

# RESTAPI DOCUMENTATION FOR CLIENTS

**Hart Sp. z o.o.**

Author	IT Department Hart Sp. z o.o.
Category	Confidential – For Client Use only, Hart Sp. z o.o.
Reference	RestAPI-HART-TN-000-RestAPI_Customer_Documentation_EN
Version	1
Revision	5
Status	Released
Date	2025-05-08
API Version	1.4.0

## Release Schedule

Version and Revision	Date (YYYY-MM-DD)	Description of Changes
1.0	2024-12-20	Initial release
1.1	2025-01-22	<p>Chapter "API Endpoints Documentation" – Changes in API response structure:</p> <p>Section „Products”:</p> <ul style="list-style-type: none"><li>▪ Added response wrapper structure: "value", "isSuccess", "errorMessage"</li><li>▪ Added "currency" field in product responses</li><li>▪ Renamed "priceNetto" to "priceNet" in responses</li><li>▪ Added "priority" field in responses regarding product availability</li></ul> <p>Section "Basket":</p> <ul style="list-style-type: none"><li>▪ Added response wrapper structure: "value", "isSuccess", "errorMessage"</li><li>▪ Renamed "minWaitingTimeInfo" to "waitingTime"</li></ul> <p>Section "Order":</p> <ul style="list-style-type: none"><li>▪ Changed the order deletion endpoint from "/v1/orders?OrderId=" to "/v1/orders/{OrderId}"</li><li>▪ Modified the way basket item IDs are sent when creating an order, changed from query param "BasketIds" to body param "basketPositionIds"</li></ul> <p>Section "Documents":</p> <ul style="list-style-type: none"><li>▪ Fixed field names in responses: "createdData" -&gt; "createdDate" "paymentData" -&gt; "paymentDate" "saledData" -&gt; "soldDate" "amountBrutto" -&gt; "amountGross" "amountNetto" -&gt; "amountNet"</li><li>▪ Added information about the necessity of encoding the "/" character in document number as "%2F"</li><li>▪ Renamed field names in corrections: "quantityBefor" -&gt; "quantityBefore" "quantityAfter" -&gt; "quantityAfter" "priceNettoBefor" -&gt; "priceNetBefore" "priceNettoAfter" -&gt; "priceNetAfter"</li></ul>
1.2	2025-02-06	<p>Chapter "API Endpoint Documentation" - Changes in API request structure:</p> <p>Section "Products":</p> <ul style="list-style-type: none"><li>• Renamed Query parameter field name: ProductCodes -&gt; HartCodes</li></ul> <p>Section "Basket":</p>

		<ul style="list-style-type: none"> <li>The Body parameter now accepts a list of products to add to the basket</li> </ul> <p>Changes in the API response structure:</p> <p>Section "Products":</p> <ul style="list-style-type: none"> <li>Added descriptions of returned objects' fields</li> <li>Added field: waitingTime</li> </ul> <p>Section "Basket":</p> <ul style="list-style-type: none"> <li>Added descriptions of returned objects' fields</li> <li>Corrected returned objects in responses</li> </ul> <p>Section "Orders":</p> <ul style="list-style-type: none"> <li>Added descriptions of returned objects' fields</li> <li>Corrected returned objects in responses</li> </ul> <p>Section "Documents":</p> <ul style="list-style-type: none"> <li>Added descriptions of returned objects' fields</li> </ul> <p>Changes in the "Usage Limits" section:</p> <ul style="list-style-type: none"> <li>Corrected description for call limits: Call limit per minute: 50 requests per client</li> </ul>
1.3	2025-04-04	Minor improvements and internal optimizations (no breaking changes or new client-facing endpoints introduced in this revision)
1.4	2025-04-09	<p>Chapter "API Endpoint Documentation"</p> <p>Section "Products":</p> <ul style="list-style-type: none"> <li>Query HartCodes parameter has been improved – it has been marked as required</li> </ul> <p>The latest version of OpenAPI files have been generated.</p>
1.5	2025-05-08	<p>Added chapter "HART Magazines"</p> <ul style="list-style-type: none"> <li>Contains a table with HART warehouses and how to use their IDs</li> </ul> <p>Added chapter "Ordering workflow"</p> <ul style="list-style-type: none"> <li>Contains a description of the sequence of using API endpoints in order to place an order correctly</li> </ul>

## Table of Contents

<b>Release Schedule .....</b>	<b>2</b>
<b>Overview and Introduction .....</b>	<b>6</b>
<b>Getting Started Guide.....</b>	<b>7</b>
<b>HART warehouses.....</b>	<b>9</b>
<b>Authentication and Authorization .....</b>	<b>11</b>
<b>Authentication.....</b>	<b>11</b>
<b>API Endpoints Documentation .....</b>	<b>12</b>
<b>Products.....</b>	<b>12</b>
Product Catalog.....	12
Product Details.....	15
Products Availability.....	17
Product Availability .....	19
<b>Basket .....</b>	<b>21</b>
Basket Contents .....	21
Adding a New Item to the Basket.....	24
Updating an Item in the Basket.....	26
Removing an Item from the Basket.....	27
<b>Orders .....</b>	<b>28</b>
Order List.....	28
Order Details .....	30
Creating and Order.....	32
Deleting an Order.....	34
<b>Documents.....</b>	<b>35</b>
Invoice Collection .....	35
Invoice .....	38
Invoice Corrections Collection .....	41
Invoice Correction .....	44
Delivery Notes Collection (WZ) .....	47
Delivery Note (WZ).....	50
Bonus Correction Invoice Collection .....	53
Bonus Correction Invoice .....	56

<b>Ordering workflow .....</b>	<b>59</b>
<b>Usage Limits.....</b>	<b>61</b>
<b>Security and Compliance.....</b>	<b>63</b>
<b>Service Level Agreement (SLA) .....</b>	<b>65</b>
<b>Production and Test Environment (Sandbox) .....</b>	<b>67</b>
<b>Production Environment .....</b>	<b>67</b>
<b>Test Environment (Sandbox) .....</b>	<b>68</b>
<b>Support and Communication Channels .....</b>	<b>69</b>
<b>Legal and Regulatory Information .....</b>	<b>70</b>
<b>I. Definitions .....</b>	<b>70</b>
<b>II. Terms of Service Provision .....</b>	<b>70</b>
<b>III. Privacy Policy.....</b>	<b>71</b>
<b>IV. Compliance Obligations .....</b>	<b>72</b>
<b>Glossary and Acronyms.....</b>	<b>74</b>
<b>Attachments .....</b>	<b>75</b>
<b>Attachment No. 1 – OpenAPI 3.x.x specification.....</b>	<b>75</b>

## Overview and Introduction

The RestAPI/API has been designed to enable customers of Hart Sp. z o.o. – a leading distributor of automotive parts – to integrate their IT systems directly with the Hart system. The API provides automated access to the product catalog, facilitates order placement, and allows the retrieval of transaction-related documents.

### API purpose

The primary goal of the API is to streamline the purchasing process. The API automates the verification of product availability and pricing and simplifies order placement. Through direct system integration, the API ensures quick access to commercial documentation, reduces the time required for routine purchasing operations, and minimizes the risk of errors.

### Use Cases

#### Order Management

The API allows for bulk verification of product availability and direct order placement from the client's warehouse systems. It also provides order status tracking and automatic retrieval of documents and invoices.

#### Optimization of Purchasing Process

With the API, it is possible to automatically monitor prices and the availability of key products. Integration with inventory management systems enables the automation of ordering frequently purchased parts.

### API Access

The API is available to existing Hart Sp. z o.o. customers as well as new customers upon establishing a business relationship. API access is granted for the purpose of conducting actual purchase transactions. The API is not intended to be used as an information source for pricing without the intent to make purchases.

### Key Features

The API provides a comprehensive set of functions for managing the product catalog, including browsing available products, checking prices and availability, and accessing detailed product information. The system supports the full purchasing process—from adding products to the cart, placing orders, and managing submitted orders. The API also grants access to complete commercial documentation, including invoices, corrective documents, WZ (goods issued) documents, and corrective invoices documenting granted bonuses.

### Business Benefits

Implementing the API results in significant time savings by automating routine purchasing processes and providing quick access to product information and documents. Automated order validation and consistent communication between systems effectively reduce the risk of errors related to manual data entry.

The API enables business process optimization by improving inventory management and accelerating order fulfillment. The system's 24/7 availability allows orders to be placed at any time, ensuring constant access to product information and automatic document retrieval.

## Getting Started Guide

### Access Requirements

The process of obtaining API access begins with contacting a Hart sales representative. The sales representative will collect the necessary company information and technical details regarding the planned integration. Upon successful verification and completion of formalities, the Hart IT Department will prepare and provide access credentials to the designated contact person within the client company.

Clients will receive access to both the testing (sandbox) and production environments. Separate authentication credentials will be provided for each environment, which is a standard practice to ensure secure integration and testing before transitioning to the production environment.

### Implementation Process

The primary technical requirement for using the API is a stable internet connection and adequately secured client systems. Security measures should comply with current industry standards, including up-to-date software, antivirus protection, and proper firewall configuration.

For managing access credentials, secure storage practices are recommended. Credentials should never be stored directly in source code. Instead, secure configuration storage mechanisms, such as environment variables or dedicated password management systems, should be used.

The implementation process should start in the testing environment, where necessary integration tests can be safely conducted. At this stage, verifying the authentication mechanisms and basic API operations is crucial. Only after successful testing and approval should the deployment proceed to the production environment.

