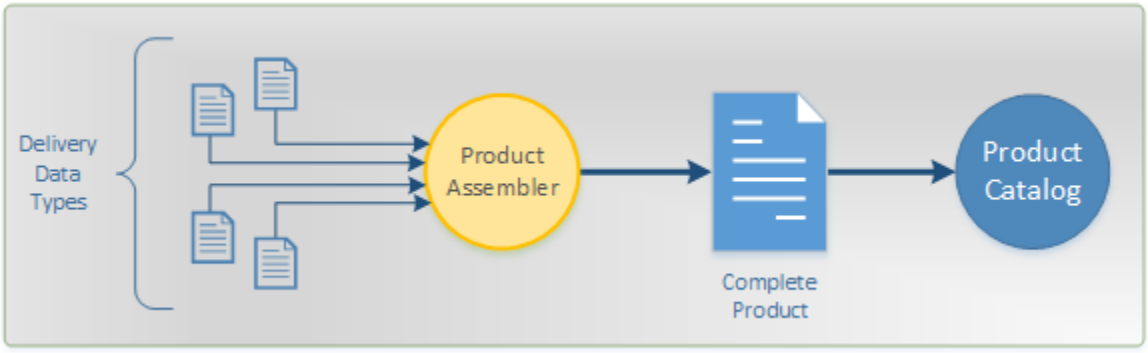
Overview

By exporting your inventory to CDON Marketplace, your products will be added to the CDON product catalog, which makes them available to CDON's customers.

In the CDON Marketplace import process, there are two vital key concepts; **data types** and **receipts**. Deliveries are made by posting different *types of data*, which populate a product with all necessary information. The *receipt* is used to keep track on the progress of the import.

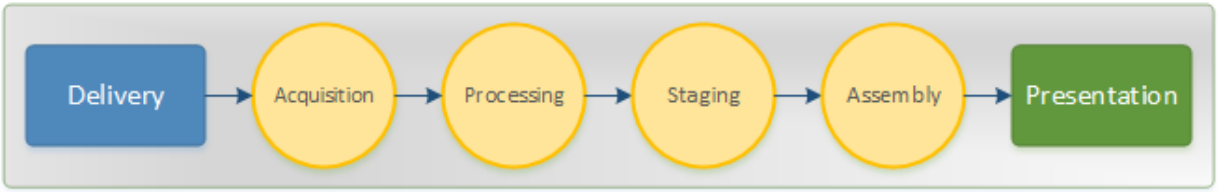


Each delivery is imported independently. When data of all types is available, the final product is assembled and added to the product catalog for presentation.

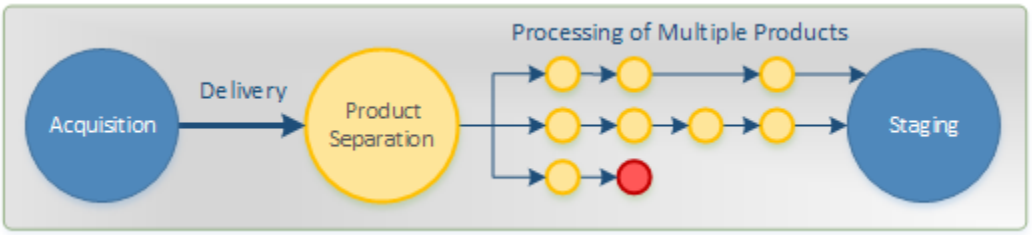


3.1 The Import Process

Data in a delivery is passed through a series of sequential steps which make out the import process. During this process, the data is validated and may be modified according to the current business rules.



Deliveries containing multiple products are split up into individual products, which in turn are imported independently from each other.



Products that fail validation are naturally not imported, but will not abort the entire delivery. Only products whose content has changed (since the last import) will be updated — unmodified data will be discarded early in the process.

> **Caution:** An inventory import is **greedy**, which means that any and all products that **can** be imported **will** be imported.

3.2 The Product

The concept of a *product* is manifested by two types:

Simple Product A single, indivisible product, i.e. a "regular" product.

Product with Variations Multiple products that are considered "the same", but varies on a specific property.

It is important to understand the distinction between these. A variation product is e.g. a beverage with different flavor variants, or a t-shirt in different sizes, etc. Correctly defining a product makes it easier for the customer to find and buy the product.

3.2.1 Variation Products

If the product has variations, the "parent" product is called the **model product**.

The model product is the common denominator which attributes shared properties to the variation products and creates the context which relates all variations to each other.

It is not possible to buy the model product, but it invites the customer to decide on one of its variations.

3.3 Support

CDON Marketplace has its own customer service that gladly helps you with Marketplace related cases.

When contacting support, please provide the following information when available:

- Merchant ID
- Receipt ID (see *receipts*)

Those details will help giving a more accurate diagnosis with technical matters.

CHAPTER 4

Receipts

Importing products to CDON Marketplace is an asynchronous process, meaning that it is not instantaneous. When you deliver data to the API, a **receipt** will be issued as soon as the data has been pre-validated and accepted.

Example of a receipt:

```
00b24f3a93124da7aec34447124e5aa1
```

The receipt is a unique 32-character long string associated with a **single** delivery.

When a delivery attempt to the *API* responds with a receipt, that means the data has been received and that the import will start as soon as possible. Since the actual import may take a long time (depending of the size and current workload), the receipt is intended to be used to inquire about the current status of that particular import at a later moment.

Important: A receipt in the delivery response does **not** mean that any products have been imported — the receipt is only an acknowledgement that the delivery has been received and accepted.

The asynchronous process has several benefits:

- **A response is generated as soon as the data is received.** There is no need to wait for the entire import to complete.
- **The import is more error resilient.** Errors and warnings will be tracked and handled to have as small impact as possible on the rest of the import.
- **Multiple imports can be enqueued in parallel.** The four different types of data can be imported simultaneously.
- **Detailed per-product results can be retrieved.** Each failed product can be examined and diagnosed in detail, and amended individually without affecting any other part of the import process.

Important: Take note that the asynchronous import also implies that there will **not** be any notification when the import has completed. Thus, it is important that you save the receipt for future use.

CHAPTER 5

Data Types

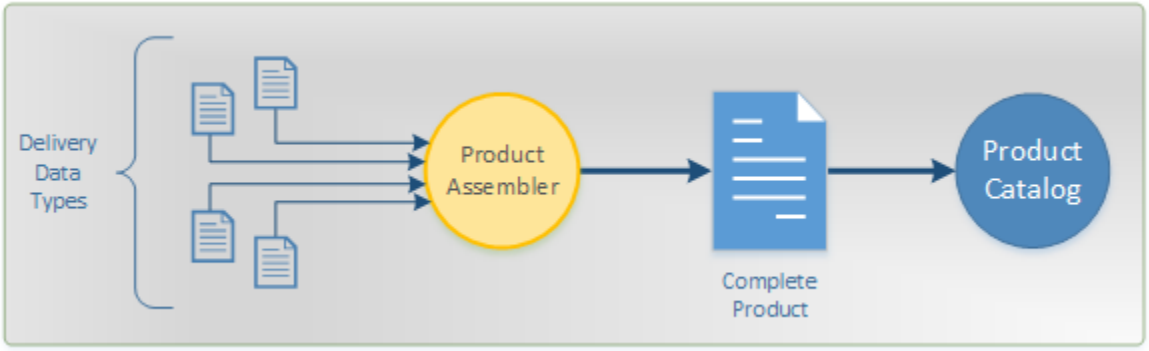
In order to present a complete and buyable product to CDON's customers, four distinct types of data are required:

- *Product Description*
- *Price*
- *Availability*
- *Media*

These four types of data are fully independent from each other in the context of importing the data. In most cases, they also have different life cycles. Depending on your business, some data is shorter lived and require more frequent updates than other data. This enables a more flexible delivery with faster imports.

Although the data is separate from each other, *Product Description*, *Price* and *Availability* are constituents necessary to form a complete product. If any of those three types is missing, the product will be **unavailable** until all product data requirements have been met. When all data is available, it will be compiled into a final product.

Media is not necessary to complete a product.



5.1 Data Contract

To facilitate a structured delivery of data, each set of data must fulfill a **data contract** that specify the structure of the data, expected data types and mandatory fields. If this contract is not met, the import process is unable to accept the data.

The contracts are made publicly available at this location:

https://schemas.cdon.com/product/4.0/

These contracts can with advantage be used as an acceptance test on the delivery before transmitting it to CDON Marketplace.

Important: Not all business rules are enforced by the contract. As the contract is primarily a data interchange format, higher order business rules are not validated until processing begins.

CHAPTER 6

Product Identities

The product identifier is the key that correlates all product data from different types.

All four *data types* specify the product key by using the element `id`.

