Nedap Sense

version 1, August 2025

Introduction

The Status Management feature introduces functionalities to mark items as ‘special’. For example, to denote that an item is faulty, displayed on a mannequin, or currently being lent to a third-party. These special states help the retailer to decide what is currently directly for sale (or not) so that (digital) product availability can be optimized and sales maximized.

This **Getting Started-guide** helps to deploy and configure Status Management for an organization.

Full experience Status Management with iD Cloud realtime

The currently available features of Status Management are designed for iD Cloud realtime. Customers already using iD Cloud realtime can therefore start using the newly released features. For 'snapshot' users there are still future developments needed to be able to make full use of Status Management. Please read the 'Getting Started guide for Status Management' thoroughly on our support portal to get to know all the ins and outs of Status Management and the current setup.

Assign Status via the iD Cloud app

As of iOS v3.17.0 and Android v3.18.0 it is possible to assign a status to an item via the Identify functionality in the iD Cloud app. More details on how this works can be found on the **Assign Status feature page** on the support portal. To be able to start using this functionality, configuration is needed on the Nedap servers. Please contact our support desk for assistance.

Manage Status in iD Cloud web

When within an organization assigned statuses are used, then a store level page in iD Cloud web can be enabled to get more insight in which items currently have a status assigned. This page needs to be turned on by our support desk. For more detailed information about this page, please refer to the **Manage Status feature page** on the support portal.

Assign Status via an API integration

Next to the Assign Status functionality in the app, it is also possible to do this via an API integration. For each supported status we have an endpoint available in our **Transaction API**. These endpoints can be used to give one or more EPCs a certain status for example from the POS system. Please check out **Support statuses in iD Cloud** for a complete overview of all options.

Example: place EPCs on 'hold' status

When items need to get the status 'hold', for example for a Click & Collect order, the following API call can be used to set one or multiple EPCs on this status.

`POST https://api.nedapretail.com/transaction/v2/hold`

Example to place an EPC on 'hold' status

```
{
    "identifiers": [
        "urn:epc:id:sgtin:2000000.000123.187825966147",
        "urn:epc:id:sgtin:2000000.000102.177626772357"
    ],
    "identifiers_type": "epc_uri",
    "biz_location": "http://nedapretail.com/loc/store-512216",
    "timestamp": "2020-08-25T10:50:00+02:00",
    "held": true
}
```

Release an assigned status

If a status that was assigned to an item is no longer valid, then this status can be released again. This can be done via the iD Cloud app, but also via an API call.

The below example shows the same two EPCs from the previous example, which are no longer needed to hold for the customer. Therefore the status can be released. This time for the `held` parameter the value `false` is used, which will change the status of the item back to `sellable_accessible` in the same bizStep as defined for this endpoint.

`POST https://api.nedapretail.com/transaction/v2/hold`

Example to release a status

```
{
    "identifiers": [
        "urn:epc:id:sgtin:2000000.000123.187825966147",
        "urn:epc:id:sgtin:2000000.000102.177626772357"
    ],
    "identifiers_type": "epc_uri",
    "biz_location": "http://nedapretail.com/loc/store-512216",
    "timestamp": "2020-08-25T10:50:00+02:00",
    "held": false
}
```

Release a status 'to' a sublocation

When an EPC is released with the biz_location set as the store level, then the item will be released with the disposition `sellable_accessible` ***. It is also possible to release an EPC back to a specific sublocation in the store, for example the sales floor or the stockroom. Releasing an EPC with biz_location set as the stockroom, will release the item with the disposition*** `sellable_not_accessible` ***. Using the proper sublocation to release the item to, will also update the realtime stock information and therefore Refill suggestions and Availability of these are set to realtime stock as data source.***

Exporting stock from our APIs

RFID Stock API

Statuses are assigned to items in the store. To export a certain stock view, the **RFID Stock API** can be used. This API returns a list of GTIN-quantities (or a list of EPCs) that represents the RFID Stock of the store. This response can be filtered on certain dispositions, to only retrieve the financial stock of a store for example.

Filter on dispositions

When no dispositions are specified, this endpoint only returns items with dispositions `urn:epcglobal:cbv:disp:sellable_accessible` ***or*** `urn:epcglobal:cbv:disp:sellable_not_accessible`***. So it will filter out any items that have been assigned a different status / disposition.***

(Approved) Difference List API

A lot of retailers make use of the Difference List to approve and export their stock to an ERP system. At the moment, this Difference List compares an RFID Count with an ERP Stock and returns the differences to the ERP system, once approved. It is possible to adjust which dispositions are included in the RFID Count results by using a managed setting. This will add the counted items of non-sellable items to the counted items in the Difference List.
The setting to use for this is: `idcloud.realtime.default_dispositions_to_count` and can be request at our support team.