### Integration Overview

API integration requires the implementation of several key components. The primary element is an authentication system that handles login processes and JWT token management. Another essential component is the communication layer responsible for sending HTTP requests and processing responses.

During integration, special attention should be given to proper error handling. The system should be prepared to handle various HTTP response codes and implement retry mechanisms for failed requests. It is equally important to introduce an appropriate level of event logging to facilitate later diagnostics of potential issues.

Before transitioning to the production environment, ensure that all test scenarios have been successfully verified and that the system properly handles failure scenarios. It is also advisable to implement monitoring of the integration's performance, allowing for quick detection and response to potential issues.

### Technical Support

Hart provides technical support at every stage of integration. In case of questions or technical issues, clients should contact the Hart IT Department through their assigned sales representative. Business-related matters are handled by the designated sales representative, who is also responsible for informing clients about planned changes and API updates.

With both test and production environments available, clients can safely develop and test their integrations. If major system changes are planned, it is recommended to test them in the testing environment before deploying them to production.



## HART warehouses

The HART warehouses table is used to store data critical for selecting the warehouse to fulfill an order. In particular, pay attention should to the **ID** column, the value of which should be placed as a **BranchId** header parameter to explicitly indicate the selected warehouse when working with the API:

- **ID**: The main column identifying the warehouse. It is used when there is a need to specify a particular warehouse instead of the one assigned to the client by default.
- **Code**: A column containing the unique code of the warehouse. It can be helpful for mapping warehouse codes to their corresponding IDs.
- **Branch**: The name of the warehouse, useful for easy reading and identifying the warehouse by users.

Example of use: When sending API requests, the **ID** column can be used as the value of the **BranchId** header, allowing the request to be associated with a specific warehouse (e.g., retrieving product availability or placing an order).

HART WAREHOUSES		
ID	Code	Branch
1	1	Centrala OPOLE - HUB
2	2	Wrocław
3	3	Bytom
4	4	Wieliczka
8	8	Poznań
9	9	Katowice
10	A	Łódź
11	B	Częstochowa
12	C	Zielona Góra
13	D	Kraków
15	F	Białystok
16	G	Targówek - Warszawa 1
17	H	Rzeszów
18	I	Bielsko-Biała

19	J	Warszawa 2
20	K	Radom
22	M	Psary
23	N	Bydgoszcz
24	O	Ostrava (CZ)
25	P	Brno (CZ)
26	R	Prešov (SK)
<b>27</b>	<b>S</b>	<b>Koluszki - HUB</b>
28	T	Szczecin
29	U	Gdańsk
30	V	Lublin
31	W	Nitra (SK)
32	X	Warszawa 3

# Authentication and Authorization

## Authentication

Authentication is performed using the access credentials provided by Hart. Upon successful authentication, a JWT token and its expiration time will be returned. Once the token expires, authentication must be performed again to obtain a new JWT token.

### POST /v1/auth

- **URL:** /v1/auth
- **Method:** POST
- **Description:** Perform authentication in the system
- **Required Parameters:**
  - **BODY:** JSON containing authentication credentials
    - username – User's login name
    - password – User's password
- **Response:** The response includes the access token (JWT) and its expiration time
  - **access\_token** – JWT access token used for authenticating subsequent requests
  - **expires\_in** – Token validity period in seconds

```
curl -X 'POST' \
  'https://sandbox.restapi.hartphp.com.pl/v1/auth' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "1111",
    "password": "****"
  }'
```

### Response Codes:

- **200 OK**

```
{
  "access_token": "string",
  "expires_in": number
}
```

- **401 Unauthorized**

## API Endpoints Documentation

### Products

The API provides access to the product catalog offered by Hart. Each request can be customized with query parameters to facilitate smooth navigation through the catalog.

### Product Catalog

#### GET /v1/products

- **URL:** /v1/products
- **Method:** GET
- **Description:** Returns a collection of products
- **Required Parameters**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **QUERY**
    - HartCodes – Selected product codes
- **Optional Parameters**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Availability – Product availability (true or false)
    - SortBy – sorting attribute name, possible to use on of the value:
      - Name – product name,
      - SupplierName – supplier name,
      - HartCode – HART code
    - SortDirection – Sorting order: ASC or DESC
    - Size – Collection size
    - Page – Page number
- **Response:** Paginated collection of products
  - **hartCode** – HART product code
  - **name** – product name
  - **supplierCode** – supplier code
  - **unit** – unit type
  - **supplier** – supplier name
  - **taxRate** – tax rate
  - **onOrder** – on order

- **withdrawn** – withdrawn from sale
- **quantity** – number of available products, if more than 10, then 10 is returned
- **waitingTime** – waiting time
- **isPatent** – whether the goods are covered by a patent in the contractor's country
- **isPriceForManyPieces** – is the price for multiple pieces in a package
- **numberOfPecesInPrice** – number of pieces in a price
- **sellingPrice** – selling price
- **currency** - currency

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/products' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

#### Responses Codes:

- **200 OK**

```
{
  "items": [
    {
      "value": {
        "hartCode": "string",
        "name": "string",
        "supplierCode": "string",
        "unit": "string",
        "supplier": "string",
        "taxRate": number,
        "onOrder": boolean,
        "withdrawn": boolean,
        "quantity": number,
        "waitingTime": "string",
        "isPatent": boolean,
        "isPriceForManyPieces": boolean,
        "numberOfPiecesInPrice": number,
        "sellingPrice": number,
        "currency": "string",
      },
      "isSuccess": boolean,
      "errorMessage": "string",
    }
  ],
  "total_pages": number,
  "current_page": number,
  "items_from": number,
  "items_to": number,
  "total_items_count": number
}
```

- **401 Unauthorized**

- **404 Products not found**

## Product Details

### GET /v1/products/{HartCode}

- **URL:** /v1/products/{HartCode}
- **Method:** GET
- **Description:** Returns details of a selected product identified by HartCode
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - HartCode – Product identifier
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Availability – Product availability
- **Response:** JSON with selected product details
  - **hartCode** – HART product code
  - **name** – product name
  - **supplierCode** – supplier code
  - **unit** – unit type
  - **supplier** – supplier name
  - **taxRate** – tax rate
  - **onOrder** – on order
  - **withdrawn** – withdrawn from sale
  - **quantity** – number of available products, if more than 10, then 10 is returned
  - **waitingTime** – waiting time
  - **isPatent** – whether the goods are covered by a patent in the contractor's country
  - **isPriceForManyPieces** – is the price for multiple pieces in a package
  - **numberOfPecesInPrice** – number of pieces in a price
  - **sellingPrice** – selling price
  - **currency** - currency

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/products/563%20728?Availability=true' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

### Response Codes:

- **200 OK**

```
{
  "value": {
    "hartCode": "string",
    "name": "string",
    "supplierCode": "string",
    "unit": "string",
    "supplier": "string",
    "taxRate": number,
    "onOrder": boolean,
    "withdrawn": boolean,
    "quantity": number,
    "waitingTime": "string",
    "isPatent": boolean,
    "isPriceForManyPieces": boolean,
    "numberOfPiecesInPrice": number,
    "sellingPrice": number,
    "currency": "string",
  },
  "isSuccess": boolean,
  "errorMessage": "string",
}
```

- **400 Bad Request**
- **401 Unauthorized**
- **404 Product not found**



## Products Availability

### GET /v1/products/availability

- **URL:** /v1/products/availability
- **Method:** GET
- **Description:** Returns a collection of availability information for selected products
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **QUERY**
    - HartCodes – Selected product codes
- **Parametry opcjonalne**
  - **HEADER**
    - BranchId – Branch ID
  - **Query**
    - SortBy – Sorting attribute, supported values:
      - BranchCode – Branch code,
      - Quantity – Quantity available
      - WaitingTime – Waiting time
    - SortDirection – Sorting order: ASC or DESC
    - Size – Collection size
    - Page – Page number
- **Response:** JSON containing availability information for selected products
  - **hartCode** – HART product code
  - **availabilityPerBranch** – availability at selected branches
    - **branchId** – branch ID
    - **branchCode** – branch code
    - **quantity** – number of available products, if more than 10, then 10 is returned
    - **waitingTime** – waiting time
    - **priority** – branch priority
    - **description**

```
curl -X 'GET' \
'https://sandbox.restapi.hartphp.com.pl/v1/products/availability?ProductCodes=318%20339&ProductCodes=149%20493&ProductCodes=925%20848' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXZW40Iiwia2lkIiA6ICJfY0RqVzd...'
```

### Response Codes:

- **200 OK**

```
{
  "items": [
    {
      "value": {
        "hartCode": "string",
        "availabilityPerBranch": [
          {
            "branchId": number,
            "branchCode": "string",
            "quantity": number,
            "waitingTime": number,
            "priority": number,
            "description": "string"
          }
        ]
      },
      "isSuccess": boolean,
      "errorMessage": "string",
    }
  ],
  "total_pages": number,
  "current_page": number,
  "items_from": number,
  "items_to": number,
  "total_items_count": number
}
```

- **401 Unauthorized**
- **404 Not found**

## Product Availability

### GET /v1/products/availability/{HartCode}

- **URL:** /v1/products/availability/{HartCode}
- **Method:** GET
- **Description:** Retrieves availability information for a specific product identified by HartCode
- **Required Parameters**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - HartCode – Product identifier (required)
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** JSON containing availability details for the selected product
  - **hartCode** – HART product code
  - **availabilityPerBranch** – availability at selected branches
    - **branchId** – branch ID
    - **branchCode** – branch code
    - **quantity** – number of available products, if more than 10, then 10 is returned
    - **waitingTime** – waiting time
    - **priority** – branch priority
    - **description**

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/products/availability/318%20339' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

### Response Codes:

- **200 OK**

```
{
  "value": {
    "hartCode": "string",
    "availabilityPerBranch": [
      {
        "branchId": number,
        "branchCode": "string",
        "quantity": number,
        "waitingTime": number,
        "priority": number,
        "description": "string"
      }
    ]
  }
}
```

```
    }  
  ],  
},  
"isSuccess": boolean,  
"errorMessage": "string",  
}
```

- **400 Bad Request**
- **401 Unauthorized**
- **404 Not found**

## Basket

The basket section is responsible for managing products added to the user's cart. It allows adding, modifying, viewing, and removing products from the basket.

