



# Pubeasy Soap API – Productinformation retrieval

## Functional Overview for Retailer

tech-version: 1

document-version: 2.0

Date: 21.01.2026

Status: approval



## 1. Table of Contents

|                                                                  |    |
|------------------------------------------------------------------|----|
| 1. Table of Contents .....                                       | 1  |
| 2. Change History.....                                           | 4  |
| 3. Introduction.....                                             | 5  |
| 4. Hosting and URL .....                                         | 6  |
| 5. General differences to the legacy retailer API .....          | 6  |
| 6. General workflow.....                                         | 6  |
| 6.1. API versioning .....                                        | 6  |
| 6.2. API updates.....                                            | 7  |
| 7. General workflow.....                                         | 7  |
| 7.1. Retailer supplier relationship .....                        | 7  |
| 8. Technical definitions .....                                   | 8  |
| 8.1. Authentication methods .....                                | 8  |
| 8.2. Communication method.....                                   | 9  |
| 8.3. wsdl and endpoint for soap services.....                    | 9  |
| 8.3.1. Namespace definitions .....                               | 10 |
| 8.3.2. XML schema definition (XSD).....                          | 10 |
| 8.3.3. field definition.....                                     | 11 |
| 8.3.4. availability and status definitions .....                 | 13 |
| 9. Retailer soap services Overview .....                         | 13 |
| 9.1. partner identification .....                                | 13 |
| 9.2. PersonalProductServiceV1.....                               | 14 |
| 9.2.1. Function overview.....                                    | 14 |
| 9.2.1. findProducts.....                                         | 15 |
| 9.2.1.1. General Product information request - findProducts..... | 15 |
| 9.2.1.2. findProductsResponse .....                              | 15 |
| 9.2.2. getServiceInfo.....                                       | 16 |



|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| 9.3. Error and exception handling.....                                      | 17 |
| 9.3.1. Not a SOAP response .....                                            | 17 |
| 9.3.2. Universal soap fault .....                                           | 17 |
| 9.3.3. Error types .....                                                    | 18 |
| 9.3.3.1. exception message fields .....                                     | 19 |
| 9.3.3.2. Current MessageCode definition .....                               | 19 |
| 9.3.4. Authentication errors.....                                           | 21 |
| 9.3.5. http status type 400 error.....                                      | 21 |
| 9.3.1. Handling of http status codes .....                                  | 21 |
| 9.3.1.1. Wrong mailbox given in soap request .....                          | 21 |
| 9.3.2. http status 500 error.....                                           | 23 |
| 9.3.2.1. malformed XML request .....                                        | 23 |
| 9.3.2.2. Wrong data like malformed mailaddress or wrong Boolean flags ..... | 24 |
| 9.3.2.3. Malformed XML request – wrong sequence of elements .....           | 25 |
| 9.3.2.4. text values not allowed as values in text fields .....             | 26 |
| 9.3.3. http status 503 - Pubeasy soap server is not available.....          | 27 |
| 10. Soap examples.....                                                      | 27 |
| 10.1. PersonalProductService .....                                          | 27 |
| 10.1.1. findProducts.....                                                   | 27 |
| 10.1.1.1. findProducts request towards pubeasy .....                        | 27 |
| 10.1.1.2. findProductsResponse response from pubeasy .....                  | 28 |
| 10.1.2. getServiceInfo.....                                                 | 30 |
| 11. Test possibilities .....                                                | 30 |
| 11.1. Service info.....                                                     | 31 |
| 11.2. Tools for testing soap service .....                                  | 31 |
| 12. Procedure and tools to setup a test environment .....                   | 31 |
| 13. Appendix .....                                                          | 32 |
| 13.1. EDItEUR onix code lists .....                                         | 32 |
| 13.2. Code lists, mappings and field definitions.....                       | 32 |
| 13.2.1. Field definitions.....                                              | 32 |



|         |                                                          |    |
|---------|----------------------------------------------------------|----|
| 13.2.2. | Code list Mapping overview .....                         | 33 |
| 13.2.3. | productAvailability mapping between legacy and ONIX..... | 34 |
| 13.2.4. | Availability status codes and UI text .....              | 34 |

## 2. Change History

| Date       | Version | change               | comment |
|------------|---------|----------------------|---------|
| 10.12.2022 | 1.5     | Revision             |         |
| 21.01.2026 | 2.0     | Restuctured document |         |
|            |         |                      |         |
|            |         |                      |         |
|            |         |                      |         |
|            |         |                      |         |
|            |         |                      |         |
|            |         |                      |         |
|            |         |                      |         |
|            |         |                      |         |

### 3. Introduction

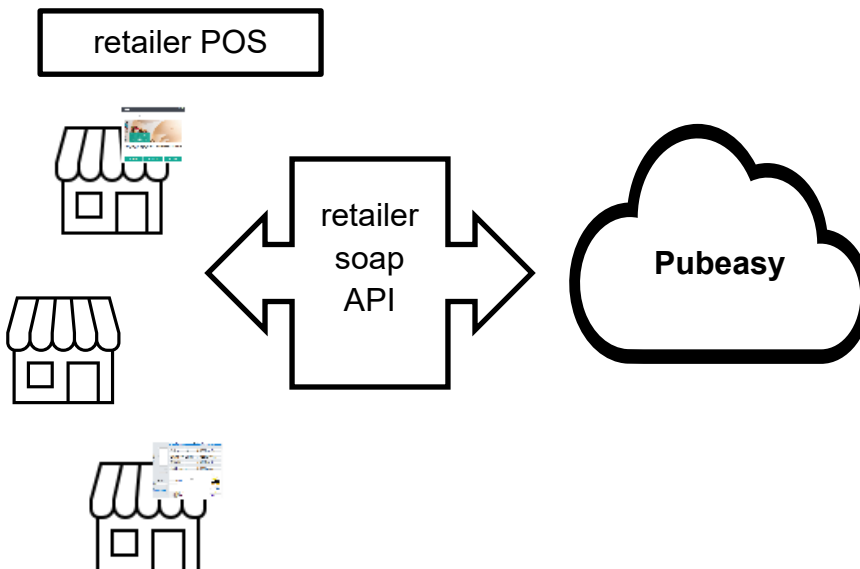
The Pubeasy service offers retailers in general an easy way to send orders to suppliers, request the order status and get personal price information. The orders can be placed by a retailer using different Pubeasy services – either the Pubeasy frontend or the Pubeasy soap api (via the retailers' POS software) and other methods.

*This document covers only the productinformation retrieval and describes no order functionality. The general new soap retailer API contains productioninformation as well as ordering functionalities. Therefore the productinformation API relies in some parts on references used for the complete retailer functionalities (ordering and productinfo).*

The Pubeasy API uses standardized Soap technology to exchange data between the retailer on one side and the Pubeasy platform.

On the retailer side the API can be implemented based on the soap description (WSDL) provided by Pubeasy.

Pubeasy acts as a service provider that receives productinformation requests from the retailer and replies with information previously provided as general metadata from the supplier.



## 4. Hosting and URL

The retailer Pubeasy soap service is hosted on the pubeasy system and is described using a WSDL.

For development and testing a sandbox system is available which is completely separated from the productive system.

sandbox domain: <https://sandbox-soap.pubeasy.com>

productive domain: <https://soap.pubeasy.com>

## 5. General differences to the legacy retailer API

The new retailer soap API is exchanging data on base of ONIX therefore values for product availability, product status and other values use mostly ONIX codelist values.

In difference to the former soap API internal Pubeasy IDs follow the rule of using [GUID](#) as unique identifier. The GUID field length is actually defined as max 50 characters.

For partner identification mailboxnames and account names are used.

See chapter 13 Appendix for further mappings.

## 6. General workflow

Generally the retailer only has to be connected to the Pubeasy system using the new API.