6.1 Model Product and Variations

The *model product* (see *The Product*) must always have its own **unique** key, regardless of whether it has any variations or not.

If the product has any variations, each variation is obligated to have its own unique product key.

A variation product cannot be redefined to later map to a different model product.

> **Attention:** No check will be done to identify duplicate (and possibly conflicting) product identities. It is the merchant's responsibility to maintain data consistency in regards of product identities.

For the product data types *availability*, *media* and *price*, data associated with the model product's key is discarded if the product has variations.

6.2 The Product Data Type

The *product data type* has encapsulated the `<id>` element inside an `<identity>`-element.

```
<identity>
  <id>main_product_key</id>
  <gtin>global_trade_item_number</gtin>
  <mpn>manufacturer_part_number</mpn>
  <sku>stock_keeping_unit</sku>
</identity>
```

The `id`-element is always **mandatory**. The other elements, `gtin`, `mpn` and `sku` are product data that is **not** used for identification during processing, but are persisted if provided.

> **Attention:** Except for the `id`-element, the identifiers are primarily optional. However, certain categories require one or more of them, which will be enforced during processing.

6.3 Case Insensitive

Note that product id:s are **not case sensitive**! For example, the following id:s are all considered equal:

```
Product_A
product_a
PRODUCT_A
pRoDuCt_A
```

6.4 Valid Identity Characters

A valid product key is 1–40 characters long, and only accepts certain characters.

The following characters are allowed in an identity token:

- a-z
- A-Z
- 0-9
- / (slash)
- - (hyphen)
- _ (underscore)

If specified, the element `gtin` must be exactly either 8, 12, 13 or 14 digits long. For reference, see gtin.info.

The restrictions are more lenient for the elements `mpn` and `sku`. If specified, 1–50 characters of any kind is permitted.

CHAPTER 7

Validation

The pre-validation performed immediately on the import request only validates that the data is acceptable in the sense that it is compliant with the delivery contract. No evaluation of the contents is done at this stage.

After the data has been accepted, one of the stages is the business validation. During this process data may be discarded or modified according to the current business rules. Any products that violate a business rule will be rejected, and the details concerning this event is retrievable using the receipt.

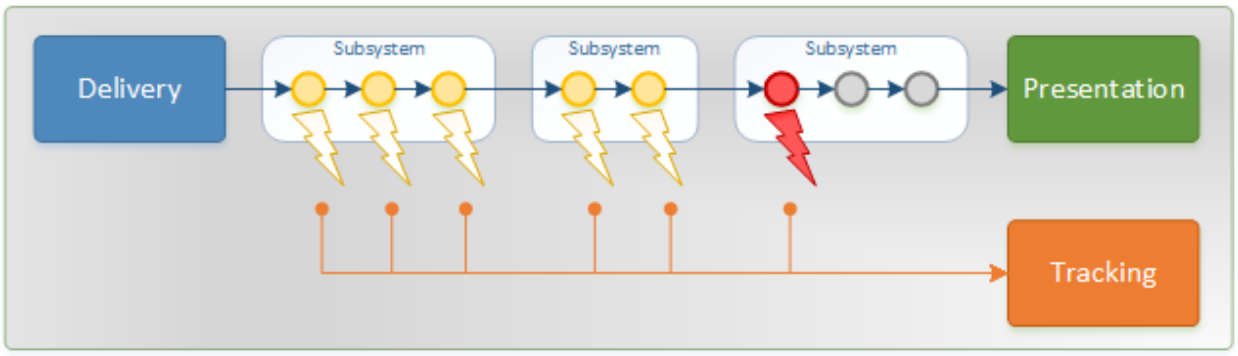
This kind of delayed validation offers two important benefits:

- Each product is evaluated individually. This allows for a complete inventory validation, as opposed to an import that halts on the first error.
- Detailed validation messages allow for self-help. Failed products have already been identified and can be amended by you for re-import.

CHAPTER 8

Tracking

Products in any delivery are processed individually as they journey through the import process. As the product passes certain milestones during the import, an entry is recorded in what is called the "tracking API". As the name suggests, these entries can be used to track the progress of the import (and individual products).



8.1 API

The tracking application can be accessed using a web API. To access the API you must supply an authorization header to identify yourself. Use the same authorization header key as when you submit the products.

The tracking API has 3 endpoints:

8.1.1 Overview

```
https://mis.cdon.com/deliveries
```

Get the status of your last 100 deliveries. (Latest first) The response will be a JSON file with a list of the deliveries:

```
[
    {
        "receiptId": "08d71ef5fe115a0800155d4af3d60000",
        "startTime": "2019-08-12T07:23:39.491834+00:00",
        "endTime": "2019-08-12T07:23:40.5525303+00:00",
        "endPoint": "Product",
        "status": "Failed",
        "errorMessage": "1 product(s) failed",
        "total": 1,
        "totalPending": 0,
        "totalSucceeded": 0,
        "totalFailed": 1
    },
    {
        "receiptId": "08d71ef3a1e2149000155d4af3d60000",
        "startTime": "2019-08-12T07:06:45.8434386+00:00",
        "endTime": null,
        "endPoint": "Product",
        "status": "Pending",
        "errorMessage": null,
        "total": 1,
        "totalPending": 1,
        "totalSucceeded": 0,
        "totalFailed": 0
    }
]
```


It is also possible to specify how many records you want to return by adding the querystring "take":

```
https://mis.cdon.com/deliveries?take=10
```

The maximum supported value is 1000.

8.1.2 Delivery status

```
https://mis.cdon.com/deliveries/<ReceiptId>
```

Get the status of a specific delivery using the ReceiptId. The response will be a JSON file with the status of that delivery:


```
{
    "receiptId": "08d71ef5fe115a0800155d4af3d60000",
    "startTime": "2019-08-12T07:23:39.491834+00:00",
    "endTime": "2019-08-12T07:23:40.5525303+00:00",
    "endPoint": "Product",
    "status": "Failed",
    "errorMessage": "1 product(s) failed",
    "total": 1,
    "totalPending": 0,
    "totalSucceeded": 0,
    "totalFailed": 1
}
```

8.1.3 Product failures

```
https://mis.cdon.com/deliveries/<ReceiptId>/failures
```

If a delivery has one or more failed products, details of these failures can be viewed using this endpoint. The response will be a JSON file with a list of the failed products of the specified delivery:

```
[
    {
        "productId": "1087760",
        "errorMessage": "Missing required GTIN."
    },
    {
        "productId": "1087761",
        "errorMessage": "Missing required GTIN."
    }
]
```

CHAPTER 9

API

Deliveries are made to Marketplace by posting **XML**-formatted data to the endpoint for the corresponding *data type*.

The base URL to the product import-API is:

```
https://mis.cdon.com/<data_type>
```

Responses are returned with a HTTP status code and possibly a body in **JSON**-format.

Each delivery that is acceptable (according to the data contract) receives a *receipt*. Make sure to store this, as it is the key to *tracking* the progress of the delivery.

9.1 Authentication

For security reasons, all HTTP requests to the API must include a `Authorization` header.

The value of the header shall be the API token, prefixed by the word `api` and a whitespace. Example:

```
Authorization: api 5875ca6c-229d-4f4c-a45f-4252b4583538
```

9.2 Data Contracts

As mentioned in *validation*, a pre-validation is performed on the delivery. This validation is based on a data contract specified in a set of XSD (XML Schema Definition) files.

The location of the schema files is:

https://schemas.cdon.com/product/4.0/

Use these files to generate XML and validate against them to avoid submitting invalid data.

9.2.1 Product Data Type

The product data type is the information that describes the product and specifies feature values. Some of this data will be published to customers while other data is metadata for controlling product behavior once published.

To reduce the quantity of imported data, products whose content has not changed since the last successful import will be discarded automatically.

The URL to post product data to is:

```
https://mis.cdon.com/product
```

Product Definition

A vital concept with this data is that it represents a **complete product definition**. Each product must be delivered in its entirety, meaning there cannot be any partial product updates. The product data must thus, in each delivery, contain all data to fully describe it, lest the product will be redefined with a lesser definition.

In practice, this means that, for example, it is not possible to update only e.g. the title of a product. That would obliterate all category- and variation related data.

Data Contract

The relevant schema files are the following:

- product.xsd
- types.xsd
- categories.xsd
- attributes.xsd
- variants.xsd

The last four files are supplements to the product.xsd, which define the overall product data structure.

Category

The category defines what attributes that describe the product.

A product must belong to exactly one category.

Variants

Some products may exist in different variants. Valid variables are:

- Size
- Color
- Flavor
- Size and Color combined
- Size and Flavor combined

Within the same product, all variants must be of the same variable(s), meaning a single product cannot have one variant that varies on e.g. *flavor*, and another variant that varies on e.g. *size and color*.