### Basket Contents

#### GET /v1/basket

- **URL:** /v1/basket
- **Method:** GET
- **Description:** Retrieves the basket contents
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Page – Page number
    - Size – Number of records per page
- **Response:** Basket details
  - **basketPositions** – products in the basket
    - **basketPositionId** – basket item ID
    - **creationDate** – creation date
    - **productName** – product name
    - **productCode** – HART product code
    - **productContractorName** – supplier name
    - **quantity** – quantity
    - **netPrice** – net price
    - **currency** – currency
    - **grossPrice** – gross price
    - **isConfirmed** – is the position confirmed
    - **isReserved** – is reserved
    - **isExpired** – is expired
    - **isBoundedDiscount** – is bounded discount
    - **isIndividualDiscount** – is individual discount
    - **isExchangeCost** – is the item exchange cost
    - **exchangeCostProductCode** – exchange cost code
    - **exchangeCostProductName** – exchange cost name
    - **productVatRate** – tax rate

- **availabilityInformation** – information about availability
- **branchCode** – branch code
- **branchName** – branch name
- **basketSummary** – basket summary
  - **totalNetSellingPrice** – total net amount
  - **totalGrossSellingPrice** – total gross amount
  - **basketCurrency** – currency
  - **totalQuantity** – total number of products in the basket
  - **exchangeCostQuantity** – number of items with exchange cost

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/basket' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3VzZXQ6Iiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- 200 OK

```
{
  "value": {
    "basketPositions": {
      "items": [
        {
          "basketPositionId": number,
          "creationDate": "string",
          "productName": "string",
          "productCode": "string",
          "productContractorName": "string",
          "quantity": number,
          "netPrice": number,
          "currency": "string",
          "grossPrice": number,
          "isConfirmed": boolean,
          "isReserved": boolean,
          "isExpired": boolean,
          "isBoundedDiscount": boolean,
          "isIndividualDiscount": boolean,
          "isExchangeCost": boolean,
          "exchangeCostProductCode": "string",
          "exchangeCostProductName": "string",
          "productVatRate": number,
          "availabilityInformation": "string",
          "branchCode": "string",
          "branchName": "string"
        }
      ],
      "total_pages": number,
      "current_page": number,
      "items_from": number,
      "items_to": number,
      "total_items_count": number
    },
    "basketSummary": {
      "totalNetSellingPrice": number,
      "totalGrossSellingPrice": number,
      "basketCurrency": "string",

```

```
    "totalQuantity": number,  
    "exchangeCostsQuantity": numer  
  },  
  "isSuccess": boolean,  
  "errorMessage": "string",  
}
```

- **400 Bad Request**
- **401 Unauthorized**

## Adding a New Item to the Basket

### POST /v1/basket

- **URL:** /v1/basket
- **Method:** POST
- **Description:** Adds a new item to the basket
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **BODY**
    - orderPositions – list of products to add to the basket
      - hartCode – HART product code
      - quantity - quantity
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** Information about item added to the basket
  - **successfulOrders**
    - **orderBufferPositionId** – basket item ID
    - **orderQuantity** – quantity
    - **availabilityInformation** – information about availability
    - **isExchangeCost** – is the item exchange cost
  - **expirationDate** – expiration date
  - **productCode** – HART product code
  - **productName** – product name
  - **branchId** – branch ID
  - **information** – information about adding a product to the basket

```
curl -X 'POST' \
  'https://sandbox.restapi.hartphp.com.pl/v1/basket' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...' \
  -d '{
    "orderPositions": [
      {
        "hartCode": "000 001",
        "quantity": 2
      }
    ]
  }'
```

### Response Codes:



- **200 OK**

```
{
  "value": {
    "successfulOrders": [
      {
        "orderBufferPositionId": number,
        "orderedQuantity": number,
        "availabilityInformation": "string",
        "isExchangeCost": boolean
      }
    ],
    "expirationDate": "string",
    "productCode": "string",
    "productName": "string",
    "branchId": number,
    "information": "string"
  },
  "isSuccess": boolean,
  "errorMessage": "string",
}
```

- **401 Unauthorized**

## Updating an Item in the Basket

### **PATCH** /v1/basket/{PositionId}

- **URL:** /v1/basket/{PositionId}
- **Method:** PATCH
- **Description:** Updates a selected item in the basket
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - PositionId – Basket position ID
  - **BODY**
    - quantity - Quantity
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** Confirmation of updating items in the basket

```
curl -X 'PATCH' \
  'https://sandbox.restapi.hartphp.com.pl/v1/basket/{PositionId}' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...' \
  -d '{
    "quantity": 2
  }'
```

### Response Codes:

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found

## Removing an Item from the Basket

**DELETE** /v1/basket/{PositionId}

- **URL:** /v1/basket/{PositionId}
- **Method:** DELETE
- **Description:** Removes a selected item from the basket
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - PositionId – Basket position ID
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** Confirmation of removing an item from the basket

```
curl -X 'DELETE' \
'https://sandbox.restapi.hartphp.com.pl/v1/basket/{PositionId}' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

### Response Codes:

- **200 OK**
- **400 Bad Request**
- **401 Unauthorized**
- **404 Not Found**

## Orders

The orders section handles processes related to order management in the system. It includes creating, modifying, viewing, and deleting orders.

### Order List

#### GET /v1/orders

- **URL:** /v1/orders
- **Method:** GET
- **Description:** Retrieves a collection of orders
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Page – Page number
    - Size – Number of records per page
    - SortDirection – Sorting order: [ASC/DESC]
- **Response:** JSON containing a collection of orders
  - **orderId** – order ID
  - **branchName** – branch name
  - **branchId** – branchID
  - **quantity** – quantity
  - **orderType** – order type
  - **productId** – product ID
  - **hartCode** – HART product code
  - **price** – price
  - **currency** – currency
  - **message**

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/orders' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- **200 OK**

```
{
  "items": [
    {
      "orderId": number,
      "branchName": "string",
      "branchId": number,
      "quantity": number,
      "orderType": "string",
      "productId": number,
      "hartCode": "string",
      "price": number,
      "currency": "string",
      "message": "string"
    }
  ],
  "total_pages": number,
  "current_page": number,
  "items_from": number,
  "items_to": number,
  "total_items_count": number
}
```

- **400 Bad Request**
- **401 Unauthorized**

## Order Details

### GET /v1/orders/{OrderId}

- **URL:** /v1/orders/{orderId}
- **Method:** GET
- **Description:** Retrieves order details by orderId
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - OrderId – Order identifier
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** JSON with order details
  - **orderId** – order ID
  - **branchName** – branch name
  - **branchId** – branchID
  - **quantity** – quantity
  - **orderType** – order type
  - **productId** – product ID
  - **hartCode** – HART product code
  - **price** – price
  - **currency** – currency
  - **message**

```
curl -X 'GET' \
'https://sandbox.restapi.hartphp.com.pl/v1/orders/{orderId}' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXZWQ6Iiwia2lkIiA6ICJfY0RqV...'
```

### Response Codes:

- **200 OK**

```
{
  "orderId": number,
  "branchName": "string",
  "branchId": number,
  "quantity": number,
  "orderType": "string",
  "productId": number,
  "hartCode": "string",
  "price": number,
  "currency": "string",
  "message": "string"
```

}

- **400 Bad Request**
- **401 Unauthorized**
- **404 Not Found**

## Creating and Order

### POST /v1/orders

- **URL:** /v1/orders
- **Method:** POST
- **Description:** Creates a new order in the system
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **BODY**
    - basketPositionIds – List of basket position IDs for the order
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** JSON containing the details of the created order
  - **orderId** – order ID
  - **branchName** – branch name
  - **branchId** – branchID
  - **quantity** – quantity
  - **orderType** – order type
  - **productId** – product ID
  - **hartCode** – HART product code
  - **price** – price
  - **currency** – currency
  - **message**
  - **replacementCosts** – replacement cost
  - **isSuccess** – is success of adding an order

```
curl -X 'POST' \
  'https://sandbox.restapi.hartphp.com.pl/v1/orders' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpzZW50L3N1bWU6Iiwia2lkIiA6ICJfY0RqV...' \
  -d '{
    "basketPositionIds": []
  }'
```

### Response Codes:

- **200 OK**

```
{
  "value": [
    {
      "orderId": number,
```



```
    "branchName": "string",  
    "branchId": number,  
    "quantity": number,  
    "orderType": "string",  
    "productId": number,  
    "hartCode": "string",  
    "price": number,  
    "currency": "string",  
    "message": "string",  
    "replacementCosts": number,  
    "isSuccess": boolean  
  }  
],  
  "isSuccess": boolean,  
  "errorMessage": "string",  
}
```

- **400 Bad Request**
- **401 Unauthorized**
- **404 Not Found**

## Deleting an Order

**DELETE** /v1/orders/{OrderId}

- **URL:** /v1/orders
- **Method:** DELETE
- **Description:** Deletes an order by its identifier
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - OrderId – Order identifier
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** Confirmation of order deletion