### 6.1. API versioning

The new SOAP service uses a versioning number for the API services which allows to develop and offer a new API version while the previous version can still be used by the customer. Customers can then decide to make an own migration plan to switch from the previous version to the following API version.

The current version is the initial version 1 which is visible by the version number in the namespace: for example: PersonalProductServiceV1

## 6.2. API updates

API updates are planned as extending the API with additional functionality which is not mandatory.

The version of the XSD files can be found at the end of the schema block

```
<xsd:schema xmlns:xsd='http://www.w3.org/2001/XMLSchema' targetNamespace='http://soap.mvb-online.com/OrderServiceMessagesV1' xmlns:message='http://soap.mvb-online.com/OrderServiceMessagesV1' xmlns:exception='http://soap.mvb-online.com/data/common/Exception/V1' xmlns:order='http://soap.mvb-online.com/data/order/OrderTypes/V1' xmlns:base='http://soap.mvb-online.com/data/common/Base/V1' elementFormDefault='qualified' attributeFormDefault='unqualified' version='1.2'>
```

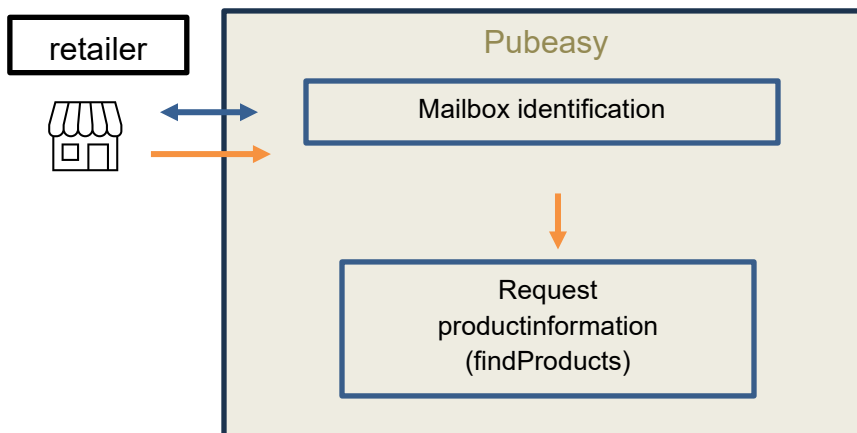
## 7. General workflow

Generally the retailer only has to be connected to the Pubeasy system using the new API.

### 7.1. Retailer supplier relationship

As with the old API there is normally at first a general connection necessary between the ordering and responding party– the account linking process.

For retrieving only general product data no account linking process is necessary. Which means that also no individual retailer price can be retrieved.



## 8. Technical definitions

The detailed technical API specification is provided by the WSDL and the HTML-specification pages.

A link to the WSDL zip file and the corresponding HTML technical API specification containing things like field names and other technical definitions.

This page contains the updated [Information & Documentation for SOAP API](#)

The latest retailer WSDL soap service descriptions are always available using the corresponding online WSDL url (see chapter 8.3 wsdl and endpoint for soap services).

### 8.1. Authentication methods

There are several authentication methods possible:

- basic authentication ([link](#))
- WS-security ([link](#)) with PasswordPlainText
- Client Certificate Authentication ([link](#))

The above authentication methods are automatically detected by Pubeasy.

PasswordDigest authentication can't be supported, as Pubeasy does not store the plain password.

Additional explanation for the authentication methods:

PasswordPlain send the authentication in the soap header

PasswordPlainText

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <wss:Security xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wss:UsernameToken xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
        <wss:Username xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">API-USERNAME</wss:Username>
        <wss:Password xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
          Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">API-PASSWORD</wss:Password>
      </wss:UsernameToken>
    </wss:Security>
  </S:Header>
```

BasicAuth is sending the authentication in the HTTP header