The Difference List itself will remain on GTIN / quantity level, so it is not possible to see the individual dispositions in those results. The included dispositions are all included in one value for RFID Count.

Supported statuses in iD Cloud

Introduction

The Status Management feature introduces functionalities to mark items as ‘special’. For example, to denote that an item is faulty, displayed on a mannequin, or currently being lent to a third-party. These special states help the retailer to decide what is currently directly for sale (or not) so that (digital) product availability can be optimized and sales maximized.

This **Getting Started-guide** helps to deploy and configure Status Management for an organization.

Supported statuses

Within the Assign status functionality in the iD Cloud app, the below statuses are supported. These can be turned on by our iD Cloud support team. These statuses can also be applied via different endpoints in the **Transaction API**.

Status	Description	Biz_step	Disposition
On display	Puts item(s) on dispay.	`http://nedapretail.com/bizstep/displaying`	`http://nedapretail.com /disp/on_display`
In showcase	Puts item(s) in a showcase	`http://nedapretail.com/bizstep/displaying`	`http://nedapretail.com /disp/in_showcase`
Lent	Lends item(s) to a third-party.	`http://nedapretail.com/bizstep/lending`	`http://nedapretail.com /disp/lent`
Damaged	Item is damaged.	`urn:epcglobal:cbv:bizstep:inspecting`	`urn:epcglobal:cbv:disp:damaged`
Missing article	Item is missing.	`urn:epcglobal:cbv:bizstep:inspecting`	`http://nedapretail.com /disp/missing_article`
Faulty	Item is faulty.	`urn:epcglobal:cbv:bizstep:inspecting`	`http://nedapretail.com /disp/faulty`

Status	Description	Biz_step	Disposition
Customized	Item(s) are customized.	`http://nedapretail.com/bizstep/customizing`	`http://nedapretail.com /disp/customized`
Hemming	Item(s) are being hemmed.	`http://nedapretail.com/bizstep/customizing`	`http://nedapretail.com /disp/hemming`
Online sold	Item is sold online to a customer.	`urn:epcglobal:cbv:bizstep:retail_selling`	`http://nedapretail.com /disp/online_sold`
Reserved	Item(s) are temporarily reserved for a customer. The item is awaiting pickup by the customer. Examples: e-reservation, reservation by telephone.	`http://nedapretail.com/bizstep/retail_reserving`	`http://nedapretail.com /disp/retail_reserved`
Reserved for peak	Item is marked as reserved for peak days.	`http://nedapretail.com/bizstep/retail_reserving`	`http://nedapretail.com /disp/retail_reserved_for_peak`
Held	Item is held for a customer and is awaiting pick-up. Typically these items are already (partly) sold to the customer.	`urn:epcglobal:cbv:bizstep:holding`	`urn:epcglobal:cbv:disp:non_sellable_other`
Received order	Item is part of a received order. The order is awaiting pickup by customer. Examples: click&collect, bopis, e.g.	`urn:epcglobal:cbv:bizstep:holding`	`http://nedapretail.com /disp/received_order`

Next to above listed assignable statuses, there are additional dispositions used within our iD Cloud features. This overview can be found here: **Business steps and dispositions used in iD Cloud**

Retrieve stock with a certain status

Introduction

With the introduction of Status Management within iD Cloud, we have extended the list of **supported statuses**. Below you can find a guideline how to retrieve stock with a certain status (disposition) from our APIs.

Retrieve the current stock levels of a store

Using the RFID Stock API it is possible to extract the current stock levels of a store, split by disposition (status). The endpoint '**Retrieve SGTINs grouped by disposition**' can be used for this.

This endpoint returns the realtime stock at a certain location in unique items (SGTINs), grouped by disposition. Items are returned based on their last known bizLocation (location). The response contains a list of stocks for all matching (sub)locations based on the given parameters:

- If the stock for a store without sublocations is requested, the stocks list in the response will contain one entry containing the stock for the entire store.
- If a store has sublocations, the list of stocks in the response will contain an entry each for the sublocations.
- If a store has stock for both sublocations and store, the list of stocks in the response contains entries for the store and the sublocations with stock.
- When no dispositions are specified, this endpoint only returns items with dispositions `urn:epcglobal:cbv:disp:sellable_accessible` or `urn:epcglobal:cbv:disp:sellable_not_accessible`.