```
curl -X 'DELETE' \
'https://sandbox.restapi.hartphp.com.pl/v1/orders/123456789' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

### Response Codes:

- 200 OK
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found

## Documents

The documents section provides access to sales documents, including invoices, corrections, and WZ documents. The response to requests includes the form of payment code (**formOfPaymentCode**). Below is a list of all available codes:

- **1** – Paid in cash
- **2** – Deferred - transfer
- **3** – Cash on delivery
- **4** – Prepayment - transfer
- **5** – Deferred - transfer (no interest)
- **6** – Deferred cash payable on hand
- **7** – POMOST - Support program
- **8** – Deferred limit - transfer
- **9** – Deferred recurring - transfer
- **0** – To pay 0
- **P** – PWS - Support program

## Invoice Collection

### GET /v1/documents/invoices

- **URL:** /v1/documents/invoices
- **Method:** GET
- **Description:** Retrieves a collection of invoices
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **QUERY**
    - DateFrom – Start date, format [yyyy-MM-dd]
    - DateTo – End date, format [yyyy-MM-dd]
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Page – Page number
    - Size – Number of records per page
    - SortDirection – Sorting order [ASC/DESC]
- **Response:** Invoice collection divided into pages
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date

- **paymentDate** – payment date
- **soldDate** – sold date
- **positionsCount** – number of items in document
- **amountGross** – gross amount
- **amountNet** – net amount
- **currencyCode** – currency
- **formOfPaymentCode** – payment code
- **seller** – seller data
  - **hartBranchId** – branch ID
  - **hartBranchName** – branch name
  - **taxPayerId** – tax payer ID
  - **regon** – REGON number
- **customer** – client data
  - **id** – client ID
  - **code** – client code
  - **name** – client name
  - **branchName** – branch name
  - **taxPayerId** – tax payer ID
  - **regon** – REGON number
  - **zipCode** – zip code
  - **city** – city
  - **street** – address
  - **countryCode** – country code
- **positions** – list of items in document
  - **productId** – product ID
  - **productCode** – product code
  - **positionNumer** – position number
  - **productName** – product name
  - **quantity** – quantity
  - **priceNetto** – net price
  - **taxRate** – tax rate

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/invoices?DateFrom=2024-09-01&DateTo=2024-09-30' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXZW4iLCJ1a2kiOiA6ICJfY0RqV...'
```

#### Response Codes:

- 200 OK

```
{
  "items": [
    {
      "header": {
        "documentNr": "string",
        "createdDate": "string",
```

```
    "paymentDate": "string",
    "soldDate": "string",
    "positionsCount": number,
    "amountGross": number,
    "amountNet": number,
    "currencyCode": "string",
    "formOfPaymentCode": "string"
  },
  "seller": {
    "hartBranchId": number,
    "hartBranchName": "string",
    "taxPayerId": "string",
    "regon": "string"
  },
  "customer": {
    "id": "string",
    "code": "string",
    "name": "string",
    "branchName": "string",
    "taxPayerId": "string",
    "regon": "string",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "countryCode": "string"
  },
  "positions": [
    {
      "productId": "string",
      "productCode": "string",
      "positionNumer": number,
      "productName": "string",
      "quantity": number,
      "priceNetto": number,
      "taxRate": number
    }
  ]
},
"total_pages": number,
"current_page": number,
"items_from": number,
"items_to": number,
"total_items_count": number
}
```

- **400 Bad Request**
- **401 Unauthorized**

## Invoice

### **GET** /v1/documents/invoices/{DocumentNumber}

- **URL:** /v1/documents/invoices/{DocumentNumber}
- **Method:** GET
- **Description:** Retrieves invoice details by DocumentNumber
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - DocumentNumber – Invoice correction identifier, the "/" character in the document number should be replaced with the corresponding code "%2F"
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** JSON with invoice details
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date
    - **paymentDate** – payment date
    - **soldDate** – sold date
    - **positionsCount** – number of items in document
    - **amountGross** – gross amount
    - **amountNet** – net amount
    - **currencyCode** – currency
    - **formOfPaymentCode** – payment code
  - **seller** – seller data
    - **hartBranchId** – branch ID
    - **hartBranchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
  - **customer** – client data
    - **id** – client ID
    - **code** – client code
    - **name** – client name
    - **branchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
    - **zipCode** – zip code
    - **city** – city
    - **street** – address

- **countryCode** – country code
- **positions** – list of items in document
  - **productId** – product ID
  - **productCode** – product code
  - **positionNumer** – position number
  - **productName** – product name
  - **quantity** – quantity
  - **priceNetto** – net price
  - **taxRate** – tax rate

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/invoices/31%2F01%2F2024%2F0074077' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- 200 OK

```
{
  "header": {
    "documentNr": "string",
    "createdDate": "string",
    "paymentDate": "string",
    "soldDate": "string",
    "positionsCount": number,
    "amountGross": number,
    "amountNet": number,
    "currencyCode": "string",
    "formOfPaymentCode": "string"
  },
  "seller": {
    "hartBranchId": number,
    "hartBranchName": "string",
    "taxPayerId": "string",
    "regon": "string"
  },
  "customer": {
    "id": "string",
    "code": "string",
    "name": "string",
    "branchName": "string",
    "taxPayerId": "string",
    "regon": "string",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "countryCode": "string"
  },
  "positions": [
    {
      "productId": "string",
      "productCode": "string",
      "positionNumer": number,
      "productName": "string",
      "quantity": number,
```

```
    "priceNetto": number,  
    "taxRate": number  
  }  
]  
}
```

- **400 Bad Request**
- **401 Unauthorized**
- **404 Not Found**



## Invoice Corrections Collection

### GET /v1/documents/invoice-corrections

- **URL:** /v1/documents/invoice-corrections
- **Method:** GET
- **Description:** Retrieves a collection of invoice corrections
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **QUERY**
    - DateFrom – Start date, format [yyyy-MM-dd]
    - DateTo – End date, format [yyyy-MM-dd]
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Page – Page number
    - Size – Number of records per page
    - SortDirection – Sorting order [ASC/DESC]
- **Response:** Collection of Invoice Corrections divided into pages
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date
    - **paymentDate** – payment date
    - **soldDate** – sold date
    - **positionsCount** – number of items in document
    - **amountGross** – gross amount
    - **amountNet** – net amount
    - **currencyCode** – currency
    - **formOfPaymentCode** – payment code
  - **seller** – seller data
    - **hartBranchId** – branch ID
    - **hartBranchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
  - **customer** – client data
    - **id** – client ID
    - **code** – client code
    - **name** – client name
    - **branchName** – branch name
    - **taxPayerId** – tax payer ID

- **regon** – REGON number
- **zipCode** – zip code
- **city** – city
- **street** – address
- **countryCode** – country code
- **positions** – list of items in document
  - **productId** – product ID
  - **productCode** – product code
  - **positionNumer** – position number
  - **productName** – product name
  - **salesDocument** – sales document
  - **quantityBefore** – quantity before
  - **quantityAfter** – quantity after
  - **priceNettoBefore** – net price before
  - **priceNettoAfter** – net price after
  - **taxRate** – tax rate

```
curl -X 'GET' \
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/invoice-corrections?DateFrom=2024-08-01&DateTo=2024-08-31' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpzZW50a3VzIiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- 200 OK

```
{
  "items": [
    {
      "header": {
        "documentNr": "string",
        "createdDate": "string",
        "paymentDate": "string",
        "soldDate": "string",
        "positionsCount": number,
        "amountGross": number,
        "amountNet": number,
        "currencyCode": "string",
        "formOfPaymentCode": "string"
      },
      "seller": {
        "hartBranchId": number,
        "hartBranchName": "string",
        "taxPayerId": "string",
        "regon": "string"
      },
      "customer": {
        "id": "string",
        "code": "string",
        "name": "string",
        "branchName": "string",
        "taxPayerId": "string",

```

```
    "regon": "string",
    "zipCode": "string",
    "city": "string",
    "street": "string",
    "countryCode": "string"
  },
  "positions": [
    {
      "productId": "string",
      "productCode": "string",
      "positionNumber": number,
      "productName": "string",
      "salesDocument": "string",
      "quantityBefore": number,
      "quantityAfter": number,
      "priceNetBefore": number,
      "priceNetAfter": number,
      "taxRate": number
    }
  ]
},
"total_pages": number,
"current_page": number,
"items_from": number,
"items_to": number,
"total_items_count": number
}
```

- **400 Bad Request**
- **401 Unauthorized**

## Invoice Correction

**GET** /v1/documents/invoice-corrections/{DocumentNumber}

- **URL:** /v1/documents/invoice-corrections/{DocumentNumber}
- **Method:** GET
- **Description:** Retrieves invoice correction details by DocumentNumber
- **Required Parameters**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - DocumentNumber – Invoice correction identifier, the "/" character in the document number should be replaced with the corresponding code "%2F"
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** JSON with invoice correction details
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date
    - **paymentDate** – payment date
    - **soldDate** – sold date
    - **positionsCount** – number of items in document
    - **amountGross** – gross amount
    - **amountNet** – net amount
    - **currencyCode** – currency
    - **formOfPaymentCode** – payment code
  - **seller** – seller data
    - **hartBranchId** – branch ID
    - **hartBranchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
  - **customer** – client data
    - **id** – client ID
    - **code** – client code
    - **name** – client name
    - **branchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
    - **zipCode** – zip code
    - **city** – city
    - **street** – address