```

2025-01-30 09:54:10.517 [background-preinit] INFO o.h.validator.internal.util.Version - HV000001: Hibernate Validator 6.2.5.Final
2025-01-30 09:54:10.581 [main] INFO d.m.p.i.helper.TestHelperApplication - Starting TestHelperApplication v3.8.10 using Java 17.0.14 on W003853-DEU with PID 1388 (C:\Temp\soapse
2025-01-30 09:54:10.581 [main] INFO d.m.p.i.helper.TestHelperApplication - No active profile set, falling back to 1 default profile: "default"
2025-01-30 09:54:11.961 [main] INFO d.m.p.i.helper.TestHelperApplication - Started TestHelperApplication in 2.009 seconds (JVM running for 2.939)
2025-01-30 09:54:13.142 [main] INFO d.m.c.c.s.AbstractWebserviceClient - Using basic authentication with user 'API-USERNAME' for webservice 'PersonalProductSupplierServiceV1'.
2025-01-30 09:54:13.366 [main] WARN c.s.x.ws.transport.http.HttpAdapter - Using deprecated 'com.sun.xml.ws.transport.http.HttpAdapter.dumpThreshold' property, use 'com.sun.xml.ws
2025-01-30 09:54:13.367 [main] DEBUG c.s.x.w.t.h.client.HttpTransportPipe - ---[HTTP request - https://SUPPLIER-SOAP-SERVICE-URL]---
Accept: text/xml, multipart/related
Authorization: Basic c-KEY-GENERATED-BY-PUBEASY-Q==
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://localhost/getProductInformation"
User-Agent: JAX-WS RI 2.3.7 git-revision#654ef92
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

```

## 8.2. Communication method

The API uses synchronous calls which means, that each call is returning an immediate response (less then 60 seconds). The response is validated against the XSD of the API.

## 8.3. wsdl and endpoint for soap services

The WSDL for the retailer soap services and an API field definition and this API specification are available as zip files – see 13**Error! Reference source not found.**  
Appendix

For the latest WSDL please use always the URL and not the offline wsdl file.

The overall architecture of the new service is divided in two segments:

- the product metadata info service (PersonalProductService)
  - WSDL: /services/retailer/product/V1?wsdl
  - Endpoint: /services/retailer/product/V1

The sandbox order service WSDL for example is accessible by using the sandbox domain and the sandbox WSDL folder: <https://sandbox-pubeasy.pubnet.org/services/retailer/product/V1?wsdl>

For accessing the productive soap services add the productive domain as prefix.

The new soap API will increase the overall performance and reduce response times and network traffic. Another great improvement is its redesigned and strongly typed data model which enforces strict data consistency.

### 8.3.1. Namespace definitions

From now on namespaces have a fixed abbreviation:

| Namespace abbreviation | URL                                                                                                              |                                                             |
|------------------------|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| xmlns:ox               | http://ns.editeur.org/onix/3.0/reference                                                                         |                                                             |
| xmlns:si               | http://soap.mvb-online.com/data/common/ServiceInformation/V1                                                     |                                                             |
| xmlns:m                | http://soap.mvb-online.com/OrderServiceMessagesV1<br>http://soap.mvb-online.com/PersonalProductServiceMessagesV1 | Message Namespace m which can be different for each service |
| xmlns:ex               | http://soap.mvb-online.com/data/common/Exception/V1                                                              |                                                             |
| xmlns:a                | http://soap.mvb-online.com/data/common/AddressTypes/V1                                                           |                                                             |
| xmlns:o                | http://soap.mvb-online.com/data/order/OrderTypes/V1                                                              |                                                             |
| xmlns:id               | http://soap.mvb-online.com/data/common/IdentificationTypes/V1                                                    |                                                             |
| xmlns:pi               | http://soap.mvb-online.com/data/productinfo/ProductInformationTypes/V1                                           |                                                             |
| xmlns:xsi              | http://www.w3.org/2001/XMLSchema-instance                                                                        |                                                             |

### 8.3.2. XML schema definition (XSD)

XML Schema files in pubeasy

| XSD                                | explanation                                                      | Some example of type definitions contined in the XSD    |
|------------------------------------|------------------------------------------------------------------|---------------------------------------------------------|
| AddressTypes.xsd                   | Defining address objects                                         |                                                         |
| BaseTypes.xsd                      | Defining basic objects which are referenced in other definitions |                                                         |
| ExceptionTypes.xsd                 | Defining error and other exception objects                       | General exception definitions                           |
| IdentificationTypes.xsd            | Defining objects for general identification types                |                                                         |
| OnixTypesExtracted.xsd             | Defining onix objects and onix lists                             | ProductAvailabilityType, PublishingStatusType           |
| OrderServiceMessagesCommonV1.xsd   | Defining messages used for the order search and order handling   | Order exceptions                                        |
| OrderServiceMessagesRetailerV1.xsd | Defining messages for retailer order handling                    |                                                         |
| OrderServiceV1.wsdl                | Defining the retailer order service objects                      |                                                         |
| OrderTypes.xsd                     | Defining general order types                                     | LineProductStatusCodeType, LineProcessingStatusCodeType |

| XSD                         | explanation                    | Some example of type definitions contined in the XSD |
|-----------------------------|--------------------------------|------------------------------------------------------|
| ServiceInformationTypes.xsd | Defining general service types |                                                      |

### 8.3.3. field definition

The HTML api documentation shows the properties of each field and if it is mandatory or optional.

element **ProductFormFeatureType/ProductFormFeatureValue**

|            |                                                                                                                       |
|------------|-----------------------------------------------------------------------------------------------------------------------|
| diagram    |                                      |
| namespace  | <a href="http://ns.editeur.org/onix/3.0/reference">http://ns.editeur.org/onix/3.0/reference</a>                       |
| type       | <a href="#">tns:ProductFormFeatureValueType</a>                                                                       |
| properties | minOcc 0<br>maxOcc 1<br>content complex                                                                               |
| facets     | Kind Value Annotation<br>maxLength 20<br>pattern \s*\S.*                                                              |
| source     | <code>&lt;xsd:element name="ProductFormFeatureValue" type="tns:ProductFormFeatureValueType" minOccurs="0"/&gt;</code> |

For example ProductFormFeatureValue:

properties minOcc 0: field does not have to occur at any time – occurrence 0 = optional

If the minOcc is not defined the field is a mandaty field.

maxOcc 1: field can only be used once at this location.

Fields referencing ONIX code lists show the corresponding code list in the field definition.

Example ProductFormType

## complexType **ProductFormType**

|            |                                                                                                 |                                                       |
|------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| diagram    |                |                                                       |
| namespace  | <a href="http://ns.editeur.org/onix/3.0/reference">http://ns.editeur.org/onix/3.0/reference</a> |                                                       |
| type       | extension of <a href="#">tns:List150</a>                                                        |                                                       |
| properties | base <a href="#">tns:List150</a>                                                                |                                                       |
| used by    | element <a href="#">ProductPartType/ProductForm</a>                                             |                                                       |
| facets     | Kind                                                                                            | Value Annotation                                      |
|            | enumeration 00                                                                                  | documentation<br>Undefined                            |
|            | enumeration AA                                                                                  | documentation<br>Audio                                |
|            |                                                                                                 | documentation<br>Audio recording – detail unspecified |
|            | enumeration AB                                                                                  | documentation<br>Audio cassette                       |
|            |                                                                                                 | documentation<br>Audio cassette (analogue)            |
|            | enumeration AC                                                                                  | documentation                                         |

The productFormType references the List 150 of the Editeur ONIX 3.0 code list.

Fields referencing other code lists (not ONIX codelist) show the corresponding code list in the field definition.

Example: LineProcessingStatusCodeType

## simpleType **LineProcessingStatusCodeType**

|            |                                                                                                                       |                       |                                      |
|------------|-----------------------------------------------------------------------------------------------------------------------|-----------------------|--------------------------------------|
| namespace  | <a href="http://soap.mvb-online.com/data/order/OrderTypes/V1">http://soap.mvb-online.com/data/order/OrderTypes/V1</a> |                       |                                      |
| type       | restriction of <a href="#">tns:TokenType</a>                                                                          |                       |                                      |
| properties | base <a href="#">base:TokenType</a>                                                                                   |                       |                                      |
| used by    | element <a href="#">LineProcessingStatusCode</a>                                                                      |                       |                                      |
| facets     | Kind                                                                                                                  | Value                 | Annotation                           |
|            | minLength                                                                                                             | 1                     |                                      |
|            | pattern                                                                                                               | [a-zA-Z0-9.;#@\\/_-]+ |                                      |
|            | enumeration RETRIEVED                                                                                                 |                       | documentation<br>Order retrieved     |
|            | enumeration AWAIT                                                                                                     |                       | documentation<br>Awaiting Processing |
|            | enumeration PROGRESS                                                                                                  |                       | documentation<br>In progress         |

## 8.3.4. *availability and status definitions*

The new retailer service differentiates between a general product availability and a product status as part of an order.

Availability types not referencing order status values are ONIX based as ONIX can't describe order status types but only product status types.

For example:

ProductAvailabilityType: ONIX based

PublishingStatusType: ONIX based

LineProductStatusCodeType: pubeasy based

See also 8.3.2 XML schema definition (XSD)

## 9. Retailer soap services Overview

The retailer soap service is designed to raise orders, exchange order status updates and provide product and personal price information.

The overall architecture of the new service is divided in two segments:

- the product metadata info service (PersonalProductService)

Each segment uses an own WSDL to describe the underlying services.

### 9.1. partner identification

Supplier and retailer can be identified in the soap request either by their mailbox name or by the account which was defined in an account linking process.

If there was no account linking process was done with a supplier, it is not possible to get individual price and product information.

In case of there is no account between a specific supplier and retailer is existing, the retailer mailbox name and for requesting data from a specific supplier the supplier mailbox name has to be used.

For example:

Supplier identification:

```
<v1:Supplier type="MailboxName">MailboxSupplier1</v1:Supplier>
```

retailer identification:

**either:**