Product Size and Weight

The `Dimensions` property of a product describe the product's size and weight.

Spatial Dimensions

Abbreviation	Unit	Comment
um	micrometre	
µm	micrometre	The same as 'um', but with the greek letter.
mm	millimetre	
cm	centimetre	
dm	decimetre	
m	metre	
km	kilometre	
in	inch	0.0254 metres
ft	foot	0.3048 metres
yd	yard	0.9144 metres

Mass Dimensions

Abbreviation	Unit	Comment
mcg	microgram	
µg	microgram	The same as 'mcg', but with the greek letter.
mg	milligram	
g	gram	
hg	hectogram	
kg	kilogram	
t	metric tonne	Equivalent to a 1000 kilograms
gr	grain	0.06479891 grams
oz	ounce	28.349523125 grams
lb	pound	453.59237 grams

Energy Classification

Energy `Energy` property of a product describe the energy classification. The energy property consists of four parts specified as tags.

The energy classification property is optional, however to determine which products that **should** include energy classifications, or to read further regarding energy classification, please visit https://europa.eu/youreurope/business/environment/energy-labels/index_en.htm.

Energy Properties

Property name	Data type	Description
class	energyClass	Energy class enumeration
label	URL	Energy label
arrow	URL	Energy arrow
sheet	URL	Energy sheet (aka Fische)

Energy Classes Enumeration

Accepted values for `energyClass` is:

- APlusPlusPlus
- APlusPlus
- APlus
- A
- B
- C
- D
- E
- F
- G

Sanitization

Textual data that will be published to end-users will be sanitized. HTML- and other scripting code as well as foul language will be removed.

9.2.2 Price Data Type

The price data type specifies pricing details for a product. Every product must have price information for every market it will be sold in. This also includes VAT and shipping rates.

This data type is incremental, which means that any data that can be applied will be applied without altering related data. Hence, it is possible to import just the updates as soon as they are available.

The URL to post price data to is:

```
https://mis.cdon.com/price
```

Data Contract

The relevant schema files are the following:

- price.xsd
- types.xsd

The last file is a supplement to the price.xsd, which define the price data structure.

Validation Rules

The sale price may not be greater than the original price.

Price

The product price consists of two price values; **original** and **sale** price.

- The `OriginalPrice` represents the list price.
- The `SalePrice` represents the price that the product is up for sales for.

Please note that the prices must include VAT.

VAT

The `Vat` allows different values depending on VAT rates in the market where the products are being sold. The value is expressed as a percentage.

Example:

```
12
```

which represents a VAT of 12%. The VAT amount must be valid for the VAT identification number used in the desired market.

Shipping Costs

The `ShippingCost` allow the follwing values:

Sweden	Denmark	Norway	Finland
0	0	0	0
19	19	19	1.95
29	29	39	2.95
39	39	49	3.95
49	49	59	4.90
59	59	79	4.95
79	79	99	5.95
99	99	399	7.95
199	199	799	9.95
395	495	995	59.00
495	N/A	N/A	79.00

9.2.3 Availability Data Type

The availability data type specifies details regarding stock, delivery times, etc.

Every product must specify the availability status for each market, as well as the expected delivery time range. The intention is to give the customer an as accurate estimation as possible of when the product will be delivered.

Attention: Products with the status `offline` will **not** be buyable.

This data type is incremental, which means that any data that can be applied will be applied without altering related data. Hence, it is possible to import just the updates as soon as they are available.

The URL to post availability data to is:

```
https://mis.cdon.com/availability
```

Data Contract

The relevant schema files are the following:

- availability.xsd
- types.xsd

The last file is a supplement to the availability.xsd, which define the availability data structure.

Validation Rules

The delivery time is a range in number of days, where the minimum value may not be greater than the maximum value.

Stock

The `Stock` value represents the number of products to be imported to CDON:s product catalog.

Regarding services, digital products or other products with **unlimited stock value**, the imported value has to be very high (but not higher than a signed 32-bit integer) so that the product do not run out of stock due to no stock updates.

Will The Product Be Buyable?

There are three parameters that control whether a product can be purchased by a customer:

- Status
- Release Date
- Stock

As mentioned above, the `Status` will ultimately decide whether the product is presented to customers or not. If it is set to `Offline`, it will not be made available for purchase regardless of the other parameters.

If `Stock` is greater than zero and `Release Date` is in the past, the product is **buyable**. If `Stock` is zero and `Release Date` is in the past, the product is not buyable, but the customer can sign up for notification when the product becomes available for purchase. If `Release Date` is in the future, the product will be **bookable**.

Note that `Release Date` is a part of the *Product data contract*.

9.2.4 Media Data Type

The media data type associate pictures and videos with a product.

The URL to post media data to is:

```
https://mis.cdon.com/media
```

Data Contract

The relevant schema files are the following:

- media.xsd
- types.xsd

The last file is a supplement to the media.xsd, which define the media data structure.

Product Variations

Product elements in the product data type support variations of the same product (e.g. different colors of a product). In the media import, it is possible to assign different media to both the main product and/or the variation products.

Product ID

The media will, of course, be imported to the specified product ID.

If media is added to the main product, this media will be presented when the product page is first displayed to the user. The media added to the variation products will presented when the user selects the variation.

Multi-variations

In cases where a product variant varies on two attributes at the same time, certain rules apply.

- Size and Color. **Color** is the predominant variable.
- Size and Flavor. **Size** is the predominant variable.

This means that the predominant variable's media will be presented to the user.

Example 1:

A shirt varies on **size** (e.g. S, M or L) and **color** (e.g. red or green). When the user has selected the size "M", the picture displayed will not change until the user selects another color.

Example 2:

A beverage varies on **size** (e.g. 2 dl, 5 dl or 1 liter) and **flavor** (e.g. banana or strawberry). When the user has selected the flavor "banana", the picture displayed will not change until the user selects another size.

Validation Rules

Every product must have exactly one main picture.

Extra pictures and/or videos are optional.

CHAPTER 10

Overview

When a customer buys something on CDON the order is saved, and sent to your specific Marketplace account (merchant account). It is your responsibility to handle the orders that have been delivered to your Merchant account. This can be done by either using the Marketplace Admin or by using the API Integration. There are greater benefits by integrating as your own employees will have fewer systems to work with but it requires an integration.

The Marketplace support team can aid you if you have specific questions. We also have integration partners that have built modules for quite a few e-commerce platforms.

10.1 Endpoints

The base URL for the imports is as follows:

```
https://admin.marketplace.cdon.com/api/
```

Append the name of the data type like so:

```
https://admin.marketplace.cdon.com/api/orderdelivery
```

10.2 Error Codes

Marketplace order API returns reponse codes to indicate what failed in your specific call to the order API. Unsuccessful responses (HTTP status code 400) return details about the error as an response object containing the following information:

Key	Value Type	Value Description
ErrorCode	integer	An internal status code that represents the error.
Message	string	A short description of the error.

The response body looks like this:

```
{
  "ErrorCode": 1000,
  "Message": "Order does not exist"
}
```

The list of `internal` response codes is as follows:

ErrorCode	Description
1000	The order was not found.
1001	The order row was not found.
1002	Order row overflow.
1003	Negative order row price.
1004	The order already exists.
1005	The package carrier does not exist.
1006	The order row is not in a valid state for the command.
1008	The same order row occurs multiple times in the same command.
1009	No rows were specified in the command.
2001	Invoice row overflow.
2002	No rows to invoice.
3001	Value of returned products is not high enough to charge for NPU.
3002	At least one order row must be marked as returned to charge for NPU.
3003	Can not charge for NPU more than once.
3004	Returned order row must have been sent with package id.
3005	There is no NPU charge to retract.

CHAPTER 11

API

11.1 Fetch Order

Order API provides you the ability to perform different actions from your own ERP system or perhaps your ecommerce software, allowing the flexibility to completely integrate it in your preferred system.