- **countryCode** – country code
- **positions** – list of items in document
  - **productId** – product ID
  - **productCode** – product code
  - **positionNumer** – position number
  - **productName** – product name
  - **salesDocument** – sales document
  - **quantityBefore** – quantity before
  - **quantityAfter** – quantity after
  - **priceNettoBefore** – net price before
  - **priceNettoAfter** – net price after
  - **taxRate** – tax rate

```
curl -X 'GET' \  
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/invoice-correc-  
tions/33%2F01%2F2024%2F0013695' \  
  -H 'Content-Type: application/json' \  
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- **200 OK**

```
{  
  "header": {  
    "documentNr": "string",  
    "createdDate": "string",  
    "paymentDate": "string",  
    "soldDate": "string",  
    "positionsCount": number,  
    "amountGross": number,  
    "amountNet": number,  
    "currencyCode": "string",  
    "formOfPaymentCode": "string"  
  },  
  "seller": {  
    "hartBranchId": number,  
    "hartBranchName": "string",  
    "taxPayerId": "string",  
    "regon": "string"  
  },  
  "customer": {  
    "id": "string",  
    "code": "string",  
    "name": "string",  
    "branchName": "string",  
    "taxPayerId": "string",  
    "regon": "string",  
    "zipCode": "string",  
    "city": "string",  
    "street": "string",  
    "countryCode": "string"  
  },  
  "positions": [  
    {  
      "productId": "string",
```

```
"productCode": "string",  
"positionNumer": number,  
"productName": "string",  
"salesDocument": "string",  
"quantityBefore": number,  
"quantityAfter": number,  
"priceNetBefore": number,  
"priceNetAfter": number,  
"taxRate": number  
}  
]  
}
```

- **400 Bad Request**
- **401 Unauthorized**
- **404 Not Found**

## Delivery Notes Collection (WZ)

### GET /v1/documents/delivery-notes

- **URL:** /v1/documents/delivery-notes
- **Method:** GET
- **Description:** Retrieves a collection of delivery notes (WZ)
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **QUERY**
    - DateFrom – Start date, format [yyyy-MM-dd]
    - DateTo – End date, format [yyyy-MM-dd]
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Page – Page number
    - Size – Number of records per page
    - SortDirection – Sorting order [ASC/DESC]
- **Response:** Collection of Delivery Notes Collection (WZ) divided into pages
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date
    - **paymentDate** – payment date
    - **soldDate** – sold date
    - **positionsCount** – number of items in document
    - **amountGross** – gross amount
    - **amountNet** – net amount
    - **currencyCode** – currency
    - **formOfPaymentCode** – payment code
  - **seller** – seller data
    - **hartBranchId** – branch ID
    - **hartBranchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
  - **customer** – client data
    - **id** – client ID
    - **code** – client code
    - **name** – client name
    - **branchName** – branch name
    - **taxPayerId** – tax payer ID

- **regon** – REGON number
- **zipCode** – zip code
- **city** – city
- **street** – address
- **countryCode** – country code
- **positions** – list of items in document
  - **productId** – product ID
  - **productCode** – product code
  - **positionNumer** – position number
  - **productName** – product name
  - **quantity** – quantity
  - **priceNetto** – net price
  - **taxRate** – tax rate

```
curl -X 'GET' \  
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/delivery-notes?DateFrom=2024-09-01&DateTo=2024-10-30' \  
  -H 'Content-Type: application/json' \  
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- **200 OK**

```
{  
  "items": [  
    {  
      "header": {  
        "documentNr": "string",  
        "createdDate": "string",  
        "paymentDate": "string",  
        "soldDate": "string",  
        "positionsCount": number,  
        "amountGross": number,  
        "amountNet": number,  
        "currencyCode": "string",  
        "formOfPaymentCode": "string"  
      },  
      "seller": {  
        "hartBranchId": number,  
        "hartBranchName": "string",  
        "taxPayerId": "string",  
        "regon": "string"  
      },  
      "customer": {  
        "id": "string",  
        "code": "string",  
        "name": "string",  
        "branchName": "string",  
        "taxPayerId": "string",  
        "regon": "string",  
        "zipCode": "string",  
        "city": "string",  
        "street": "string",
```



```
    "countryCode": "string",
  },
  "positions": [
    {
      "productId": "string",
      "productCode": "string",
      "positionNumber": number,
      "productName": "string",
      "quantity": number,
      "priceNetto": number,
      "taxRate": number
    }
  ]
},
"total_pages": number,
"current_page": number,
"items_from": number,
"items_to": number,
"total_items_count": number
}
```

- **400 Bad Request**
- **401 Unauthorized**

## Delivery Note (WZ)

**GET** /v1/documents/delivery-notes/{DocumentNumber}

- **URL:** /v1/documents/delivery-notes/{DocumentNumber}
- **Method:** GET
- **Description:** Retrieves details of a delivery note (WZ) identified by DocumentNumber
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - DocumentNumber – document number, the "/" character in the document number should be replaced with the corresponding code "%2F"
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** JSON with details of delivery note (WZ)
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date
    - **paymentDate** – payment date
    - **soldDate** – sold date
    - **positionsCount** – number of items in document
    - **amountGross** – gross amount
    - **amountNet** – net amount
    - **currencyCode** – currency
    - **formOfPaymentCode** – payment code
  - **seller** – seller data
    - **hartBranchId** – branch ID
    - **hartBranchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
  - **customer** – client data
    - **id** – client ID
    - **code** – client code
    - **name** – client name
    - **branchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
    - **zipCode** – zip code
    - **city** – city
    - **street** – address

- **countryCode** – country code
- **positions** – list of items in document
  - **productId** – product ID
  - **productCode** – product code
  - **positionNumer** – position number
  - **productName** – product name
  - **quantity** – quantity
  - **priceNetto** – net price
  - **taxRate** – tax rate

```
curl -X 'GET' \  
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/delivery-  
notes/28%2F01%2F2024%2F0384191' \  
  -H 'Content-Type: application/json' \  
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- 200 OK

```
{  
  "header": {  
    "documentNr": "string",  
    "createdDate": "string",  
    "paymentDate": "string",  
    "solddDate": "string",  
    "positionsCount": number,  
    "amountGross": number,  
    "amountNet": number,  
    "currencyCode": "string",  
    "formOfPaymentCode": "string"  
  },  
  "seller": {  
    "hartBranchId": number,  
    "hartBranchName": "string",  
    "taxPayerId": "string",  
    "regon": "string"  
  },  
  "customer": {  
    "id": "string",  
    "code": "string",  
    "name": "string",  
    "branchName": "string",  
    "taxPayerId": "string",  
    "regon": "string",  
    "zipCode": "string",  
    "city": "string",  
    "street": "string",  
    "countryCode": "string"  
  },  
  "positions": [  
    {  
      "productId": "string",  
      "productCode": "string",  
      "positionNumer": number,  
      "productName": "string",  
      "quantity": number,  
      "priceNetto": number,  
      "taxRate": number
```

```
"quantity": number,  
  "priceNetto": number,  
  "taxRate": number  
}  
]  
}
```

- **400 Bad Request**
- **401 Unauthorized**
- **404 Not Found**

## Bonus Correction Invoice Collection

### **GET** /v1/documents/bonus-corrections

- **URL:** /v1/documents/bonus-corrections
- **Method:** GET
- **Description:** Retrieves a collection of bonus correction invoices
- **Required Parameters:**
  - **HEADER**
    - Authorization – JWT token prefixed with "Bearer"
  - **QUERY**
    - DateFrom – Start date, format [yyyy-MM-dd]
    - DateTo – End date, format [yyyy-MM-dd]
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
  - **QUERY**
    - Page – Page number
    - Size – Number of records per page
    - SortDirection – Sorting order [ASC/DESC]
- **Response:** Collection of Bonus Correction Invoice divided into pages
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date
    - **paymentDate** – payment date
    - **soldDate** – sold date
    - **positionsCount** – number of items in document
    - **amountGross** – gross amount
    - **amountNet** – net amount
    - **currencyCode** – currency
    - **formOfPaymentCode** – payment code
  - **seller** – seller data
    - **hartBranchId** – branch ID
    - **hartBranchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
  - **customer** – client data
    - **id** – client ID
    - **code** – client code
    - **name** – client name
    - **branchName** – branch name
    - **taxPayerId** – tax payer ID

- **regon** – REGON number
- **zipCode** – zip code
- **city** – city
- **street** – address
- **countryCode** – country code
- **positions** – list of items in document
  - **positionNumer** – position number
  - **productName** – product name
  - **quantity** – quantity
  - **priceNetto** – net price
  - **taxRate** – tax rate

```
curl -X 'GET' \  
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/bonus-corrections?DateFrom=2024-09-01&DateTo=2024-10-31' \  
  -H 'Content-Type: application/json' \  
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqV...'
```

#### Response Codes:

- 200 OK

```
{  
  "items": [  
    {  
      "header": {  
        "documentNr": "string",  
        "createdDate": "string",  
        "paymentDate": "string",  
        "soldDate": "string",  
        "positionsCount": number,  
        "amountGross": number,  
        "amountNet": number,  
        "currencyCode": "string",  
        "formOfPaymentCode": "string"  
      },  
      "seller": {  
        "hartBranchId": number,  
        "hartBranchName": "string",  
        "taxPayerId": "string",  
        "regon": "string"  
      },  
      "customer": {  
        "id": "string",  
        "code": "string",  
        "name": "string",  
        "branchName": "string",  
        "taxPayerId": "string",  
        "regon": "string",  
        "zipCode": "string",  
        "city": "string",  
        "street": "string",  
        "countryCode": "string"  
      },  
      "positions": [  
        {  
          "positionNumer": number,  
          "productName": "string",  
          "quantity": number,  
          "priceNetto": number,  
          "taxRate": number
```