```
<v1:Retailer type="MailboxName"> MailboxRetailer1</v1:Retailer>
```

The partner identification and the authentication are done at different levels of the soap request. See also the section 9.3 Error and exception handling.

A partner identification is not necessary to send a getServiceInfo function call.

The following table shows when an additional retailer or supplier identification is necessary additionally to the user login when doing a function call.

This table lists the special cases for the authentication when an API user has the right to access different retailer mailboxes and the account can't be used.

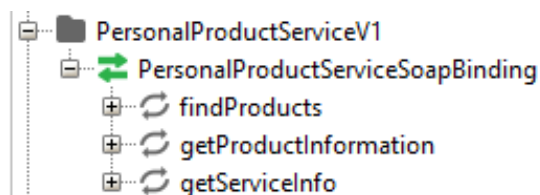
| functioncall          | Retailer identification mailbox | Retailer identification account    |
|-----------------------|---------------------------------|------------------------------------|
| findProducts          | ok                              | Only for single mailbox user       |
| getProductInformation | ok                              | For single and multi-mailbox users |

## 9.2. PersonalProductServiceV1

The personalProductService bundles the product information and personal price and availability information.

The product data is using ONIX based information as the internal structure stores article information ONIX based as bookindustry standard.

### 9.2.1. Function overview



## 9.2.1. *findProducts*

Find products allows to search for products on base of the ISBN or EAN of the product.

The response can be one product or, if the product is available at several suppliers, a list of products in which different supplier names are listed.

If an article is not available in the supplier system but can be found in the Pubeasy system the API will raise an error (see chapter 9.3 Error and exception handling).

### 9.2.1.1. *General Product information request - findProducts*

The retailers request is forwarded through Pubeasy and is the base of the PersonalProductServiceV1 WSDL.

The main features are:

1. retailer identification
2. supplier identification (optional in request)
3. product information

### 9.2.1.2. *findProductsResponse*

The response can contain:

1. general price information and
2. availability information
3. availability date

The availability for order (and with this the ProductCanBeOrdered flag in the new retailer API) will be set as a combination of:

1. ProductAvailability code (see table)
2. Sales rights
3. Order setting for a user in the pubeasy administration (if the user is allowed to order generally)

The findProductsResponse can contain productinformation for one ISBN / EAN from different suppliers. If only information of one specific supplier is needed, the request has to contain the supplier mailboxname.



### 9.2.2. *getServiceInfo*

The main features are:

Check login and availability of the pubeasy service and get pubeasy system version information.

This functionality is similar to the `getServiceInfo` at `OrderService`.



### 9.3. Error and exception handling

The pubeasy soap service has different error handling for authentication errors and other errors resulting from a wrong or malformed soap request.

As the user verification is done before interpreting the soap request, the error handling is different in the authentication case compared to other error cases.

Shortdescription error handling: For records previously send to Pubeasy it is expected that they are available and if not existend generate an error. Records Pubeasy does not expect to be available generate an empty response if nothing found.

See the ExceptionTypes.xsd, OrderServiceMessagesCommonV1.xsd for further information on error handling.

#### 9.3.1. *Not a SOAP response*

If the authentication is done through header autentication, an HTTP error code 401 or 403 should be generated. At this point the SOAP request can't be interpreted at all.

The following HTTP errors can be used in the HTTP response:

[http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#HTTP\\_Client\\_Error\\_Status\\_Codes](http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#HTTP_Client_Error_Status_Codes)

#### 9.3.2. *Universal soap fault*

Malformed XML errors (mostly HTTP errors, as the soap processing can't read the soap message). This kind of error is not defined in the WSDL



### 9.3.3. Error types

The following errors are defined in the WSDL.

Some exception are defined function specific others can be used in several of the functions. As each function has an authentication process and each function checks on malformed XML the error and exceptions resulting on these checks can happen for every of the functions.

The following list shows the faults defined in the WSDL.

The fault is always valid for the complete order and not only one orderline of the order.

- PartnerUnknownException
  - If the RetailerAccountNumberForSupplier sent by the retailer is unknown in the Pubeasy system.  
MessageCode UNKNOWN\_PARTNER
  - If the suppliermailbox name is unknown UNKNOWN\_PARTNER is shown
  - If the customeraccount is not allowed for trading within Pubeasy but for example only pubnet then  
MessageCode CUSTOMERACCOUNT\_NOTPUBEASY would be displayed
  - (see MessageCode definition 9.3.3.2 Current MessageCode definition)
- AuthorizationException
  - Any of the allowed authentication methods returns an error if the credentials or token are not valid.  
For this exception there is no info to the customer, as the customer normally has no possibility to change these credentials
- UnexpectedException
  - Other errors that can't be found in the above exception list

For the latest exceptions see the wsdl and xsd files.

## 9.3.3.1. *exception message fields*

The following table lists the fields an exception can have.

Always check against the latest WSDL.

| element       | values                                          | format                                                     | explanation                                                                                                                                                     |
|---------------|-------------------------------------------------|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *fault        | BusinessFaultType<br>predefined in<br>WSDL      | Predefined in<br>WSDL                                      | fault type is set at this level                                                                                                                                 |
| MessageLevel  | Predefined in<br>WSDL                           | Predefined in<br>WSDL                                      | General message level setting the type of message                                                                                                               |
| MessageCode   | Pubeasy defined<br>list which can be<br>adapted | Uppercase<br>letters, see<br>WSDL for<br>format definition | List of messagecodes that can be defined in discussion with<br>the supplier                                                                                     |
| Message       | Free text                                       | Free text, see<br>WSDL for<br>format definition            | This message can be used for example if a login on the<br>supplier side has changed and the Pubeasy application still<br>uses the old password / authentication |
| MessageDetail |                                                 |                                                            | Currently not processed during exception handling                                                                                                               |

## 9.3.3.2. *Current MessageCode definition*

The following list of MessageCodes is not defined in the WSDL but only defined within this specification.

The message code is always valid for the complete order and not only one orderline of the order.



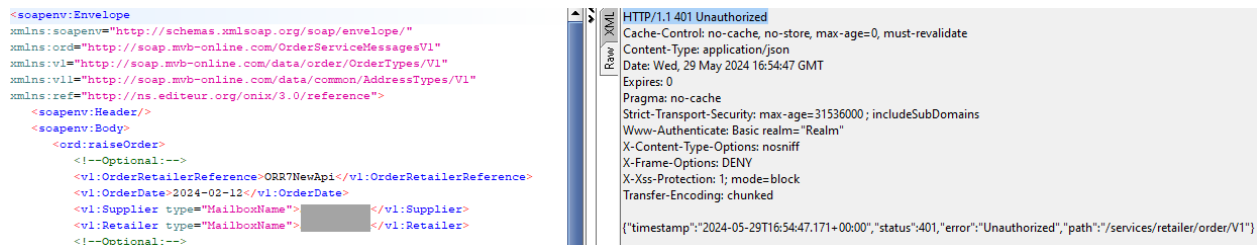
| MessageCode            | fault                | explanation                                                                                                                   |
|------------------------|----------------------|-------------------------------------------------------------------------------------------------------------------------------|
| UNKNOWN_PARTNER        | PartnerUnknownFault  | In any situation when the retailer or supplier identification given in the request does not match the Pubeays identification. |
| Different messagecodes | OrderNotFoundFault   | On requesting the order status of a single order                                                                              |
| DISPLAY_NOT_ALLOWED    | ProductNotFoundFault | On requesting productinformation of a single ISBN / article, where the display of the product is not allowed                  |
| UNKNOWN_PRODUCT        | ProductNotFoundFault | On requesting productinformation of a single ISBN / article and the product ist not in the Pubeasy database                   |
| Different messagecodes | ProductNotFoundFault | On requesting productinformation of a single ISBN / article                                                                   |



### 9.3.4. Authentication errors

Wrong user credentials for the soap request generate a HTTP status 401 response.

The following screenshot is done using soapUI as web service testing application.



### 9.3.5. http status type 400 error

Content errors are errors which are sent when the soap request itself is right formulated but the content could not be validated.

#### 9.3.1. Handling of http status codes

Responses with HTTP status code 500 can be analysed by the Pubeasy system, whereas HTTP 400 status codes are not further processed.

##### 9.3.1.1. Wrong mailbox given in soap request

For example a wrong mailbox name was given in the request



Raw XML

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <soap:Fault>
4       <faultcode>soap:Server</faultcode>
5       <faultstring>Partner 'SP4TRAPUSH-xxx' with type 'MAILBOX_NAME' is unknown.</faultstring>
6       <detail>
7         <ex:PartnerUnknownFault xmlns:si="http://soap.mvb-online.com/data/common/ServiceInformation/V1"
8           xmlns:ox="http://ns.editeur.org/onix/3.0/reference"
9           xmlns:oc="http://soap.mvb-online.com/data/order/OrderTypes/V1"
10          xmlns:om="http://soap.mvb-online.com/OrderServiceMessages/V1"
11          xmlns:id="http://soap.mvb-online.com/data/common/IdentificationTypes/V1"
12          xmlns:ex="http://soap.mvb-online.com/data/common/Exception/V1"
13          xmlns:a="http://soap.mvb-online.com/data/common/AddressTypes/V1">
14           <ex:Message>
15             <ex:MessageLevel>ERROR</ex:MessageLevel>
16             <ex:MessageCode>UNKNOWN_PARTNER</ex:MessageCode>
17             <ex:Message>Partner 'SP4TRAPUSH-xxx' with type 'MAILBOX_NAME' is unknown.</ex:Message>
18           </ex:Message>
19         </ex:PartnerUnknownFault>
20       </detail>
21     </soap:Fault>
22   </soap:Body>
23 </soap:Envelope>
```

| Header                    | Value                                                                |
|---------------------------|----------------------------------------------------------------------|
| X-Content-Type-Options    | nosniff                                                              |
| Pragma                    | no-cache                                                             |
| Date                      | Wed, 29 May 2024 17:00:43 GMT                                        |
| X-Frame-Options           | DENY                                                                 |
| Strict-Transport-Security | max-age=31536000; includeSubDomains                                  |
| Cache-Control             | no-cache, no-store, max-age=0, must-revalidate                       |
| X-Xss-Protection          | 1; mode=block                                                        |
| Set-Cookie                | JSESSIONID=6444DCEA38CB33A4D94D915458EFAE3; Path=/; Secure; HttpOnly |
| Expires                   | 0                                                                    |
| #status#                  | HTTP/1.1 500 Internal Server Error                                   |
| Content-Length            | 968                                                                  |
| Content-Type              | text/xml; charset=UTF-8                                              |