Retrieve all current stock with a certain disposition

The below example API call requests the stock for the following dispositions:

- `sellable_accessible` (Sales Floor)
- `sellable_not_accessible` (Stockroom)
- `urn:epcglobal:cbv:disp:damaged` (Damaged items)
- `urn:epcglobal:cbv:disp: non_sellable_other` (Held items)

Example request

```
curl --request GET \
     --url 'https://api.nedapretail.com/rfid_stock/v1/
retrieve_grouped_by_disposition?
```

```
location=http%253A%252F%252Fnedapretail.com%252Floc%252F512345&dispositions%5B%5D=urn%253Aepcglobal%253Acbv%253Adisp%253Asellable_accessible&dispositions%5B%5D=urn%253Aepcglobal%253Acbv%253Adisp%253Asellable_not_accessible&dispositions%5B%5D=urn%253Aepcglobal%253Acbv%253Adisp%253Adamaged&dispositions%5B%5D=urn%253Aepcglobal%253Acbv%253Adisp%253A%2520non_sellable_other' \
     --header 'Accept: application/json'
```

The response that is provide will give a list of EPCs per specified location and for each requested disposition. Below we have only shown the responses that included items. In practice for each available sublocation in the store a response will be provided per disposition. So if no items are present then the disposition will still be listed, with quantity `0`.

Example response


```
{
    "stocks": [
        {
            "location": "http://nedapretail.com/loc/sublocation-1001233",
            "time": "2022-03-16T10:38:12.912Z",
            "quantity": 5,
            "sgtins": [
                "urn:epc:id:sgtin:2000000.000124.253603768425",
                "urn:epc:id:sgtin:2000000.000004.149283639107",
                "urn:epc:id:sgtin:2000000.000102.253867561986",
                "urn:epc:id:sgtin:2000000.000122.251084200077",
                "urn:epc:id:sgtin:2000000.000102.187088508764"
            ],
            "disposition": "urn:epcglobal:cbv:disp:sellable_accessible"
        },
        {
            "location": "http://nedapretail.com/loc/sublocation-1001234",
            "time": "2022-03-16T10:38:12.912Z",
            "quantity": 5,
            "sgtins": [
                "urn:epc:id:sgtin:2000000.000124.223603768425",
                "urn:epc:id:sgtin:2000000.000004.143283639103",
                "urn:epc:id:sgtin:2000000.000102.259867561986",
                "urn:epc:id:sgtin:2000000.000122.250084200078",
                "urn:epc:id:sgtin:2000000.000102.184088508765"
            ],
            "disposition": "urn:epcglobal:cbv:disp:sellable_not_accessible"
        },
        {
            "location": "http://nedapretail.com/loc/sublocation-1001235,
            "time": "2022-03-16T10:38:12.912Z",
            "quantity": 5,
            "sgtins": [
                "urn:epc:id:sgtin:2000000.000124.223603768425",
                "urn:epc:id:sgtin:2000000.000004.143283639107",
```

```
                "urn:epc:id:sgtin:2000000.000102.259867561986",
                "urn:epc:id:sgtin:2000000.000122.250084200077",
                "urn:epc:id:sgtin:2000000.000102.184088508764"
            ],
            "disposition": "urn:epcglobal:cbv:disp:sellable_not_accessible"
        },
        {
            "location": "http://nedapretail.com/loc/store-512345",
            "time": "2022-03-16T10:38:12.912Z",
            "quantity": 1,
            "sgtins": [
                "urn:epc:id:sgtin:2000000.000123.266883396582"
            ],
            "disposition": "urn:epcglobal:cbv:disp:damaged"
        },
        {
            "location": "http://nedapretail.com/loc/store-512345",
            "time": "2022-03-16T10:38:12.912Z",
            "quantity": 2,
            "sgtins": [
                "urn:epc:id:sgtin:2000000.000122.250084200077",
                "urn:epc:id:sgtin:2000000.000102.184088508764
            ],
            "disposition": "urn:epcglobal:cbv:disp:non_sellable_other"
        }
    ]
}
```


Retrieve only just added EPCs to a certain disposition

Using the same endpoint **/rfid_stock/v1/retrieve_grouped_by_disposition** also provides the option to retrieve added EPCs to a certain disposition after a specific time.

By adding the parameter `after_timestamp` to the API call, the API will only return EPCs that were added to that specific disposition that have `timestamp` field greater or equal to specified date/time.

Document Version 1

Document Last modification date 11 August 2025

Document PDF Exported 11 August 2025 **by** Nedap Retail | Operations

support-retail@nedap.com

Nedap Retail
Parallelweg 2
NL7141 DC Groenlo
The Netherlands

nedap-retail.com