```
{
  "positionNumber": number,
  "productName": "string",
  "quantity": number,
  "priceNetto": number,
  "taxRate": number
}
]
},
"total_pages": number,
"current_page": number,
"items_from": number,
"items_to": number,
"total_items_count": number
}
```

- **400 Bad Request**
- **401 Unauthorized**

## Bonus Correction Invoice

**GET** /v1/documents/bonus-corrections/{DocumentNumber}

- **URL:** /v1/documents/bonus-corrections/{DocumentNumber}
- **Method:** GET
- **Description:** Retrieves details of a bonus correction invoice identified by DocumentNumber
- **Required Parameters:**
  - **HEADER0**
    - Authorization – JWT token prefixed with "Bearer"
  - **ROUTE**
    - DocumentNumber – document number, the "/" character in the document number should be replaced with the corresponding code "%2F"
- **Optional Parameters:**
  - **HEADER**
    - BranchId – Branch ID
- **Response:** JSON with details of bonus correction invoice
  - **header** – document data
    - **documentNr** – document number
    - **createdDate** – created date
    - **paymentDate** – payment date
    - **soldDate** – sold date
    - **positionsCount** – number of items in document
    - **amountGross** – gross amount
    - **amountNet** – net amount
    - **currencyCode** – currency
    - **formOfPaymentCode** – payment code
  - **seller** – seller data
    - **hartBranchId** – branch ID
    - **hartBranchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
  - **customer** – client data
    - **id** – client ID
    - **code** – client code
    - **name** – client name
    - **branchName** – branch name
    - **taxPayerId** – tax payer ID
    - **regon** – REGON number
    - **zipCode** – zip code
    - **city** – city
    - **street** – address



- **countryCode** – country code
- **positions** – list of items in document
  - **positionNumer** – position number
  - **productName** – product name
  - **quantity** – quantity
  - **priceNetto** – net price
  - **taxRate** – tax rate

```
curl -X 'GET' \  
  'https://sandbox.restapi.hartphp.com.pl/v1/documents/bonus-correc-  
tions/52%2F01%2F2024%2F0000196' \  
  -H 'Content-Type: application/json' \  
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJfY0RqVz...'
```

#### Response Codes:

- **200 OK**