## 9.3.2. http status 500 error

### 9.3.2.1. malformed XML request

Malformed soap XML requests, in the following example it was a malformed closing tag: `<v1:Quantity>15</v1:Quantity-1>`, result in an http 500 error.

The screenshot shows a web browser window displaying a SOAP XML error. The XML content is as follows:

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <soap:Fault>
4       <faultcode>soap:Client</faultcode>
5       <faultstring>Unmarshalling Error: Unexpected close tag <v1:Quantity-1>;
6         expected <v1:Quantity>.
7       at [row,col {unknown-source}]: [26,45]</faultstring>
8     </soap:Fault>
9   </soap:Body>
10 </soap:Envelope>
```

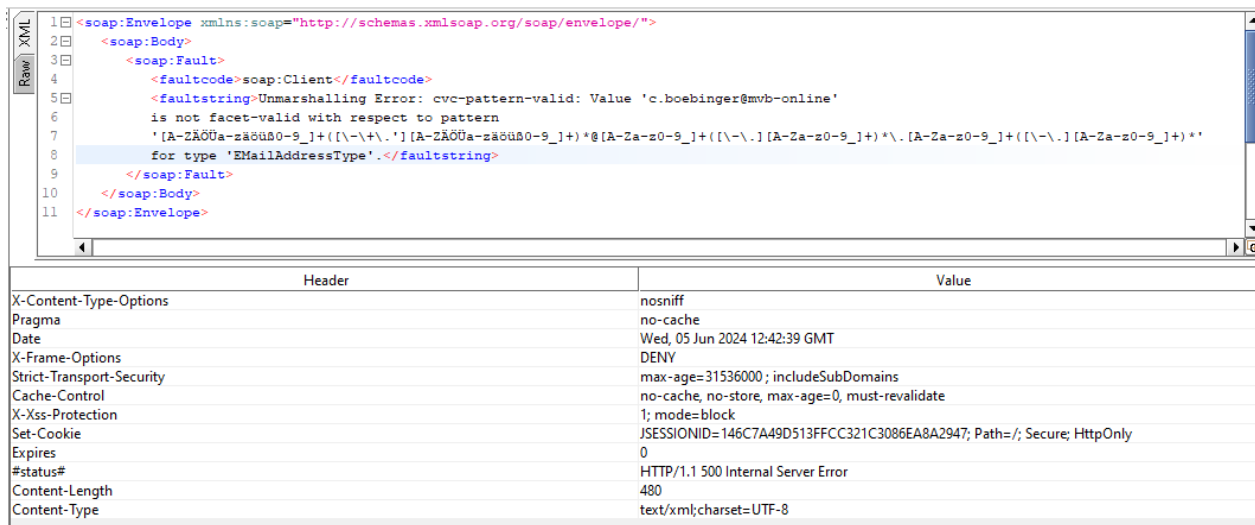
Below the XML content, the HTTP headers and values are listed in a table:

| Header                    | Value                                                      |
|---------------------------|------------------------------------------------------------|
| X-Content-Type-Options    | nosniff                                                    |
| Pragma                    | no-cache                                                   |
| Date                      | Wed, 05 Jun 2024 08:10:59 GMT                              |
| X-Frame-Options           | DENY                                                       |
| Strict-Transport-Security | max-age=31536000; includeSubDomains                        |
| Cache-Control             | no-cache, no-store, max-age=0, must-revalidate             |
| X-Xss-Protection          | 1; mode=block                                              |
| Set-Cookie                | JSESSIONID=8B04026B3FE91E297B61F1F65E32DF25; Path=/; Se... |
| Expires                   | 0                                                          |
| #status#                  | HTTP/1.1 500 Internal Server Error                         |
| Content-Length            | 326                                                        |
| Content-Type              | text/xml; charset=UTF-8                                    |

## 9.3.2.2. Wrong data like malformed mailaddress or wrong Boolean flags

For some fields a format check is available which results in an http 500 error if the request format does not match the requirements

```
<faultcode>soap:Client</faultcode>
<faultstring>Unmarshalling Error: cvc-pattern-valid: Value 'c.boebinger@mvb-online' is not facet-
valid with respect to pattern
'[A-ZÄÖÜa-zäöüß0-9_]+([\-\+\.\']+[A-ZÄÖÜa-zäöüß0-9_])*@[A-Za-z0-9_]+([\-\.\']+[A-Za-z0-9_]+)*\.[A-
Za-z0-9_]+([\-\.\']+[A-Za-z0-9_]+)*'
for type 'EmailAddressType'.</faultstring>
```



The screenshot shows a web browser window with an XML error message displayed in the main content area. The error message is a SOAP fault indicating a pattern validation failure for an email address. Below the XML, a table lists the HTTP headers and their values.

| Header                    | Value                                                                 |
|---------------------------|-----------------------------------------------------------------------|
| X-Content-Type-Options    | nosniff                                                               |
| Pragma                    | no-cache                                                              |
| Date                      | Wed, 05 Jun 2024 12:42:39 GMT                                         |
| X-Frame-Options           | DENY                                                                  |
| Strict-Transport-Security | max-age=31536000; includeSubDomains                                   |
| Cache-Control             | no-cache, no-store, max-age=0, must-revalidate                        |
| X-Xss-Protection          | 1; mode=block                                                         |
| Set-Cookie                | JSESSIONID=146C7A49D513FFCC321C3086EA8A2947; Path=/; Secure; HttpOnly |
| Expires                   | 0                                                                     |
| #status#                  | HTTP/1.1 500 Internal Server Error                                    |
| Content-Length            | 480                                                                   |
| Content-Type              | text/xml; charset=UTF-8                                               |

```
<faultcode>soap:Client</faultcode>
<faultstring>Unmarshalling Error: cvc-datatype-valid.1.2.1:
'YES' is not a valid value for 'boolean'.</faultstring>
```

XML

Raw

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <soap:Fault>
4       <faultcode>soap:Client</faultcode>
5       <faultstring>Unmarshalling Error: cvc-datatype-valid.1.2.1:
6         'YES' is not a valid value for 'boolean'.</faultstring>
7     </soap:Fault>
8   </soap:Body>
9 </soap:Envelope>

```

| Header                    | Value                                                                 |
|---------------------------|-----------------------------------------------------------------------|
| X-Content-Type-Options    | nosniff                                                               |
| Pragma                    | no-cache                                                              |
| Date                      | Wed, 05 Jun 2024 12:49:02 GMT                                         |
| X-Frame-Options           | DENY                                                                  |
| Strict-Transport-Security | max-age=31536000; includeSubDomains                                   |
| Cache-Control             | no-cache, no-store, max-age=0, must-revalidate                        |
| X-Xss-Protection          | 1; mode=block                                                         |
| Set-Cookie                | JSESSIONID=1392121C71EC297D5AEB1E16C1AED4FD; Path=/; Secure; HttpOnly |
| Expires                   | 0                                                                     |
| #status#                  | HTTP/1.1 500 Internal Server Error                                    |
| Content-Length            | 284                                                                   |
| Content-Type              | text/xml; charset=UTF-8                                               |

### 9.3.2.3. Malformed XML request – wrong sequence of elements

<faultcode>soap:Client</faultcode>

<faultstring>Unmarshalling Error: cvc-complex-type.2.4.a:

Invalid content was found starting with element 'v1:PromotionalCode'.

