Customer-facing API

- Inventory

Inventory

Purpose

This API allows you to retrieve your current warehouse inventory from FMT Bree.

Two views are available:

- Detailed inventory: one record per pallet
- Summary inventory: grouped by product and pallet type, including pallet counts and weight totals

The API supports both JSON and XML.

What You Need Before You Start

FMT Bree will provide:

- Your customer API key

Authentication

Every request must include your customer API key.

Preferred method:

```
X-API-Key: your-api-key-here
```

Fallback method:

```
?api_key=your-api-key-here
```

The query parameter is supported for systems that cannot send custom headers, but the header is recommended because it is safer.

Response Format

You can request JSON or XML.

JSON

Send this header:

```
Accept: application/json
```

XML

Send this header:

```
Accept: application/xml
```

Optional format parameter

You can also force the format with:

```
?format=json
```

or:

```
?format=xml
```

If both are used, the `format` query parameter takes precedence.

Endpoints

1. Detailed inventory

Returns your current stock per pallet.

```
GET /api/customer/inventory
```

Example:

```
curl -X GET "https://wms.fmtbree.be/api/customer/inventory" \
  -H "Accept: application/json" \
  -H "X-API-Key: your-api-key-here"
```

XML example:

```
curl -X GET "https://wms.fmtbree.be/api/customer/inventory" \
  -H "Accept: application/xml" \
  -H "X-API-Key: your-api-key-here"
```

2. Summary inventory

Returns your current stock grouped by product and pallet type.

```
GET /api/customer/inventory/summary
```

Example:

```
curl -X GET "https://wms.fmtbree.be/api/customer/inventory/summary" \
  -H "Accept: application/json" \
  -H "X-API-Key: your-api-key-here"
```

Detailed Inventory Response

The detailed endpoint returns:

- Customer information
- Snapshot timestamp
- A list of pallets currently in stock

Example JSON:


```
{
  "customer": {
    "id": 1,
    "name": "Example Customer"
  },
  "snapshot_at": "2026-03-03T10:15:00+01:00",
  "inventory": [
    {
      "barcode": "PALLET-001",
      "external_id": "ERP-001",
      "batch": "BATCH-001",
      "status": "stock",
      "pallet_type": "EUR",
      "location_identifier": "A01 01",
      "depth": 1,
      "arrived_at": "2026-03-01",
      "produced_at": "2026-02-20",
      "frozen_at": "2026-02-21",
      "expired_at": "2027-02-21",
      "nett": 800,
      "tare": 25,
```

```
      "gross": 825,
      "quantity": 40,
      "bio": false,
      "c3": false,
      "product": {
        "id": 10,
        "customer_pid": "SKU-001",
        "name": "Chicken fillet",
        "barcode": null
      },
      "packaging": {
        "id": 2,
        "name": "Box"
      }
    }
  ]
}
```


Field explanation

- `barcode`: unique pallet number in the warehouse
- `external_id`: your external reference if available
- `batch`: batch or lot number
- `status`: current pallet status
- `pallet_type`: pallet size/type such as `EUR`, `IND`, `EXT`, `EUH`, `INH`
- `location_identifier`: warehouse location
- `depth`: pallet depth within a location
- `arrived_at`: date the pallet entered stock
- `produced_at`: production date
- `frozen_at`: freezing date
- `expired_at`: expiry date
- `nett`: net weight
- `tare`: tare weight
- `gross`: gross weight
- `quantity`: quantity on the pallet
- `bio`: biological product flag
- `c3`: category 3 flag
- `product.customer_pid`: your own product code in the warehouse system

Summary Inventory Response

The summary endpoint returns:

- Customer information
- Snapshot timestamp
- Grouped stock lines
- Grand totals

Example JSON:


```
{
  "customer": {
    "id": 1,
    "name": "Example Customer"
  },
  "snapshot_at": "2026-03-03T10:15:00+01:00",
  "summary": [
    {
      "product": {
        "id": 10,
        "customer_pid": "SKU-001",
        "name": "Chicken fillet",
        "barcode": null
      },
      "pallet_type": "EUR",
      "pallet_count": 12,
      "weights": {
        "nett": 9600,
        "tare": 300,
        "gross": 9900
      },
      "quantity": 480
    }
  ],
  "totals": {
    "pallet_count": 12,
    "nett": 9600,
    "tare": 300,
    "gross": 9900,
    "quantity": 480
  }
}
```


Summary meaning

Each line in `summary` is grouped by:

- Product
- Pallet type

This means the same product can appear multiple times if it exists on different pallet types.

XML Responses

The XML response contains the same information as JSON.

Example request:

```
curl -X GET "https://wms.fmtbree.be/api/customer/inventory/summary" \
  -H "Accept: application/xml" \
  -H "X-API-Key: your-api-key-here"
```

Example XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<inventorySummaryResponse>
  <customer>
    <id>1</id>
    <name>Example Customer</name>
  </customer>
  <snapshot_at>2026-03-03T10:15:00+01:00</snapshot_at>
  <summary>
    <item>
      <product>
        <id>10</id>
        <customer_pid>SKU-001</customer_pid>
        <name>Chicken fillet</name>
        <barcode></barcode>
      </product>
      <pallet_type>EUR</pallet_type>
      <pallet_count>12</pallet_count>
      <weights>
        <nett>9600</nett>
        <tare>300</tare>
        <gross>9900</gross>
      </weights>
```

```
      <quantity>480</quantity>
    </item>
  </summary>
  <totals>
    <pallet_count>12</pallet_count>
    <nett>9600</nett>
    <tare>300</tare>
    <gross>9900</gross>
    <quantity>480</quantity>
  </totals>
</inventorySummaryResponse>
```

Error Responses

Missing API key

Status:

```
401 Unauthorized
```

JSON example:

```
{
  "error": "unauthorized",
  "message": "API key is required."
}
```

Invalid API key

Status:

```
401 Unauthorized
```

JSON example:

```
{
  "error": "unauthorized",
  "message": "Invalid API key."
}
```

Unsupported format

Status:

```
406 Not Acceptable
```

JSON example:

```
{
  "error": "not_acceptable",
  "message": "Unsupported response format."
}
```

Practical Recommendations

- Prefer `X-API-Key` over `?api_key=`
- Use HTTPS only
- Keep your API key private
- Store the API key in your integration settings, not in source code if possible
- If you believe your key has been exposed, ask FMT Bree to replace it

Typical Integration Flow

1. Receive your API key from FMT Bree
2. Call `/api/customer/inventory/summary` if you only need totals per product and pallet type
3. Call `/api/customer/inventory` if you need pallet-level detail
4. Choose JSON or XML depending on your ERP, WMS, or EDI integration

Support

If you need:

- a new API key
- a replacement API key
- help mapping fields to your system

contact FMT Bree.