```
{  
  "header": {  
    "documentNr": "string",  
    "createdDate": "string",  
    "paymentDate": "string",  
    "soldDate": "string",  
    "positionsCount": number,  
    "amountGross": number,  
    "amountNet": number,  
    "currencyCode": "string",  
    "formOfPaymentCode": "string"  
  },  
  "seller": {  
    "hartBranchId": number,  
    "hartBranchName": "string",  
    "taxPayerId": "string",  
    "regon": "string"  
  },  
  "customer": {  
    "id": "string",  
    "code": "string",  
    "name": "string",  
    "branchName": "string",  
    "taxPayerId": "string",  
    "regon": "string",  
    "zipCode": "string",  
    "city": "string",  
    "street": "string",  
    "countryCode": "string"  
  },  
  "positions": [  
    {  
      "positionNumer": number,  
      "productName": "string",  
      "quantity": number,  
      "priceNetto": number,  
      "taxRate": number  
    }  
  ]  
}
```

}

- **400 Bad Request**
- **401 Unauthorized**
- **404 Not Found**

## Ordering workflow

The workflow for ordering goods from HART using the API. Below is a step-by-step guide on how to utilize the available endpoints to efficiently handle the process:

### 1. Checking Product Availability and Details

Before placing an order, the first step involves verifying the availability and details of the products in the warehouse. This can be done using the following endpoints:

- [GET /v1/products](#) – use this endpoint to fetch the list of products and their detailed information, such as product name, description, price, and other attributes,
- [GET /v1/products/availability](#) – use this endpoint to check the stock levels and availability of specific products. Ensure to query this endpoint to avoid adding unavailable products to the cart.

### 2. Managing Goods in the Basket

Once the desired products are identified and confirmed to be available, the next step is to manage them in the shopping cart. This includes adding, updating, or removing products. The following endpoint should be used:

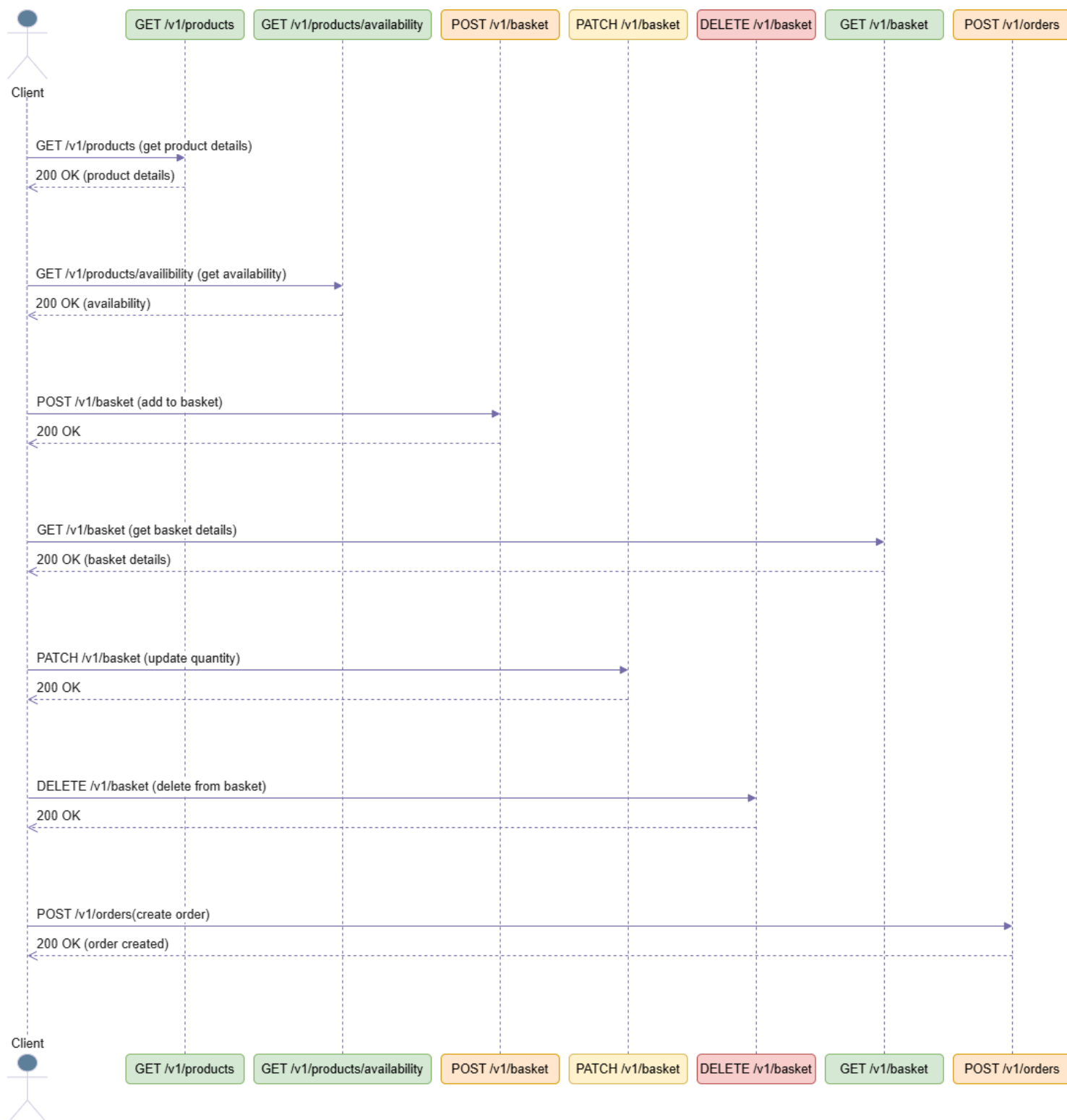
- [POST /v1/basket](#) – add products to the basket,
- [PATCH /v1/basket](#) – update quantities for products already in the basket,
- [DELETE /v1/basket](#) – remove products from the basket,
- [GET /v1/basket](#) – retrieve the current state of the basket.

Make sure the basket contains correct product details and quantities before proceeding to order creation.

### 3. Creating an Order

After finalizing the basket, you can create an order using:

- [POST /v1/orders](#) – utilize this endpoint to submit the basket as an order for processing.



## Usage Limits

The HART API has defined usage limits to ensure high availability and service stability for all clients. This section details the limits and how they can be monitored.

### API Call Restrictions

The system enforces the following rate limits for API calls:

- Requests per minute: 50 requests per client

These limits apply globally to all clients and endpoints. Once the limit is reached, subsequent requests will be temporarily rejected until the limit resets.

### Monitoring Usage Limits

API usage is monitored through the following HTTP headers, which provide real-time information on request limits for users:

- X-RateLimit-Limit: Indicates the maximum number of allowed requests within a defined time window.
- X-RateLimit-Remaining: Indicates how many requests are remaining in the current time window.
- X-RateLimit-Reset: Specifies the time (in seconds) until the request counter resets.

### Exceeding Limits

If the usage limit is exceeded:

- The system will automatically reject additional requests.
- The system will return an HTTP status code: 429 Too Many Requests.
- Limits are reset automatically after the specified time interval.
- No additional actions are required from the user.

### Best Practices

To avoid reaching the limits, we recommend:

- Implementing caching mechanisms for frequently accessed data.
- Grouping requests where possible.
- Implementing detection and handling of limit exceedance in client-side code.
- Monitoring API usage patterns within the client's system.

## Support

If the predefined limits are insufficient for business needs, clients should contact their assigned sales representative to discuss possible solutions.

### **Additional Notes**

HART reserves the right to modify these limits in the future. Any changes will be communicated to clients in advance through their sales representatives.

## Security and Compliance

### Data Security Measures

HART prioritizes data security and protection. All data transmitted via the API is secured using the HTTPS protocol, ensuring encrypted communication between client systems and HART servers.

Communication with the API requires authentication using JWT tokens, providing an additional layer of security and access control. Detailed information on the authentication process is available in the "Authentication and Authorization" section.

### Compliance Standards

HART has aligned its processes and infrastructure with the requirements of:

- General Data Protection Regulation (GDPR)
- ISO 27001:2015 standard for information security management systems

To comply with these standards, the company has implemented comprehensive procedures and technical safeguards for personal data processing and information security management.

### Personal Data Protection

HART ensures compliance with GDPR requirements regarding personal data processing. Any questions regarding personal data processing can be directed to: [rodo@hartphp.com.pl](mailto:rodo@hartphp.com.pl).

Clients utilizing the API should ensure their systems and processes comply with GDPR, particularly in terms of:

- Secure data storage
- Processing data only to the necessary extent
- Adhering to data retention policies
- Implementing appropriate technical and organizational safeguards

### Reporting Security Incidents

If a potential security incident or vulnerability is detected, clients should:

- Immediately send detailed information to [hart@hartphp.com.pl](mailto:hart@hartphp.com.pl)
- Simultaneously inform their assigned sales representative

A security incident report should include:

- A detailed description of the identified issue
- Date and time of detection
- Potential impact on data security
- Contact details of the reporting individual

- Any additional information that may help assess and resolve the issue

HART is committed to promptly responding to security-related reports and keeping all involved parties informed about the actions taken and the resolution status.

### **Audit and Monitoring**

HART conducts regular security audits of its infrastructure and systematically monitors all systems for potential threats. This allows for rapid identification and response to any security concerns.

### **Security Updates**

Information on significant security updates will be communicated to clients via their assigned sales representatives. In the case of critical security updates, clients may receive additional notifications through agreed communication channels.



## Service Level Agreement (SLA)

### Service Availability

HART strives to provide the highest quality and availability of API services. Our infrastructure is continuously monitored and maintained by a qualified team. However, due to the complexity of IT systems and factors beyond our control, we cannot guarantee uninterrupted service availability.

The API is provided on an "as is" basis, without any express or implied warranties regarding availability, performance, or fitness for a particular purpose.

### System Performance

We make every effort to ensure that the system operates efficiently and responsively. The infrastructure is continuously monitored for performance and optimized as needed. However, response times may vary depending on several factors, including:

- Current system load
- Complexity of executed operations
- Internet connection quality
- External factors affecting network performance

### Technical Support

Technical support is provided by a dedicated team through the assigned sales representative. While we strive to respond to inquiries as quickly as possible, we do not guarantee specific response times. The response time may depend on multiple factors, including current support team workload and the complexity of the reported issue.

### Limitation of Liability

HART is not responsible for any direct, indirect, incidental, special, or consequential damages resulting from the use or inability to use the API, including but not limited to:

- Data loss
- Loss of profits
- Business interruptions
- Other financial losses

Clients use the API at their own risk and are responsible for implementing appropriate precautions, such as:

- Regularly backing up data
- Implementing error-handling mechanisms

- Monitoring their own systems
- Testing integrations in a sandbox environment before deploying to production

### **Maintenance Work**

HART reserves the right to perform necessary maintenance and system updates. In the case of planned maintenance that may impact service availability, we will make an effort to notify clients in advance via their sales representatives.

### **Service Modifications**

HART reserves the right to modify, suspend, or discontinue API services at any time without prior notice. For significant changes, we will endeavor to provide clients with advance notice.

### **Monitoring and Reporting**

Although we conduct continuous monitoring of our systems, we do not provide detailed performance metrics or availability reports. In the event of major incidents affecting service operation, we will inform clients about the situation and any corrective actions taken via their sales representatives.

## Production and Test Environment (Sandbox)

### Production Environment

<https://restapi.hartphp.com.pl>

The production environment is the final destination where the fully functional API operates for real business transactions. All data in this environment is real, and its configuration is optimized for performance, security, and reliability.

### Addressing and Access

1. **API URL**

<https://restapi.hartphp.com.pl>

All methods operate under version v1.0. The API version must be specified in the address, e.g. */v1/basket*

2. **Communication Protocol**

All connections are secured using TLS/HTTPS encryption. Only encrypted connections are accepted; unencrypted HTTP requests will be rejected.

3. **Authorization**

Each API request requires the inclusion of a JWT authorization token. To obtain an access token, the */v1/auth* endpoint must be called with the provided credentials.

### Technical Requirements

1. **HTTP Request Headers**

During integration, adhere to the standard data transmission guidelines:

- Authorization: Header containing the JWT token in the format:  
Authorization: Bearer [TOKEN]
- Content-Type: All requests must have the header set to:  
Content-Type: application/json

2. **Versioning Handling**

The API version is defined in the address (*/v1/[endpoint]*)

3. **Security and Limits**

- Rate limiting
  - Maximum of 50 requests per minute
- Token Expiration: JWT tokens expire after 5 hours and require regeneration

## Test Environment (Sandbox)

<https://sandbox.restapi.hartphp.com.pl>

The test environment is provided to allow clients to integrate with our API under conditions similar to production. This enables clients to verify API calls, test system functions, and ensure proper implementation on their end. The data used in the test environment is simulated and independent of real production data.

### Addressing and Access

1. **API URL**

<https://sandbox.restapi.hartphp.com.pl>

All methods operate under version v1.0. The API version must be specified in the address, e.g.,  
/v1/basket

2. **Communication Protocol**

All connections are secured using TLS/HTTPS encryption. Only encrypted connections are accepted; unencrypted HTTP requests will be rejected.

3. **Authorization**

Each API request requires the inclusion of a JWT authorization token. To obtain an access token, the /v1/auth endpoint must be called with the provided credentials.

### Technical Requirements

1. **HTTP Request Headers**

During integration, adhere to the standard data transmission guidelines:

- Authorization: Header containing the JWT token in the format:  
Authorization: Bearer [TOKEN]
- Content-Type: All requests must have the header set to:  
Content-Type: application/json

2. **Versioning Handling**

The API version is defined in the address (/v1/[endpoint])

3. **Security and Limits**

- Rate limiting
  - Maximum of 50 requests per minute
- Token Expiration: JWT tokens expire after 5 hours and require regeneration

## Support and Communication Channels

If you encounter technical issues or have questions regarding API functionality, your first point of contact is the assigned sales representative. To ensure quick and effective assistance, it is necessary to provide complete information about the encountered problem.

### Reporting Issues

When reporting technical issues, please prepare the following information:

- A detailed description of the problem along with the steps leading to its occurrence
- API response identifier (if available)
- Date and time when the issue occurred
- System logs containing:
  - Request details (with sensitive data removed)
  - Received response
  - Error messages
- Indicate whether the issue occurs in the test or production environment
- Specify if the problem is repeatable and under what circumstances

### Issue Resolution Process

Once a report is received, the sales representative will forward it to the appropriate technical support team. Depending on the nature and complexity of the issue, additional information or further testing may be required.

To expedite issue resolution, it is recommended to perform basic diagnostics before reporting:

- Verify the accuracy of authentication data
- Check the status of the API environment on the status page
- Confirm that the issue is not related to local infrastructure

### Communication of Changes and Updates

Information about planned API changes, updates, or maintenance work will be communicated through the sales representative. For significant changes impacting integration, advance notice will be provided to allow preparation.