One of '{ "http://soap.mvb-online.com/data/order/OrderTypes/V1":Quantity}' is expected.</faultstring>

XML

Raw

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <soap:Fault>
4       <faultcode>soap:Client</faultcode>
5       <faultstring>Unmarshalling Error: cvc-complex-type.2.4.a:
6         Invalid content was found starting with element 'v1:PromotionalCode'.
7         One of '{ "http://soap.mvb-online.com/data/order/OrderTypes/V1":Quantity}' is expected.</faultstring>
8     </soap:Fault>
9   </soap:Body>
10 </soap:Envelope>

```

| Header                    | Value                                                                 |
|---------------------------|-----------------------------------------------------------------------|
| X-Content-Type-Options    | nosniff                                                               |
| Pragma                    | no-cache                                                              |
| Date                      | Wed, 05 Jun 2024 08:16:51 GMT                                         |
| X-Frame-Options           | DENY                                                                  |
| Strict-Transport-Security | max-age=31536000; includeSubDomains                                   |
| Cache-Control             | no-cache, no-store, max-age=0, must-revalidate                        |
| X-Xss-Protection          | 1; mode=block                                                         |
| Set-Cookie                | JSESSIONID=922772014650F70B7514EAECBFD8175A; Path=/; Secure; HttpOnly |
| Expires                   | 0                                                                     |
| #status#                  | HTTP/1.1 500 Internal Server Error                                    |
| Content-Length            | 397                                                                   |
| Content-Type              | text/xml; charset=UTF-8                                               |



This error was generated by setting the correctly formed  
<v1:PromotionalCode>PromCodNewAPI</v1:PromotionalCode>  
at the wrong position in the soap request.

The appropriate XSD shows in which sequence the elements can be used in the soap request:

For example PromotionalCode

```
<xsd:complexType name='OrderLineRaiseRetailerBaseType'>
  <xsd:sequence>
    <xsd:element ref='tns:LineRetailerReference' minOccurs='0'/>
    <xsd:element ref='onix:ProductIdentifier'/>
    <xsd:element ref='tns:Quantity'/>
    <xsd:element ref='onix:TitleDetail' minOccurs='0' maxOccurs='unbounded'/>
    <xsd:element ref='onix:Contributor' minOccurs='0' maxOccurs='unbounded'/>
    <xsd:element ref='onix:Price' minOccurs='0' maxOccurs='unbounded'/>
    <xsd:element ref='onix:Discount' minOccurs='0'/>
    <xsd:element ref='tns:PromotionalCode' minOccurs='0'/>
  </xsd:sequence>
</xsd:complexType>
```

#### 9.3.2.4. text values not allowed as values in text fields

In this exception example a not allowed values was used for the OrderRetailerReference field.

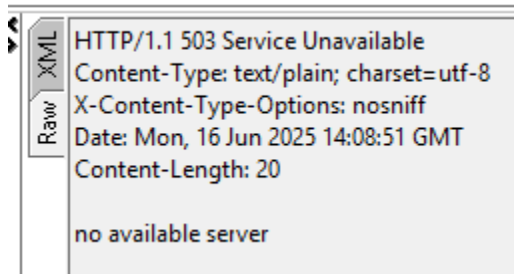
The asterisks "\*" is not allowed as Reference field value.

The screenshot shows a SOAP API client interface. On the left, the XML request is displayed, containing a `<v1:OrderRetailerReference>ORR-NewApi*</v1:OrderRetailerReference>` element. On the right, the SOAP fault response is shown, indicating an 'Unmarshalling Error: cvc-pattern-valid: Value 'ORR-NewApi\*' is not facet-valid with respect to pattern '[a-zA-Z0-9.-:~#\\/\_-]+' for type 'OrderRetailerReferenceType'.

| Header                    | Value                                                               |
|---------------------------|---------------------------------------------------------------------|
| X-Content-Type-Options    | nosniff                                                             |
| Pragma                    | no-cache                                                            |
| Date                      | Wed, 04 Jun 2025 12:11:01 GMT                                       |
| X-Frame-Options           | DENY                                                                |
| Strict-Transport-Security | max-age=31536000; includeSubDomains                                 |
| Cache-Control             | no-cache, no-store, max-age=0, must-revalidate                      |
| X-Xss-Protection          | 1; mode=block                                                       |
| Set-Cookie                | JSESSIONID=148A9264C34FAD7D123A837CE2AA4503; Path=/; Secure; Htt... |
| Expires                   | 0                                                                   |
| #status#                  | HTTP/1.1 500 Internal Server Error                                  |
| Content-Length            | 361                                                                 |
| Content-Type              | text/xml; charset=UTF-8                                             |

### 9.3.3. *http status 503 - Pubeasy soap server is not available*

If the Pubeasy soap service is not available an http status 503 will be shown additional to the text “no available server”



## 10. Soap examples

### 10.1. PersonalProductService

In case no account linking is existing between retailer and supplier only the findProducts can be used to retrieve product data.

#### 10.1.1. *findProducts*

##### 10.1.1.1. *findProducts request towards pubeasy*

The soap request towards the pubeasy system for requesting product information is as follows. Providing the supplier mailbox is optional in the findProducts request.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:per="http://soap.mvb-online.com/PersonalProductServiceMessagesV1"
  xmlns:ref="http://ns.editeur.org/onix/3.0/reference">
  <soapenv:Header/>
  <soapenv:Body>
    <per:findProducts>
      <!--Optional:-->
      <per:Retailer type="MailboxName" displayName="RetailerDisplayName">RetailerMailbox</per:Retailer>
      <!--Optional:-->
      <per:Supplier type="MailboxName" displayName="SupplierDisplayName">SupplierMailbox</per:Supplier>
      <ref:ProductIdentifier>
        <ref:ProductIDType>15</ref:ProductIDType>
        <ref:IDValue> 9783765731426</ref:IDValue>
      </ref:ProductIdentifier>
    </per:findProducts>
  </soapenv:Body>
</soapenv:Envelope>
```