11.1.1 Request Example

GET api/order/{id}

```
GET https://admin.marketplace.cdon.com/api/order/111111 HTTP/1.1
Accept: application/json
Authorization: api <apiKey>
```

11.1.2 Response Example - JSON

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.

```
{
  "OrderDetails": {
          "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
          "OrderId": 42277358,
          "State": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
          "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "CountryCode": "Sweden",
          "CurrencyCode": "SEK",
          "TotalAmount": 1495.0,
```

(continues on next page)

(continued from previous page)


```
          "TotalAmountExcludingVat": 1196.0,
          "TotalSalesAmount": 1495.0,
          "CustomerInfo": {
            "CustomerId": 62880501,
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "Phones": {
              "PhoneMobile": "0703013319",
              "PhoneWork": null,
              "PhoneHome": null
            }
          },
          "OrderRows": [
          {
            "OrderRowId": 1,
            "FulfillmentStatus": "Invoiced",
            "PaymentStatus": "AwaitingPayment",
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "DeliveredQuantity": 1,
            "InvoicedQuantity": 1,
            "CancelledQuantity": 0,
            "ReturnedQuantity": 0,
            "PickedQuantity": null,
            "PricePerUnit": 1495.0,
            "OrdinaryPricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "PackageId": "test",
            "DebitedAmount": 1495.0,
            "CreditedAmount": 0.0,
            "PaidAmount": 0.0,
            "RefundedAmount": 0.0,
            "AddonToProductId": null
          }
],
          "InvoiceNumbers": [
            "1000052"
          ],
            "TotalVat": 299.0
```


(continues on next page)

(continued from previous page)


```
        },
        "invoices": [
        {
          "Rows": [
        {
          "TotalPaymentAmount": 0.0,
          "TotalCreditNoteAmount": 0.0,
          "Status": "AwaitingPayment",
          "InvoiceRowNumber": 1,
          "OrderRowId": 1,
          "ProductId": "ART000494",
          "ProductName": "Star wars",
          "ProductType": "Article",
          "Quantity": 1,
          "PricePerUnit": 1495.0,
          "VatPerUnit": 299.0,
          "VatPercentage": 25.0000,
          "TotalAmount": 1495.0,
          "TotalVat": 299.0
        }
        ],
        "Status": "AwaitingPayment",
        "Payments": null,
        "InvoiceNumber": "1000052",
        "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
        "OrderId": 42277358,
        "CustomerId": 62880501,
        "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
        "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
        "TotalAmount": 1495.0,
        "TotalVat": 299.0,
        "CurrencyCode": "SEK"
        }
  ]
}
```


11.1.3 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.

Continued on next page

Table 1 – continued from previous page

Variable	Type	Description
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.

Continued on next page

Table 1 – continued from previous page

Variable	Type	Description
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.2 Fetch Pending Orders

Retrieve a list of orders by providing filter paramaters.

11.2.1 Request Example

GET api/order/{parameters}

```
GET https://admin.marketplace.cdon.com/api/order/?CountryCode=Denmark&
↪DateTimeRangeMin=2017-02-23&DateTimeRangeMax=2017-03-10 HTTP/1.1
Accept: application/json
Authorization: api <apiKey>
```

11.2.2 Response Example - JSON

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.


```
{
  "OrderDetails": {
          "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
          "OrderId": 42277358,
          "State": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
          "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "CountryCode": "Sweden",
          "CurrencyCode": "SEK",
          "TotalAmount": 1495.0,
          "TotalAmountExcludingVat": 1196.0,
          "TotalSalesAmount": 1495.0,
          "CustomerInfo": {
            "CustomerId": 62880501,
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "Phones": {
              "PhoneMobile": "0703013319",
              "PhoneWork": null,
              "PhoneHome": null
            }
          },
          "OrderRows": [
          {
            "OrderRowId": 1,
            "FulfillmentStatus": "Invoiced",
            "PaymentStatus": "AwaitingPayment",
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "DeliveredQuantity": 1,
            "InvoicedQuantity": 1,
            "CancelledQuantity": 0,
            "ReturnedQuantity": 0,
            "PickedQuantity": null,
```


(continues on next page)

(continued from previous page)


```
            "PricePerUnit": 1495.0,
            "OrdinaryPricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "PackageId": "test",
            "DebitedAmount": 1495.0,
            "CreditedAmount": 0.0,
            "PaidAmount": 0.0,
            "RefundedAmount": 0.0,
            "AddonToProductId": null
          }
],
          "InvoiceNumbers": [
          ],
            "TotalVat": 299.0
          },
          "invoices": [
  ]
}
```


11.2.3 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.

Continued on next page

Table 2 – continued from previous page

Variable	Type	Description
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.

Continued on next page

Table 2 – continued from previous page

Variable	Type	Description
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.3 Order Delivery

Order delivery is supposed to be called when the order or specific item has been sent to the customer. Hence this call marks specified orderrows as delivered. When all orderrows have been delivered an invoice is automatically created. Keep in mind that if there is a delivery fee associated with the order this will show up as a separate orderrow with the productid 'Postage'. This must also be set as delivered for the order to be considered fully delivered.

11.3.1 Request Example - JSON

```
{
  "OrderId": 1,
  "Products": [
        {
          "OrderRowId": 1,
          "QuantityToDeliver": 2,
          "PackageId": "sample string 3",
          "PackageCarrierId": 1
        },
        {
          "OrderRowId": 1,
          "QuantityToDeliver": 2,
          "PackageId": "sample string 3",
          "PackageCarrierId": 1
        }
  ]
}
```

11.3.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes
OrderRowId	integer	Refers to the order row associated to a specific order.	Yes
QuantityToDeliver	integer	Indicates how many products you intend to set as delivered for the specific order row.	Yes
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.	No
PackageCarrierId	integer	The id of the carrier used to ship the package (e.g. Posten, DHL).	No

Attention: An updated list of available packagecarriers and their id can be retrieved by performing a GET request to api/packagecarrier.

11.3.3 Response Example - json:

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.


```
{
  "OrderDetails": {
          "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
          "OrderId": 42277358,
          "State": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
          "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "CountryCode": "Sweden",
          "CurrencyCode": "SEK",
          "TotalAmount": 1495.0,
          "TotalAmountExcludingVat": 1196.0,
          "TotalSalesAmount": 1495.0,
          "CustomerInfo": {
            "CustomerId": 62880501,
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
```


(continues on next page)

(continued from previous page)


```
            "ZipCode": "12345",
            "City": "Ankeborg",
            "Country": "SE"
          },
          "Phones": {
            "PhoneMobile": "0703013319",
            "PhoneWork": null,
            "PhoneHome": null
          }
        },
        "OrderRows": [
        {
          "OrderRowId": 1,
          "FulfillmentStatus": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "ProductId": "ART000494",
          "ProductName": "Star wars",
          "ProductType": "Article",
          "Quantity": 1,
          "DeliveredQuantity": 1,
          "InvoicedQuantity": 1,
          "CancelledQuantity": 0,
          "ReturnedQuantity": 0,
          "PickedQuantity": null,
          "PricePerUnit": 1495.0,
          "OrdinaryPricePerUnit": 1495.0,
          "VatPerUnit": 299.0,
          "VatPercentage": 25.0000,
          "PackageId": "test",
          "DebitedAmount": 1495.0,
          "CreditedAmount": 0.0,
          "PaidAmount": 0.0,
          "RefundedAmount": 0.0,
          "AddonToProductId": null
        }
],
        "InvoiceNumbers": [
          "1000052"
        ],
          "TotalVat": 299.0
        },
        "invoices": [
        {
          "Rows": [
        {
          "TotalPaymentAmount": 0.0,
          "TotalCreditNoteAmount": 0.0,
          "Status": "AwaitingPayment",
          "InvoiceRowNumber": 1,
          "OrderRowId": 1,
          "ProductId": "ART000494",
          "ProductName": "Star wars",
          "ProductType": "Article",
          "Quantity": 1,
          "PricePerUnit": 1495.0,
          "VatPerUnit": 299.0,
          "VatPercentage": 25.0000,
```


(continues on next page)

(continued from previous page)


```
            "TotalAmount": 1495.0,
            "TotalVat": 299.0
          }
          ],
          "Status": "AwaitingPayment",
          "Payments": null,
          "InvoiceNumber": "1000052",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "OrderId": 42277358,
          "CustomerId": 62880501,
          "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
          "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
          "TotalAmount": 1495.0,
          "TotalVat": 299.0,
          "CurrencyCode": "SEK"
          }
  ]
}
```


11.3.4 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.

Continued on next page

Table 3 – continued from previous page

Variable	Type	Description
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.

Continued on next page

Table 3 – continued from previous page

Variable	Type	Description
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.4 Order Return

Order return is used to signify customers return of an item. When a return is noted a credit note is created to (depending on what and how many of the products have been returned) repay the customers money.