## Legal and Regulatory Information

### I. Definitions

The following terms used in this section shall have the following meanings:

- a. Service Provider/HART – HART Spółka z ograniczoną odpowiedzialnością (Limited Liability Company) with its registered office in Opole (45-531), at ul. Morcinka 43, registered in the Register of Entrepreneurs maintained by the District Court in Opole, 8th Commercial Division of the National Court Register under KRS number: 0000170602, NIP: 7542748922, REGON: 532402005, BDO: 000016760, a large enterprise as defined by the Act of March 8, 2013, on counteracting excessive delays in commercial transactions, with a share capital of PLN 123,320,000.00;
- b. User – a customer registered in the HART customer database, assigned an individual client ID number according to the applicable HART policies;
- c. Service/API (Application Programming Interface) – an application programming interface that enables the User's system to integrate with HART software, facilitating automatic and secure data exchange and functionality access.
- d. Regulations – the regulations for providing electronic services described in this section "Legal and Regulatory Information".

### II. Terms of Service Provision

1. HART provides Users with API services (hereinafter referred to as "API" or "Service"), enabling the integration of the User's IT system with the HART system in areas including, but not limited to:
  - 1.1.1. Retrieving price information for a single reference
  - 1.1.2. Retrieving stock availability for a single reference
  - 1.1.3 Placing orders for references
  - 1.1.4 Accessing information about their order and related documents
  - 1.1.5 Other functionalities described in this documentationThe API service is available exclusively to logged-in Users. Access to the Service is possible only after the User logs into the Service/API.
2. Agreement The agreement for the provision of the API service is concluded at the moment HART provides the User with access credentials (login and password) to the Service/API. API access is granted through authentication using the provided credentials and a generated JWT token. The agreement may be terminated by either party by giving notice to the other party. HART reserves the right to deactivate access credentials upon termination of the agreement.
3. Upon conclusion of the agreement, HART grants the Client a free, non-exclusive, non-transferable, and time-limited license to use the application within the following fields of use: installation and use of the provided resources to obtain information.
4. The use of the Service requires a computer device with Internet access and compliance with the technical requirements specified in this documentation.
5. HART informs Users that failure to meet the specified minimum technical requirements may result in the inability to use the Services, difficulties in access, or improper functioning of the Service.
6. The Service is provided free of charge. The User bears the cost of Internet access and usage, in accordance with their agreement with their telecommunications operator.
7. HART will make every effort to ensure that the service is available 24/7 and returns accurate data. However, it reserves the right that returned results may not always reflect reality.
8. HART is not responsible for any malfunction of the Service or any damages or losses resulting from the use of the API.

9. HART secures data transmitted to or retrieved from the User and will take all necessary measures to prevent unauthorized disclosure. However, it is not liable for any data loss or disclosure, except as stipulated in Section III regarding personal data processing.
10. HART reserves the right to discontinue or modify the Service at any time (including immediate termination of the service agreement), particularly if the User violates or fails to comply with any provisions regarding the use of the Service.
11. HART reserves the right to discontinue or modify the Service at any time without providing a reason, with appropriate prior notice.
12. The User has no claims against HART in connection with the discontinuation of the Service.
13. Agreements concluded with Users and any disputes arising from legal relationships between these parties are subject to Polish law and the jurisdiction of Polish courts. Disputes related to the use of the Service and agreements concluded in connection with its use shall be settled by the court in Opole.

### III. Privacy Policy

14. HART declares that the personal data provided will be processed solely for the purpose of providing the Services, concluding and executing the Agreement, and fulfilling HART's legal obligations in accordance with:
  - a. Regulation (EU) 2016/679 of the European Parliament and of the Council of April 27, 2016, on the protection of natural persons regarding the processing of personal data and on the free movement of such data, repealing Directive 95/46/EC (General Data Protection Regulation - GDPR);
  - b. The Personal Data Protection Act of May 10, 2018 (Journal of Laws of 2019, item 1781); and in case circumstances require processing beyond this scope, HART commits to obtaining separate consent from the User.
15. For the purpose of using the Service, HART collects the following data of Users and their end customers: name, surname, address, tax identification number (NIP), device IP number, information on viewed references, order details, and other data necessary for the proper functioning of the service.
16. Personal data is processed for the following purposes and based on the following legal grounds:
  - a. To conclude and execute the agreement, ensuring proper service quality – for the duration of the contract and afterward based on Article 6(1)(b) GDPR (performance of a contract)
  - b. To fulfill legal obligations imposed on us, based on Article 6(1)(c) GDPR (legal obligation), such as:
    - considering and responding to complaints, which we will use for the period required by law to store data related to the provision of the Service;
    - for the duration of the performance of obligations or for the period required by law, in which we may suffer legal consequences of failure to perform the obligations imposed by law, e.g. receive a fine from state or local government offices, i.e. pursuant to Art. 6 sec. 1 f of the GDPR;
  - c. For direct marketing purposes (legitimate interest) – including sending electronic offers and commercial/promotional information (legal basis: our legitimate interest), for the duration of the contract or until consent is withdrawn or an objection is raised;
  - d. for the purpose of pursuing claims and defending (performance of the contract) – for the duration of the contract (legal basis: performance of the contract), until the expiry of the limitation period for claims resulting from the contracts between us, and in the event of pursuing claims by us – for the duration of the relevant proceedings (legal basis: our legitimate interest).
17. Personal data will not be transferred to third parties, particularly outside the European Economic Area (EEA) (comprising the European Union, Norway, Liechtenstein, and Iceland). If such a need arises during our cooperation, we will do so within the limits allowed by law.
18. In relation to the processing of personal data by HART, the User has the right to:
  - a. Access their personal data,

- b. Request deletion of personal data (right to be forgotten),
- c. Update/correct personal data,
- d. Transfer personal data,
- e. Restrict processing of personal data,
- f. Object to the processing of their personal data,

To exercise these rights, Users should contact us at [rodo@hartphp.com.pl](mailto:rodo@hartphp.com.pl). The deletion, transfer, restriction, or objection to the processing of data may impact the provision of services and may result in their improper performance.

19. The User also has the right to lodge a complaint with the President of the Personal Data Protection Office if the processing of personal data by HART violates legal provisions.
20. To protect data against unauthorized access or loss, the following security measures have been implemented:
  - 20.1. Server infrastructure is located in a secure room at the company headquarters
  - 20.2. API access is protected by a multi-level authentication system (access credentials and JWT token)
  - 20.3. Continuous monitoring of infrastructure by a qualified team of administrators
  - 20.4. Regular data backups
  - 20.5. Restricted access to server rooms to authorized personnel only
  - 20.6. System event logs are maintained to detect unauthorized access attempts
21. Detailed information regarding HART's data processing is available at: <https://www.hartphp.com.pl/polityka-prywatnosci>.

#### IV. Compliance Obligations

22. Using the service constitutes acceptance of the this Regulations.
23. The User assumes full responsibility for all data and content transmitted via the API.
24. The User is obliged to ensure that data and content comply with applicable legal regulations, including data protection requirements, and commits to adhering to security standards.
25. It is strictly prohibited to transmit unlawful content through the Service, particularly:
  - a. Content violating mandatory legal provisions;
  - b. Content infringing on the rights of others, including personal rights, intellectual property rights, trade secrets, and industrial property rights;
  - c. Content inciting the commission of a crime;
  - d. Content supporting or promoting radical social ideologies, especially any form of discrimination based on sexual orientation, religion, gender, age, disability, race, nationality, political or social beliefs, trade union membership, or ethnic origin;
26. If the User becomes aware that any content related to the operation of the Service is unlawful or violates the Regulations, they should report this to HART, particularly by sending an email to [hart@hartphp.com.pl](mailto:hart@hartphp.com.pl) or informing their assigned sales representative. If HART receives credible information or an official notice about unlawful content shared by the User, HART has the right to restrict access to such content.
27. The use of the Service in any way that disrupts or may disrupt the operation of the IT system or any of its components is strictly prohibited, including:
  - a. Unauthorized modifications;
  - b. Use of the service for purposes other than purchasing products;
  - c. Reverse engineering;
  - d. Attempts to bypass security measures;
  - e. Attempts to gain unauthorized access or interfere with API operations.



If HART obtains information regarding improper use of the Service in violation of the Regulations, HART has the right to deny the User access to the Service. The User is required to promptly report any detected security breaches or unauthorized access.

28. The use of the Service is only permitted under the conditions specified in the Regulations and must adhere to the documentation under the following conditions:

28.1. The User must use the Service as intended by its creators, as outlined in the documentation;

28.2. The User shall not share credentials or data obtained from HART with third parties

28.3. The User shall not use the Service in activities that may cause harm

28.4. The User shall not disclose operational details or login credentials to third parties and shall take care to prevent their accidental exposure;

28.5. The User has the right to submit a written complaint regarding the Service to the address specified in Section I or through their assigned sales representative. HART will respond to complaints via the email address provided by the User and inform them of further actions.

## Glossary and Acronyms

This section provides explanations of specialized terms, abbreviations, and concepts used in the API documentation. The aim is to facilitate understanding of the technical terminology and processes described in the documentation.

### Terminology and Acronyms:

- API (Application Programming Interface) - A programming interface that enables communication between different IT systems.
- JWT (JSON Web Token) - A standard for securely transmitting information between parties in JSON format.
- REST (Representational State Transfer) - A software architecture style defining a set of constraints and properties based on the HTTP protocol.
- HTTP (Hypertext Transfer Protocol) - A protocol for transmitting hypertext documents used on the World Wide Web.
- HTTPS (Hypertext Transfer Protocol Secure) - The HTTP protocol secured with encryption.
- JSON (JavaScript Object Notation) - A lightweight data exchange format based on JavaScript.
- OAuth - An open authorization standard used for secure access delegation.
- URL (Uniform Resource Locator) - A standardized format for addressing resources on the Internet.
- WZ - An external release document for goods.

### Concept Explanations:

- Authorization and Authentication - Authorization refers to granting permissions to specific resources, whereas authentication is the process of verifying a user's identity.
- Pagination - A method of dividing large data sets into smaller parts (pages) for more efficient management and transmission of data.
- Access Token - A unique string used to authorize API requests, generated after successful authentication.
- Endpoint - A specific API endpoint, defined as a URL where a particular functionality is accessible.
- HTTP Response Codes - Standardized numerical codes indicating the status of a request:
  - 2xx: Success (e.g., 200 OK)
  - 4xx: Client-side error (e.g., 401 Unauthorized)
  - 5xx: Server-side error
- Test Environment (Sandbox) - An isolated environment allowing secure API integration testing without affecting production data.
- Production Environment - The primary API environment used for real business operations.
- Rate Limiting - A mechanism that restricts the number of API requests a client can make within a specified time frame.
- Parallel Requests - The ability to execute multiple simultaneous API requests.
- Cache - A mechanism for storing frequently used data for faster access..

## Attachments

Attachment No. 1 – OpenAPI 3.x.x specification