### 10.1.1.2. *findProductsResponse* response from pubeasy

The following immediate response is expected from supplier as product information response towards pubeasy:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <m:findProductsResponse
      xmlns:ox="http://ns.editeur.org/onix/3.0/reference"
      xmlns:si="http://soap.mvb-online.com/data/common/ServiceInformation/V1"
      xmlns:m="http://soap.mvb-online.com/PersonalProductServiceMessagesV1"
      xmlns:ex="http://soap.mvb-online.com/data/common/Exception/V1"
      xmlns:pi="http://soap.mvb-online.com/data/productinfo/ProductInformationTypes/V1"
      xmlns:id="http://soap.mvb-online.com/data/common/IdentificationTypes/V1">
      <m:ProductInformation>
        <m:Supplier type="MailboxName" displayName="MOCK2-SUP-SOAP">MOCK2-SUP-SOAP</m:Supplier>
        <m:Product>
          <ox:ProductIdentifier>
            <ox:ProductIDType>15</ox:ProductIDType>
            <ox:IDValue>9783765731426</ox:IDValue>
          </ox:ProductIdentifier>
          <ox:TitleDetail>
            <ox:TitleType>01</ox:TitleType>
            <ox:TitleElement>
              <ox:TitleElementLevel>01</ox:TitleElementLevel>
              <ox:TitleText>Mock test main title 5</ox:TitleText>
              <ox:Subtitle>fifth test sub title</ox:Subtitle>
            </ox:TitleElement>
          </ox:TitleDetail>
          <ox:TitleDetail>
            <ox:TitleType>01</ox:TitleType>
            <ox:TitleElement>
              <ox:TitleElementLevel>02</ox:TitleElementLevel>
              <ox:TitleText>Retailer test title 2 collection</ox:TitleText>
            </ox:TitleElement>
          </ox:TitleDetail>
          <ox:Contributor>
            <ox:ContributorRole>A01</ox:ContributorRole>
            <ox:PersonName>Lastname, Firstname</ox:PersonName>
          </ox:Contributor>
          <ox:Publisher>
            <ox:PublishingRole>01</ox:PublishingRole>
            <ox:PublisherName>Mock2 service Imprint</ox:PublisherName>
          </ox:Publisher>
          <ox:Publisher>
            <ox:PublishingRole>01</ox:PublishingRole>
            <ox:PublisherName>MVB test publisher2</ox:PublisherName>
          </ox:Publisher>
          <ox:PublishingDate>
            <ox:PublishingDateRole>01</ox:PublishingDateRole>
            <ox:Date>2012-10-03</ox:Date>
          </ox:PublishingDate>
          <ox:Price>
            <ox:PriceType>01</ox:PriceType>
            <ox:PriceAmount>15</ox:PriceAmount>
            <ox:CurrencyCode>USD</ox:CurrencyCode>
          </ox:Price>
          <pi:ProductAvailabilityInfo>
            <pi:ProductCanBeOrdered>false</pi:ProductCanBeOrdered>
            <pi:ProductAvailability>33</pi:ProductAvailability>
            <pi:AvailabilityDate>2025-01-08</pi:AvailabilityDate>
          </pi:ProductAvailabilityInfo>
          <ox:ProductPart>
            <ox:ProductForm>BC</ox:ProductForm>
          </ox:ProductPart>
        </m:Product>
      </m:ProductInformation>
    </m:findProductsResponse>
  </soap:Body>
</soap:Envelope>
```



```
</m:Product>
</m:ProductInformation>
<m:ProductInformation>
  <m:Supplier type="MailboxName" displayName="MOCK-SUP-NEWSOAP">MOCK-SUP-NEWSOAP</m:Supplier>
  <m:Product>
    <ox:ProductIdentifier>
      <ox:ProductIDType>15</ox:ProductIDType>
      <ox:IDValue>9783765731426</ox:IDValue>
    </ox:ProductIdentifier>
    <ox:TitleDetail>
      <ox:TitleType>01</ox:TitleType>
      <ox:TitleElement>
        <ox:TitleElementLevel>01</ox:TitleElementLevel>
        <ox:TitleText>Mock test main title 5</ox:TitleText>
        <ox:Subtitle>fifth test sub title</ox:Subtitle>
      </ox:TitleElement>
    </ox:TitleDetail>
    <ox:TitleDetail>
      <ox:TitleType>01</ox:TitleType>
      <ox:TitleElement>
        <ox:TitleElementLevel>02</ox:TitleElementLevel>
        <ox:TitleText>Retailer test title 2 collection</ox:TitleText>
      </ox:TitleElement>
    </ox:TitleDetail>
    <ox:Contributor>
      <ox:ContributorRole>A01</ox:ContributorRole>
      <ox:PersonName>Lastname, Firstname</ox:PersonName>
    </ox:Contributor>
    <ox:Publisher>
      <ox:PublishingRole>01</ox:PublishingRole>
      <ox:PublisherName>Mock service Imprint</ox:PublisherName>
    </ox:Publisher>
    <ox:Publisher>
      <ox:PublishingRole>01</ox:PublishingRole>
      <ox:PublisherName>MVB test publisher</ox:PublisherName>
    </ox:Publisher>
    <ox:PublishingDate>
      <ox:PublishingDateRole>01</ox:PublishingDateRole>
      <ox:Date>2012-10-03</ox:Date>
    </ox:PublishingDate>
    <ox:Price>
      <ox:PriceType>01</ox:PriceType>
      <ox:PriceAmount>15</ox:PriceAmount>
      <ox:CurrencyCode>USD</ox:CurrencyCode>
    </ox:Price>
    <pi:ProductAvailabilityInfo>
      <pi:ProductCanBeOrdered>false</pi:ProductCanBeOrdered>
      <pi:ProductAvailability>33</pi:ProductAvailability>
      <pi:AvailabilityDate>2025-01-08</pi:AvailabilityDate>
    </pi:ProductAvailabilityInfo>
    <ox:ProductPart>
      <ox:ProductForm>BC</ox:ProductForm>
    </ox:ProductPart>
  </m:Product>
</m:ProductInformation>
</m:findProductsResponse>
</soap:Body>
</soap:Envelope>
```



### 10.1.2. *getServiceInfo*

This request is sometimes sent at the beginning after a longer break to check if the responding service is available.

Request:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <si:getServiceInfo
      xmlns:ox="http://ns.editeur.org/onix/3.0/reference"
      xmlns:si="http://soap.mvb-online.com/data/common/ServiceInformation/V1"
      xmlns:m="http://soap.mvb-online.com/OrderServiceMessages/V1"
      xmlns:a="http://soap.mvb-online.com/data/common/AddressTypes/V1"
      xmlns:ex="http://soap.mvb-online.com/data/common/Exception/V1"
      xmlns:o="http://soap.mvb-online.com/data/order/OrderTypes/V1"
      xmlns:id="http://soap.mvb-online.com/data/common/IdentificationTypes/V1"/>
    </S:Body>
  </S:Envelope>
```