11.4.1 Request Example - JSON

```
{
  "OrderId": 1,
  "Products": [
        {
          "OrderRowId": 1,
          "QuantityToReturn": 2
        },
        {
          "OrderRowId": 2,
          "QuantityToReturn": 2
        }
  ]
}
```

11.4.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes
OrderRowId	integer	Refers to the order row associated to a specific order.	Yes
QuiantityToReturn	integer	Indicates how many products you intend to set as returned for the specific order row.	Yes

11.4.3 Response Example - json:

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.


```
{
  "OrderDetails": {
          "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
          "OrderId": 42277358,
          "State": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
          "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "CountryCode": "Sweden",
          "CurrencyCode": "SEK",
          "TotalAmount": 1495.0,
          "TotalAmountExcludingVat": 1196.0,
          "TotalSalesAmount": 1495.0,
          "CustomerInfo": {
            "CustomerId": 62880501,
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "Phones": {
              "PhoneMobile": "0703013319",
              "PhoneWork": null,
              "PhoneHome": null
            }
          },
```


(continues on next page)

(continued from previous page)


```
          "OrderRows": [
          {
            "OrderRowId": 1,
            "FulfillmentStatus": "Invoiced",
            "PaymentStatus": "AwaitingPayment",
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "DeliveredQuantity": 1,
            "InvoicedQuantity": 1,
            "CancelledQuantity": 0,
            "ReturnedQuantity": 0,
            "PickedQuantity": null,
            "PricePerUnit": 1495.0,
            "OrdinaryPricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "PackageId": "test",
            "DebitedAmount": 1495.0,
            "CreditedAmount": 0.0,
            "PaidAmount": 0.0,
            "RefundedAmount": 0.0,
            "AddonToProductId": null
          }
],
          "InvoiceNumbers": [
            "1000052"
          ],
            "TotalVat": 299.0
          },
          "invoices": [
          {
            "Rows": [
          {
            "TotalPaymentAmount": 0.0,
            "TotalCreditNoteAmount": 0.0,
            "Status": "AwaitingPayment",
            "InvoiceRowNumber": 1,
            "OrderRowId": 1,
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "PricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "TotalAmount": 1495.0,
            "TotalVat": 299.0
          }
          ],
          "Status": "AwaitingPayment",
          "Payments": null,
          "InvoiceNumber": "1000052",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "OrderId": 42277358,
          "CustomerId": 62880501,
```


(continues on next page)

(continued from previous page)

```
            "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
            "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
            "TotalAmount": 1495.0,
            "TotalVat": 299.0,
            "CurrencyCode": "SEK"
            }
    ]
}
```

11.4.4 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.

Continued on next page

Table 4 – continued from previous page

Variable	Type	Description
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.

Continued on next page

Table 4 – continued from previous page

Variable	Type	Description
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.5 Order Cancel

Order cancel is supposed to be called when the order or specific item cannot be delivered. The reason for failed delivery could be out of stock or customers active choice to no longer desire the item.

11.5.1 Request Example - JSON

```
{
  "OrderId": 1,
  "Rows": [
        {
          "OrderRowId": 1,
          "QuantityToCancel": 2
        },
        {
          "OrderRowId": 2,
          "QuantityToDeliver": 2
        }
  ]
}
```

11.5.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes
OrderRowId	integer	Refers to the order row associated to a specific order.	Yes
QuantityToCancel	integer	Indicates how many products you intend to set as cancelled for the specific order row.	Yes

11.5.3 Response Example - json:

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.

```
{
  "OrderDetails": {
          "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
          "OrderId": 42277358,
          "State": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
          "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "CountryCode": "Sweden",
          "CurrencyCode": "SEK",
          "TotalAmount": 1495.0,
          "TotalAmountExcludingVat": 1196.0,
          "TotalSalesAmount": 1495.0,
          "CustomerInfo": {
            "CustomerId": 62880501,
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "Phones": {
              "PhoneMobile": "0703013319",
              "PhoneWork": null,
              "PhoneHome": null
            }
          },
          "OrderRows": [
          {
            "OrderRowId": 1,
            "FulfillmentStatus": "Invoiced",
            "PaymentStatus": "AwaitingPayment",
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "DeliveredQuantity": 1,
            "InvoicedQuantity": 1,
            "CancelledQuantity": 0,
            "ReturnedQuantity": 0,
            "PickedQuantity": null,
            "PricePerUnit": 1495.0,
            "OrdinaryPricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
```


(continues on next page)

(continued from previous page)


```
            "PackageId": "test",
            "DebitedAmount": 1495.0,
            "CreditedAmount": 0.0,
            "PaidAmount": 0.0,
            "RefundedAmount": 0.0,
            "AddonToProductId": null
          }
],
          "InvoiceNumbers": [
            "1000052"
          ],
            "TotalVat": 299.0
          },
          "invoices": [
          {
            "Rows": [
          {
            "TotalPaymentAmount": 0.0,
            "TotalCreditNoteAmount": 0.0,
            "Status": "AwaitingPayment",
            "InvoiceRowNumber": 1,
            "OrderRowId": 1,
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "PricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "TotalAmount": 1495.0,
            "TotalVat": 299.0
          }
          ],
          "Status": "AwaitingPayment",
          "Payments": null,
          "InvoiceNumber": "1000052",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "OrderId": 42277358,
          "CustomerId": 62880501,
          "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
          "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
          "TotalAmount": 1495.0,
          "TotalVat": 299.0,
          "CurrencyCode": "SEK"
          }
  ]
}
```


11.5.4 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.

Continued on next page

Table 5 – continued from previous page

Variable	Type	Description
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity

Continued on next page

Table 5 – continued from previous page

Variable	Type	Description
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.6 Order Write-down

Order write-down is used to give compensation to customers. Reasons may vary and each Merchant responsible for their reasons to grant compensation.

11.6.1 Request Example - JSON

```
{
  "OrderId": 1,
  "Rows": [
        {
          "OrderRowId": 1,
          "QuantityToWriteDown": 2,
          "WriteDownAmountPerItem": 3.0
        },
        {
          "OrderRowId": 2,
          "QuantityToWriteDown": 2,
          "WriteDownAmountPerItem": 3.0
        }
  ]
}
```

11.6.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes
OrderRowId	integer	Refers to the order row associated to a specific order.	Yes
QuantityToWriteDown	integer	Indicates how many products you intend to write-down for the specific order row.	Yes
WriteDownAmountPerItem	decimal	The amount you write down per item.	Yes

11.6.3 Response Example - json:

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.

```
{
  "OrderDetails": {
        "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
        "OrderId": 42277358,
        "State": "Invoiced",
        "PaymentStatus": "AwaitingPayment",
        "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
        "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
        "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
        "CountryCode": "Sweden",
        "CurrencyCode": "SEK",
        "TotalAmount": 1495.0,
        "TotalAmountExcludingVat": 1196.0,
        "TotalSalesAmount": 1495.0,
        "CustomerInfo": {
          "CustomerId": 62880501,
```

(continues on next page)

(continued from previous page)


```
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "Phones": {
              "PhoneMobile": "0703013319",
              "PhoneWork": null,
              "PhoneHome": null
            }
          },
          "OrderRows": [
          {
            "OrderRowId": 1,
            "FulfillmentStatus": "Invoiced",
            "PaymentStatus": "AwaitingPayment",
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "DeliveredQuantity": 1,
            "InvoicedQuantity": 1,
            "CancelledQuantity": 0,
            "ReturnedQuantity": 0,
            "PickedQuantity": null,
            "PricePerUnit": 1495.0,
            "OrdinaryPricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "PackageId": "test",
            "DebitedAmount": 1495.0,
            "CreditedAmount": 0.0,
            "PaidAmount": 0.0,
            "RefundedAmount": 0.0,
            "AddonToProductId": null
          }
],
          "InvoiceNumbers": [
            "1000052"
          ],
            "TotalVat": 299.0
          },
          "invoices": [
          {
            "Rows": [
```


(continues on next page)

(continued from previous page)


```
          {
            "TotalPaymentAmount": 0.0,
            "TotalCreditNoteAmount": 0.0,
            "Status": "AwaitingPayment",
            "InvoiceRowNumber": 1,
            "OrderRowId": 1,
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "PricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "TotalAmount": 1495.0,
            "TotalVat": 299.0
          }
          ],
          "Status": "AwaitingPayment",
          "Payments": null,
          "InvoiceNumber": "1000052",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "OrderId": 42277358,
          "CustomerId": 62880501,
          "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
          "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
          "TotalAmount": 1495.0,
          "TotalVat": 299.0,
          "CurrencyCode": "SEK"
          }
  ]
}
```


11.6.4 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.

Continued on next page

Table 6 – continued from previous page

Variable	Type	Description
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.

Continued on next page

Table 6 – continued from previous page

Variable	Type	Description
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.7 Order Invoice

Order invoice is created by default when the whole order has been delivered. In some rare occasions you may want to create an invoice after only delivering (sending to the customer) one out of two ordered products. A reason for this could be that item two will require a longer time before it can be delivered. In this scenario you can issue a call to create an invoice for every order row that is delivered (and is not already invoiced).