Pubeasy soap api getServiceInfo response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <si:getServiceInfoResponse
      xmlns:ox="http://ns.editeur.org/onix/3.0/reference"
      xmlns:si="http://soap.mvb-online.com/data/common/ServiceInformation/V1"
      xmlns:m="http://soap.mvb-online.com/OrderServiceMessages/V1"
      xmlns:ex="http://soap.mvb-online.com/data/common/Exception/V1"
      xmlns:a="http://soap.mvb-online.com/data/common/AddressTypes/V1"
      xmlns:o="http://soap.mvb-online.com/data/order/OrderTypes/V1"
      xmlns:id="http://soap.mvb-online.com/data/common/IdentificationTypes/V1">
      <si:Name>RetailerService</si:Name>
      <si:Version>3.5.0-SNAPSHOT</si:Version>
      <si:ExecutionEnvironment>
        <si:Environment>productive</si:Environment>
        <si:ServerName>pubx-ws-retailer-application-697f7f7b5-8jlqj</si:ServerName>
      </si:ExecutionEnvironment>
    </si:getServiceInfoResponse>
  </soap:Body>
</soap:Envelope>
```

## 11. Test possibilities



For test purpose the WSDL can be imported in SOAP test tools like soapUI. These tools generate based on the WSDL empty requests which can be used to generate example requests to

### 11.1. Service info

The service info helper response can be used as a first test to check the functionality and availability of the new soap service.

### 11.2. Tools for testing soap service

A good tool to do first tests with the soap service is the tool soapUI (<https://www.soapui.org/>) . SoapUI can read each soap service WSDL.

## 12. Procedure and tools to setup a test environment

The new SOAP retailer service is provided by Pubeasy. The WSDL and XSD files are the main description to build the service.

- The latest version of the WSDL is always available when calling the WSDL URL.
  - a. The new SOAP retailer service is currently split into two services, which can be switched independently from each other. The current description only covers the product service (PersonalProductServiceV1)
- The Pubeasy Technical Account Manager will provide a new test mailbox on the Sandbox system which will be setup to connect during the testing phase to the new Pubeasy SOAP service (test system).



## 13. Appendix

### 13.1. EDItEUR onix code lists

The latest onix 3.0 code lists can be found on:

<https://ns.editeur.org/onix/en>

When a code list is used it is mentioned with its code list number.

For example ProductIdentifier ProductIDType is using code list 5 ( <https://ns.editeur.org/onix/en/5> ).

### 13.2. Code lists, mappings and field definitions

For code lists there is a difference between article code list and code lists used in order correlation. Background of this difference is, that ONIX does not support order status values but only availability status values.

#### 13.2.1. *Field definitions*

Some field definitions



| field                | Onix 3 code list | values                                 | meaning                                           | format               | comment                                                             |
|----------------------|------------------|----------------------------------------|---------------------------------------------------|----------------------|---------------------------------------------------------------------|
| publishingStatusDate | -                |                                        |                                                   | Epoch Unix Timestamp |                                                                     |
| SupplierSpecific     | No code list     | Individual definition on consultation. | Data for one specific supplier can be transferred |                      | one implementation for supplierspecific data is currently available |

### 13.2.2. Code list Mapping overview

Codelists could change. Please also check the XSD definitions.

| field                 | function              | Onix 3 / other code list          | values | meaning            | comment                                                                                  |
|-----------------------|-----------------------|-----------------------------------|--------|--------------------|------------------------------------------------------------------------------------------|
| ProductIDType         |                       | ONIX code list 5                  |        |                    | See XSD definition which currently allows only a reduced set of value from the code list |
| publishingStatus      | getProductinformation | <a href="#">ONIX code list 64</a> |        |                    |                                                                                          |
| marketRestrictionCode |                       |                                   | SNP    | Sale not permitted | Currently only one value is set.                                                         |
| productAvailability   | getProductinformation | <a href="#">ONIX code list 65</a> |        |                    | See XSD definition 8.3.2 XML schema definition (XSD)                                     |
| CurrencyCode          |                       | ONIX code list 96                 |        |                    |                                                                                          |
| PriceType             |                       | ONIX code list 58                 |        |                    |                                                                                          |
| TitleElementLevel     |                       | ONIX code list 149                |        |                    |                                                                                          |
| TitleType             |                       | ONIX code list 15                 | 15.01  |                    | See XSD definition which currently allows only one value of he code list: List15.01      |
| ContributorRole       |                       | ONIX code list 17                 | 17.A01 |                    | See XSD definition which currently allows only one value of he code list                 |
| PublishingDateRole    |                       | ONIX code list 163                | 163.01 |                    | See XSD definition which currently allows only one value of he code list                 |
| ProductForm           |                       | ONIX code list 150                |        |                    |                                                                                          |
| ProductFormDetail     |                       | ONIX code list 175                |        |                    |                                                                                          |
| ProductFormFeature    |                       | ONIX code list 79                 |        |                    |                                                                                          |
| ProductContentType    |                       | ONIX code list 81                 |        |                    |                                                                                          |



| field            | function | Onix 3 / other code list | values | meaning | comment                                                                                  |
|------------------|----------|--------------------------|--------|---------|------------------------------------------------------------------------------------------|
| ProductPackaging |          | ONIX code list 80        |        |         |                                                                                          |
| MeasureType      |          | ONIX code list 48        |        |         |                                                                                          |
| MeasureUnitCode  |          | ONIX code list 50        |        |         |                                                                                          |
| PublishingRole   |          | ONIX code list 45        | 45.01  |         | See XSD definition which currently allows only a reduced set of value from the code list |

The XSD OnixTypesExtracted lists the currently used onix codelists and the codelist exceptions.

### 13.2.3. *productAvailability mapping between legacy and ONIX*

The product availability values are based on the EDItEUR ONIX [code list 65](#).

Codelists could change. Please also check the XSD definitions at 8.3.2 XML schema definition (XSD).

### 13.2.4. *Availability status codes and UI text*

The availability for order (and with this the ProductCanBeOrdered towards the retailer) will be set as a combination of:

- ProductAvailability code (see table)
- Sales rights
- Order setting for a user in the pubeasy administration (if the user is allowed to order generally)



| Availability status labels used in pubeasy UI                   | ProductAvailability<br>Onix code (list 65) | Availability for order in UI (same as ProductCanBeOrdered) | Article Colour code | For reference (legacy code send by supplier - Appendix D - Product Availability Status Codes) |
|-----------------------------------------------------------------|--------------------------------------------|------------------------------------------------------------|---------------------|-----------------------------------------------------------------------------------------------|
| Cancelled                                                       | 1                                          | N                                                          | RED                 | AB                                                                                            |
| Not yet available, postponed indefinitely                       | 9                                          | N                                                          | RED                 |                                                                                               |
| Not yet available                                               | 10                                         | Y                                                          | YELLOW              | NP                                                                                            |
| Awaiting stock                                                  | 11                                         | Y                                                          | YELLOW              | TU, RP                                                                                        |
| Not yet available, will be POD                                  | 12                                         | Y                                                          | YELLOW              | MD                                                                                            |
| Available                                                       | 20                                         | Y                                                          | GREEN               | IP, I2                                                                                        |
| In stock                                                        | 21                                         | Y                                                          | GREEN               |                                                                                               |
| To order                                                        | 22                                         | Y                                                          | YELLOW              | TO, OO, IT                                                                                    |
| POD                                                             | 23                                         | Y                                                          | GREEN               |                                                                                               |
| Temporarily unavailable                                         | 30                                         | Y                                                          | YELLOW              |                                                                                               |
| Out of stock                                                    | 31                                         | Y                                                          | RED                 | OS                                                                                            |
| Reprinting                                                      | 32                                         | N                                                          | RED                 |                                                                                               |
| Awaiting reissue                                                | 33                                         | Y                                                          | YELLOW              | RE                                                                                            |
| Temporarily withdrawn from sale                                 | 34                                         | N                                                          | RED                 |                                                                                               |
| Not available (reason unspecified)                              | 40                                         | N                                                          | RED                 | UC, RR, OP                                                                                    |
| Not available, replaced by new product                          | 41                                         | N                                                          | YELLOW              | SB, OR                                                                                        |
| Not available, other format available                           | 42                                         | N                                                          | YELLOW              | PK, OF, HK                                                                                    |
| No longer supplied by us                                        | 43                                         | N                                                          | RED                 |                                                                                               |
| Apply direct                                                    | 44                                         | N                                                          | YELLOW              | AD                                                                                            |
| Not sold separately                                             | 45                                         | Y                                                          | YELLOW              |                                                                                               |
| Withdrawn from sale                                             | 46                                         | N                                                          | RED                 |                                                                                               |
| Remaindered                                                     | 47                                         | Y                                                          | YELLOW              | RM                                                                                            |
| Not available, replaced by POD                                  | 48                                         | N                                                          | YELLOW              |                                                                                               |
| Recalled                                                        | 49                                         | N                                                          | RED                 |                                                                                               |
| Not sold as set                                                 | 50                                         | N                                                          | RED                 |                                                                                               |
| Not available, publisher indicates OP                           | 51                                         | Y                                                          | RED                 |                                                                                               |
| Not available, publisher no longer sells product in this market | 52                                         | N                                                          | RED                 | RF                                                                                            |
| No recent update received                                       | 97                                         | Y                                                          | WHITE               |                                                                                               |



| Availability status labels used in pubeasy UI | ProductAvailability<br>Onix code (list 65) | Availability for order in UI (same as ProductCanBeOrdered) | Article Colour code | For reference (legacy code send by supplier - Appendix D - Product Availability Status Codes) |
|-----------------------------------------------|--------------------------------------------|------------------------------------------------------------|---------------------|-----------------------------------------------------------------------------------------------|
| No longer receiving updates                   | 98                                         | Y                                                          | WHITE               |                                                                                               |
| Unknown - Contact supplier                    | 99                                         | N                                                          | WHITE               | CS, CB                                                                                        |