11.7.1 Request Example - JSON

```
{
  "OrderId": 1
}
```

11.7.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes

11.7.3 Response Example - json:

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.


```
{
  "OrderDetails": {
          "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
          "OrderId": 42277358,
          "State": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
          "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "CountryCode": "Sweden",
          "CurrencyCode": "SEK",
          "TotalAmount": 1495.0,
          "TotalAmountExcludingVat": 1196.0,
          "TotalSalesAmount": 1495.0,
          "CustomerInfo": {
            "CustomerId": 62880501,
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "Phones": {
              "PhoneMobile": "0703013319",
              "PhoneWork": null,
              "PhoneHome": null
            }
          },
          "OrderRows": [
          {
            "OrderRowId": 1,
            "FulfillmentStatus": "Invoiced",
            "PaymentStatus": "AwaitingPayment",
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "DeliveredQuantity": 1,
            "InvoicedQuantity": 1,
            "CancelledQuantity": 0,
            "ReturnedQuantity": 0,
            "PickedQuantity": null,
```


(continues on next page)

(continued from previous page)


```
            "PricePerUnit": 1495.0,
            "OrdinaryPricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "PackageId": "test",
            "DebitedAmount": 1495.0,
            "CreditedAmount": 0.0,
            "PaidAmount": 0.0,
            "RefundedAmount": 0.0,
            "AddonToProductId": null
          }
],
          "InvoiceNumbers": [
            "1000052"
          ],
            "TotalVat": 299.0
          },
          "invoices": [
          {
            "Rows": [
          {
            "TotalPaymentAmount": 0.0,
            "TotalCreditNoteAmount": 0.0,
            "Status": "AwaitingPayment",
            "InvoiceRowNumber": 1,
            "OrderRowId": 1,
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "PricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "TotalAmount": 1495.0,
            "TotalVat": 299.0
          }
          ],
          "Status": "AwaitingPayment",
          "Payments": null,
          "InvoiceNumber": "1000052",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "OrderId": 42277358,
          "CustomerId": 62880501,
          "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
          "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
          "TotalAmount": 1495.0,
          "TotalVat": 299.0,
          "CurrencyCode": "SEK"
          }
  ]
}
```


11.7.4 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity

Continued on next page

Table 7 – continued from previous page

Variable	Type	Description
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.8 Order Package

Order package can be used to update an orders package information. This acts as an additional feature if you need to update or could not provide the information during the api/orderdelivery call. This will notify the customer with what

package carrier and what package id the order is to be delivered with. This information will reach the customer in the form of a delivery email being sent out. Note that the order row must be in a state where this command is valid (e.g. delivered or invoiced).

11.8.1 Request Example - JSON

```
{
  "OrderId": 1,
  "Rows": [
        {
          "OrderRowId": 1,
          "PackageId": "sample string 3",
          "PackageCarrierId": 1
        },
        {
          "OrderRowId": 1,
          "PackageId": "sample string 3",
          "PackageCarrierId": 1
        }
  ]
}
```

11.8.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes
OrderRowId	integer	Refers to the order row associated to a specific order.	Yes
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.	Yes
PackageCarrierId	integer	The id of the carrier used to ship the package (e.g. Posten, DHL).	Yes

Attention: An updated list of available packagecarriers and their id can be retrieved by performing a GET request to api/packagecarrier.

11.8.3 Response Example - json:

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.

```
{
  "OrderDetails": {
        "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
        "OrderId": 42277358,
        "State": "Invoiced",
        "PaymentStatus": "AwaitingPayment",
```

(continues on next page)

(continued from previous page)


```
        "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
        "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
        "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
        "CountryCode": "Sweden",
        "CurrencyCode": "SEK",
        "TotalAmount": 1495.0,
        "TotalAmountExcludingVat": 1196.0,
        "TotalSalesAmount": 1495.0,
        "CustomerInfo": {
          "CustomerId": 62880501,
          "EmailAddress": "",
          "ShippingAddress": {
            "Name": "Testperson",
            "StreetAddress": "Stårgatan 1xa",
            "CoAddress": "",
            "ZipCode": "12345",
            "City": "Ankeborg",
            "Country": "SE"
          },
          "BillingAddress": {
            "Name": "Testperson",
            "StreetAddress": "Stårgatan 1xa",
            "CoAddress": "",
            "ZipCode": "12345",
            "City": "Ankeborg",
            "Country": "SE"
          },
          "Phones": {
            "PhoneMobile": "0703013319",
            "PhoneWork": null,
            "PhoneHome": null
          }
        },
        "OrderRows": [
        {
          "OrderRowId": 1,
          "FulfillmentStatus": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "ProductId": "ART000494",
          "ProductName": "Star wars",
          "ProductType": "Article",
          "Quantity": 1,
          "DeliveredQuantity": 1,
          "InvoicedQuantity": 1,
          "CancelledQuantity": 0,
          "ReturnedQuantity": 0,
          "PickedQuantity": null,
          "PricePerUnit": 1495.0,
          "OrdinaryPricePerUnit": 1495.0,
          "VatPerUnit": 299.0,
          "VatPercentage": 25.0000,
          "PackageId": "test",
          "DebitedAmount": 1495.0,
          "CreditedAmount": 0.0,
          "PaidAmount": 0.0,
          "RefundedAmount": 0.0,
          "AddonToProductId": null
```


(continues on next page)

(continued from previous page)


```
            }
],
            "InvoiceNumbers": [
              "1000052"
            ],
              "TotalVat": 299.0
            },
            "invoices": [
            {
              "Rows": [
            {
              "TotalPaymentAmount": 0.0,
              "TotalCreditNoteAmount": 0.0,
              "Status": "AwaitingPayment",
              "InvoiceRowNumber": 1,
              "OrderRowId": 1,
              "ProductId": "ART000494",
              "ProductName": "Star wars",
              "ProductType": "Article",
              "Quantity": 1,
              "PricePerUnit": 1495.0,
              "VatPerUnit": 299.0,
              "VatPercentage": 25.0000,
              "TotalAmount": 1495.0,
              "TotalVat": 299.0
            }
            ],
            "Status": "AwaitingPayment",
            "Payments": null,
            "InvoiceNumber": "1000052",
            "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
            "OrderId": 42277358,
            "CustomerId": 62880501,
            "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
            "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
            "TotalAmount": 1495.0,
            "TotalVat": 299.0,
            "CurrencyCode": "SEK"
            }
  ]
}
```


11.8.4 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4

Continued on next page

Table 8 – continued from previous page

Variable	Type	Description
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.

Continued on next page

Table 8 – continued from previous page

Variable	Type	Description
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

11.9 Order Delivery Note

Delivery Note API provides you the ability to generate Delivery Note report for specified orders.

11.9.1 Request Example - JSON

```
[
 {
  "OrderId": 1,
```

(continues on next page)

(continued from previous page)


```
  "AddressId": "04e02c6f-ca99-4903-8ee9-692a28044111",
  "DeliveryNoteRows": [
  {
          "ProductId": "sample string 1",
          "ProductName": "sample string 2",
          "Quantity": 3,
          "PickingLocation": "sample string 4"
  },
  {
          "ProductId": "sample string 1",
          "ProductName": "sample string 2",
          "Quantity": 3,
          "PickingLocation": "sample string 4"
  }
]
 },
 {
  "OrderId": 1,
  "AddressId": "04e02c6f-ca99-4903-8ee9-692a28044111",
  "DeliveryNoteRows": [
  {
          "ProductId": "sample string 1",
          "ProductName": "sample string 2",
          "Quantity": 3,
          "PickingLocation": "sample string 4"
  },
  {
          "ProductId": "sample string 1",
          "ProductName": "sample string 2",
          "Quantity": 3,
          "PickingLocation": "sample string 4"
  }
 ]
 }
]
```


11.9.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes
AddressId	guid	Refers to the return address associated to a specific order.	Yes
DeliveryNoteRows	IEnumerable{DeliveryNoteRow}	Indicates DeliveryNoteRow list.	Yes

Variable	Type	Description	Required
ProductId	integer	An id which refers to the product and store in the Marketplace.	Yes
ProductName	string	Refers to the product name associated to a specific product.	Yes
Quantity	integer	Indicates how many products will be delivered.	Yes

11.9.3 Response

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a binary stream of delivery note PDF document.

11.9.4 Code Example - C#

```
public bool GetDeliveryNote()
{
  var parameters = BuildRequestParameter();
  var apiKey = "bbbbbbb7-bb99-999b-bbb7-bbbbbbbbbbbb";
  var baseUri = new Uri("https://admin.marketplace.cdon.com/");

  var client = new HttpClient(new HttpClientHandler()) { BaseAddress = baseUri };
  client.DefaultRequestHeaders.Add("Authorization", "api " + apiKey);
  var response = client.PostAsync("/api/deliverynote/",
                                  new StringContent(JsonConvert.
↪SerializeObject(request), Encoding.UTF8, "application/json")).Result;
  if (response.IsSuccessStatusCode)
  {
          var disposition = response.Content.Headers.ContentDisposition.ToString();
          var filename = disposition.Substring(disposition.IndexOf('=') + 1);
          var stream = response.Content.ReadAsStreamAsync().Result;

          var filepath = Path.Combine("d:\\Downloads", filename);
          using (var fileStream = File.Create(filepath))
          {
                  stream.Seek(0, SeekOrigin.Begin);
                  stream.CopyTo(fileStream);
          }
}
}

private static IEnumerable BuildRequestParameter()
{
  var result = new List
  {
          new DeliveryNoteModel
          {
                  AddressId = new Guid("6778a353-f685-40a1-ac5b-f9694fb85ac3"),
                  OrderId = 242842680,
                  DeliveryNoteRows = new List(new[]
                  {
                          new DeliveryNoteRow
                          {
                                  ProductId = "1111111111",
                                  ProductName = "Product Name 1",
                                  Quantity = 2
                          }
                  })
          },
          new DeliveryNoteModel
          {
                  AddressId = new Guid("6eff7efc-b4cd-47db-b933-02c47631742f"),
                  OrderId = 41989521,
                  DeliveryNoteRows = new List(new[]
```

(continues on next page)

(continued from previous page)

```
                {
                        new DeliveryNoteRow
                        {
                                ProductId = "2222222222",
                                ProductName = "Product Name 2",
                                Quantity = 1
                        }
                })
        },
        new DeliveryNoteModel
        {
                AddressId = new Guid("6eff7efc-b4cd-47db-b933-02c47631742f"),
                OrderId = 41358099,
                DeliveryNoteRows = new List(new[]
                {
                        new DeliveryNoteRow
                        {
                                ProductId = "3333333333",
                                ProductName = "Product Name 3",
                                Quantity = 1
                        }
                })
        }
  };
  return result;
}
```

11.10 Order Return Address

To use the Delivery Note API you first need to register a return address in the Marketplace admin. Thereafter you can fetch your address ids and use them in the delivery note API call.

11.10.1 Request Example

```
GET https://admin.marketplace.cdon.com/api/returnaddress HTTP/1.1
Accept: application/json
Authorization: api <apiKey>
```

11.10.2 Response Example - JSON

```
{
  "DisplayName": "Display name",
  "AddressId": "8f8355a0-768f-45cb-b877-aeb5d127c766",
  "StreetAddress": "SomeStreet 3",
  "PostalCode": "34450",
  "City": "Malmö",
  "Country": "Sweden",
  "COAddress": null,
  "BoxAddress": null,
}
```

11.10.3 Response Attributes

Variable	Type	Description
DisplayName	string	Display name for the return address.
AddressId	guid	An id which refers to your return address.
StreetAddress	string	The street address.
PostalCode	string	The street postal code your return address resides in.
City	string	The city your return address resides in.
Country	string	The country your return address resides in.
COAddress	string	CO address of you return address.
BoxAddress	string	Box address of you return address.

11.11 Order Picking

Order picking can be used to set an order row to a state where a cancel operation is not allowed. This can be used for example in a situation where you want to mark rows as impossible to cancel for now (e.g. when the rows have been sent to the warehouse and it is too late to cancel). Note that the order row must be in the state 'pending' for this command to be valid.

11.11.1 Request Example - JSON

```
{
  "OrderId": 1,
  "Rows": [
        {
          "OrderRowId": 1,
          "QuantityToPick": 2
        },
        {
          "OrderRowId": 1,
          "QuantityToPick": 2
        }
  ]
}
```

11.11.2 Request Attributes

Variable	Type	Description	Required
OrderId	integer	An id which refers to your order and store in the Marketplace.	Yes
OrderRowId	integer	Refers to the order row associated to a specific order.	Yes
QuantityToPick	integer	Indicates how many products you intend to set as picked for the specific order row.	Yes

11.11.3 Response Example - JSON

This request returns an http status code, indicating how the call went, where the desired result is OK (200), including a comprehensive list of order details and invoice information.


```
{
  "OrderDetails": {
          "OrderKey": "c6840daf-6163-45ef-adce-7f5e8d8f2afe-42277358",
          "OrderId": 42277358,
          "State": "Invoiced",
          "PaymentStatus": "AwaitingPayment",
          "CreatedDateUtc": "2014-02-07T19:22:48.5942457",
          "LastModifiedDateUtc": "2014-02-07T19:22:48.5942457",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "CountryCode": "Sweden",
          "CurrencyCode": "SEK",
          "TotalAmount": 1495.0,
          "TotalAmountExcludingVat": 1196.0,
          "TotalSalesAmount": 1495.0,
          "CustomerInfo": {
            "CustomerId": 62880501,
            "EmailAddress": "",
            "ShippingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "BillingAddress": {
              "Name": "Testperson",
              "StreetAddress": "Stårgatan 1xa",
              "CoAddress": "",
              "ZipCode": "12345",
              "City": "Ankeborg",
              "Country": "SE"
            },
            "Phones": {
              "PhoneMobile": "0703013319",
              "PhoneWork": null,
              "PhoneHome": null
            }
          },
          "OrderRows": [
          {
            "OrderRowId": 1,
            "FulfillmentStatus": "Invoiced",
            "PaymentStatus": "AwaitingPayment",
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "DeliveredQuantity": 1,
            "InvoicedQuantity": 1,
            "CancelledQuantity": 0,
            "ReturnedQuantity": 0,
            "PickedQuantity": null,
```


(continues on next page)

(continued from previous page)


```
            "PricePerUnit": 1495.0,
            "OrdinaryPricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "PackageId": "test",
            "DebitedAmount": 1495.0,
            "CreditedAmount": 0.0,
            "PaidAmount": 0.0,
            "RefundedAmount": 0.0,
            "AddonToProductId": null
          }
],
          "InvoiceNumbers": [
            "1000052"
          ],
            "TotalVat": 299.0
          },
          "invoices": [
          {
            "Rows": [
          {
            "TotalPaymentAmount": 0.0,
            "TotalCreditNoteAmount": 0.0,
            "Status": "AwaitingPayment",
            "InvoiceRowNumber": 1,
            "OrderRowId": 1,
            "ProductId": "ART000494",
            "ProductName": "Star wars",
            "ProductType": "Article",
            "Quantity": 1,
            "PricePerUnit": 1495.0,
            "VatPerUnit": 299.0,
            "VatPercentage": 25.0000,
            "TotalAmount": 1495.0,
            "TotalVat": 299.0
          }
          ],
          "Status": "AwaitingPayment",
          "Payments": null,
          "InvoiceNumber": "1000052",
          "MerchantId": "3b1addb2-2b6f-49bc-a185-2b5cfb445d66",
          "OrderId": 42277358,
          "CustomerId": 62880501,
          "CreatedDateUtc": "2014-02-07T12:29:12.8663761Z",
          "BookingDateUtc": "2014-02-07T12:29:12.8663761Z",
          "TotalAmount": 1495.0,
          "TotalVat": 299.0,
          "CurrencyCode": "SEK"
          }
  ]
}
```


11.11.4 Response Attributes

Variable	Type	Description
OrderKey	string	Your unique order identifier. Composition of merchant id and order id.
OrderId	integer	An id which refers to your order and store in the Marketplace.
FulfillmentStatus/State	enum	**Indicates the state of the order or order row. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
PaymentStatus	enum	**Indicates the state of the payment. Available states are:** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4
CreatedDateUtc	datetime	The date and time the order was placed on CDON.
MerchantId	string	Your unique merchant identifier.
CountryCode	string	Country of the order, indicating in what channel the order was placed.
CurrencyCode	string	Currency code for the order.
TotalAmount	decimal	The total amount of the order. Including VAT.
TotalAmountExcludingVat	decimal	The total amount excluding VAT.
TotalSalesAmount	decimal	The total amout of the order including VAT and other fees.
CustomerId	integer	A customer's unique identifier
EmailAddress	string	Hidden field.
Name	string	Customers name. May include surname.
StreetAddress	string	Customer's street address. Applies to Shipping- and Billing address.
CoAddress	string	Customer's in care of address. Applies to Shipping- and Billing address.
ZipCode	string	Customer´s zip code.
City	string	Customer´s city.
Country	string	Customer´s country.
PhoneMobile	string	Customer's mobile phone number.
PhoneWork	string	Customer's work phone number.
PhoneHome	string	Customer's home phone number.
OrderRowId	integer	Refers to the order row associated to a specific order.
ProductId	string	Merchants own unique product identifier.
AddonToProductId	string	Indicates that this product is an add-on to different product in the order.
ProductName	string	Merchants product title.
ProductType	enum	**Indicated the type of the product. Available types are:** Article = 0 Service = 1 Postage = 2 Fee = 3 Compensation = 4
Quantity	integer | Indicates the total quantity ordered for a specific product.	
DeliveredQuantity	integer	Indicates the delivered quantity. May not exceed quantity.
InvoicedQuantity	integer	Indicates the invoiced quantity. May not exceed quantity.
CancelledQuantity	integer	Indicates the cancelled quantity. May not exceed quantity

Continued on next page

Table 9 – continued from previous page

Variable	Type	Description
ReturnedQuantity	integer	Indicates the returned quantity. May not exceed quantity.
PickedQuantity	integer(null)	Indicates the picked quantity. May not exceed quantity. Can be null.
PricePerUnit	decimal	Sales price for the product.
OrdinaryPricePerUnit	decimal	Ordinary price for the product. If the sales price is lower this will be seen as a discount and will be displayed as such on CDON.
VatPerUnit	decimal	VAT for the product.
VatPercentage	string	VAT as percentage for the product.
PackageId	string	Allows the customer to track the deliver. Also included in the delivery mail sent to the customer.
DebitedAmount	decimal	The amount the customer needs to pay associated to an invoice.
CreditedAmount	decimal	The amount that gets refunded to the customer associated to an invoice.
PaidAmount	decimal	The amount that has already been paid.
RefundedAmount	decimal	The refunded amount in case of return.
InvoiceNumber	string	The invoice number associated with the order and delivery.
TotalVat	decimal	The total order VAT.
TotalPaymentAmount	decimal	The total amount the customer needs to pay.
TotalCreditNoteAmount	decimal	The total amount that needs to be refunded to the customer.
InvoiceRowNumber	string	Refers to the invoice number associated to a specific order.
BookingDateUtc	datetime	Invoice booking date. The date the debt is booked.

CHAPTER 12

Overview

The reports API is used to generate detailed reports about inventory, orders and payment information. It supports many different filtering options so that a merchant is able to get exactly the information that is relevant to him at the time.

CHAPTER 13

API

13.1 Fetch Order Report Type

Gets a list of available report types.

13.1.1 Response Attributes

Variable	Type	Description
ReportId	guid	The unique id of the report type.
DisplayName	string	The name of the report type.
Description	string	A description of the report type.

13.1.2 Response Example - JSON

```
[
 {
  "ReportId": "00000000-0000-0000-0000-000000000001",
  "DisplayName": "The name of the report",
  "Description": "Description for the report"
 }
 {
  "ReportId": "00000000-0000-0000-0000-000000000002",
  "DisplayName": "The name of another report",
  "Description": "Description for another report"
 }
]
```

The ReportId is used when calling the other report APIs.

13.1.3 Code Example C#

```
public string Get(string path)
{
  var httpClient = new HttpClient() { BaseAddress = new Uri("https://admin.
↪marketplace.cdon.com/") };
  httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("api
↪", ApiKey);
  var response = httpClient.GetAsync(path).Result;
  response.EnsureSuccessStatusCode();

  return response.Content.ReadAsStringAsync().Result;
}
```

13.2 Fetch Filter for Report Type

GET api/reports/{reportId}
Gets the available parameters for the report with the specifed reportId.

13.2.1 Response Example - JSON

```
{
 "Parameters": [
 {
  "Type": "term",
  "Metadata": {
         "Label": "Country",
         "PostFieldName": "CountryCodes"
  },
  "Values": [
  {
         "Term": "Sweden",
         "DisplayName": "Sweden",
         "Selected": false,
         "Enabled": true,
         "FilteredCount": 100
  },
  {
         "Term": "Denmark",
         "DisplayName": "Denmark",
         "Selected": false,
         "Enabled": true,
         "FilteredCount": 50
  }
 ]
 },
 {
  "Type": "term",
  "Metadata": {
         "Label": "State",
         "PostFieldName": "States"
  },
  "Values": [
```

(continues on next page)

(continued from previous page)


```
  {
          "Term": "0",
          "DisplayName": "Pending",
          "Selected": false,
          "Enabled": true,
          "FilteredCount": 10
  },
  {
          "Term": "2",
          "DisplayName": "Cancelled",
          "Selected": false,
          "Enabled": true,
          "FilteredCount": 10
  },
  {
          "Term": "3",
          "DisplayName": "Returned",
          "Selected": false,
          "Enabled": true,
          "FilteredCount": 30
  },
  {
          "Term": "4",
          "DisplayName": "Invoiced",
          "Selected": false,
          "Enabled": true,
          "FilteredCount": 100
  }
  ]
 }
 ],
 "Formats": [
 {
   "DisplayName": "Excel",
   "Key": "excel"
 },
 {
   "DisplayName": "JSON",
   "Key": "json"
 },
 {
   "DisplayName": "XML",
   "Key": "xml"
 }
 ]
}
```


The “Parameters”-array contains several the types of parameters available and values that can be used to filter on this parameter. The “Formats”-array contains the available formats that the report can be generated in. For the Parameters the PostFieldName is the key for the filter and Term is the value of the filter.

In the above example (which is for the order report api) we see that we can filter the report to contain Swedish or Danish orders and also on the state of the order (Invoiced, Returned or Cancelled). We also see how many orders there are that match the different filter values. For example, there are 100 Swedish orders and 30 orders that have Returned as the state. In the Formats-section we see that the available formats for the report is Excel, JSON and XML. For more information on how to use this information to generate a report see the documentation for POST api/reports.

13.2.2 Code Example - C#

Below you can find a method that calls the GET api to get the details of how to generate a specific report. The method takes the path to the API, i.e. /api/reports, and the reportId of the report that you wish to get parameter details for as arguments.

```
public string Get(Guid repordId, string path)
{
  var httpClient = new HttpClient() { BaseAddress = new Uri("https://admin.
↪marketplace.cdon.com/") };
  httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("api
↪", ApiKey);
  var response = httpClient.GetAsync(path + repordId).Result;
  response.EnsureSuccessStatusCode();

  return response.Content.ReadAsStringAsync().Result;
}
```

13.3 Fetch Report

In order to generate a report you perform a POST call to the reports API with the parameters you wish to use for the report. The call must also include the fields ReportId and Format for the type of report you wish to generate and the format of the report.

Let's say for example that you wish to create a report of all Swedish orders that are either returned or cancelled and you want the result as JSON.

From the call to GET api/reports you know that the ReportId of order report is "d4ea173d-bfbc-48f5-b121-60f1a5d35a34".

From the call to GET api/reports/d4ea173d-bfbc-48f5-b121-60f1a5d35a34 you know that to filter on Swedish orders you set the CountryCodes attribute to "Sweden" and to get returned and cancelled orders you set the States attribute to 2 and 3. So in the end the filter would look like this:

```
{
  "CountryCodes": [ "Sweden" ],
  "States": ["2", "3"]
}
```

You then post the parameters as form data (content-type: application/x-www-form-urlencoded) so the request body would look like this:

```
ReportId=d4ea173d-bfbc-48f5-b121-60f1a5d35a34&format=json&filter={"CountryCodes":[
↪"Sweden"],"States":["2","3"]}
```

Note that not specifying a parameter is the same as specifying all the values for that parameter, e.g. not including the CountryCodes attribute is the same as getting orders for all countries.

13.3.1 Order Report Attributes

Below we list some the filter parameters for the order API that have static values and what those values are.

Variable	Type	Description
States	enum	**The state of the order. Available states are:** Pending = 0 Delivered = 1 Cancelled = 2 Returned = 3 Invoiced = 4
CountryCodes	string	**Country of the order, indicating in what channel the** Sweden Denmark Norway Finland
PaymentStates	enum	**Indicates the state of the payment. Available states a** NotApplicable = 0 AwaitingPayment = 1 Paid = 2 AwaitingRefund = 3 Refunded = 4

13.3.2 Code Example - C#

Below you can find an example of a method that calls the POST api to get a report of pending orders. The method takes the path to the API, i.e. /api/reports, and the reportId of the kind of report to generate.

```
public string Post(Guid repordId, string path)
{
  var filter = new JavaScriptSerializer().Serialize(new
  {
         States = new[] { "0" } // Pending state
  });

  var content = new FormUrlEncodedContent(new[]
  {
         new KeyValuePair("ReportId", repordId.ToString()),
         new KeyValuePair("format", "json"),
         new KeyValuePair("filter", filter)
  });

  var httpClient = new HttpClient() { BaseAddress = new Uri("https://admin.
↪marketplace.cdon.com/") };
  httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("api
↪", ApiKey);
  var response = httpClient.PostAsync(path, content).Result;
  response.EnsureSuccessStatusCode();

  return response.Content.ReadAsStringAsync().Result;
}
```

Welcome to the documentation of the CDON Marketplace merchant integration!

Marketplace has a number of **APIs** for system integration, and also an **administration** page for manual management.

Note: To use either an API or the administration pages, you need to have a valid account with CDON Marketplace.

The APIs make it possible for merchants to send and receive data via HTTP to manage their inventory and orders. Communication with the APIs is not restricted to any specific programming language. However, the data exchange must be formatted in the right way, which is described in these documents.