



ANB5Web

API Documentation

Revision 1.13
03-07-2020

Content

Retreiving prices.....	2
Pricing/Stock check.....	3
Order creation.....	5
Retrieving XML files.....	7

Retrieving price list

API URL: log in via the website and choose api for my account

Data CSVGET: ?username=<username>&token=<token> (GET)

Username: Username as it is in the webshop

Token: API token as provided by the web shop owner. The token is based on your webshop password.

This generates a price list in csv, with a semicolon as delimiter.

Price/Stock Check

API URL: log in via the website and choose api for my account

Data: XML, to be sent as raw data or as a file upload (POST)

It is possible to make a link with our price/stock system. With this you can always use retrieve the latest information.

Example request:

```
<pna>
<username>gebruiker</username>
<password>wachtwoordhash</password>
<vendorpartnr>NK2983</vendorpartnr>
<vendorpartnr>21345678564651684</vendorpartnr>
<prodcode>16800</prodcode>
</pna>
```

Example response:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<pna>
<product>
<prodcode>00019891</prodcode>
<quantity>11</quantity>
<eta>2010-10-02</eta>
<vendorpartnr>NK2983</vendorpartnr>
<price>100</price>
</product>
<product>
<vendorpartnr>21345678564651684</vendorpartnr>
<error>Item not found</error>
</product>
<product>
<prodcode>00016800</prodcode>
<quantity>2</quantity>
<eta>2010-10-02</eta>
<vendorpartnr>DIR-8424</vendorpartnr>
<price>50</price>
</product>
</pna>
```

When the input of the username and/or password is not accepted, you will get the following result back:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<pna>
<status>ERROR</status>
<code>10-2</code>
<error>Please verify your username/password.</error>
</pna>
```

Execute orders

API URL: log in via the website and choose api for my account

Data: XML, to be sent as raw content or as a file upload (POST)

One way to place orders is via XML. This will create a shopping cart in the webshop which will be filled with products and then put in order under the standard conditions that apply to this customer.

Example order:

```
<Order>
<Orderref>mijnref123</Orderref>
<username>Gebruiker</username>
<password>wachtwoordhash</password>
<verzendadres>
<geadresseerde>Bedrijfsnaam</geadresseerde>
<terattentiev>Contactpersoon</terattentiev>
<adres>Straatnaam 1234</adres>
<postcode>1111AA</postcode>
<plaats>Plaatsnaam</plaats>
<land>NL</land>
<telefoon>010-1234567</telefoon>
<email>jan@hotmail.com</email>
<verzendreferentie>order 13</verzendreferentie>
</verzendadres>
<Line>
<ProdCode>931</ProdCode>
<Aantal>10</Aantal>
</Line>
<Line>
<ProdCode>761</ProdCode>
<Aantal>1</Aantal>
</Line>
<Line>
<ProdCode>877</ProdCode>
<Aantal>1</Aantal>
</Line>
</Order>
```

This can be used to place an order. Shipping address is optional for drop shipments.

The response to this, as an example below, gives an overview of the order how it enters the system. PLEASE NOTE: Products that are not found, are ignored and are therefore NOT in the result. You need to check this yourself.

Example response:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<order>
<status>OK</status>

<ref>1234</ref> Referentie webshop
<product>
<sku>00000931</sku> Artikelnummer
<vpn>R-425071</vpn> Vendorpartnummer
<description> Internal Cardreader 3.5 inch - bulk- black</description> Artikelomschrijving
<quantity>10</quantity> Aantal
<price>38.15 </price> Prijs
</product>
<product>
<sku>00000761</sku>
<vpn>Accelero L2 Pro</vpn>
<description>Arctic Cooling, Accelero L2</description>
<quantity>1</quantity>
<price>18</price>
</product>
<product>
<sku>00000877</sku>
<vpn>RK007</vpn>
<description>Revoltex VGA Cooler RK007 V101</description>
<quantity>1</quantity>
<price>15</price>
</product>
</order>
```

Retrieve XML files

API URL: log in via the website and choose api for my account

Dates: RETRIEVEINFO:

?username=<username>&token=<token>&type=<documenttype>[&truncate=true] (GET)

Username: Username as it is in the webshop

Token: API token as provided by the web shop owner

Document type: Valid types: everything, invoice, quotation, order, repair, letter, packing slip,

backorder, customer list, bus stock, all, invoice, offer, order, rma, letter, packing slip

Truncate: Optional. Ensures that at the next request the information displayed now is not re-inserted

Invoice and other information can be retrieved with a pull-request. This can be done with a GET request

to the URL:

Example output if something goes wrong:

```
<?xml version="1.0" encoding="UTF-8"?>
<response date="2013-04-03 13:08:58">ERROR 0.0: Error message hier</response>
```

If the request is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<response date="2013-04-03 13:00:38">
<document type="order" date="2013-04-03 12:58:00" initialreference="4567890" reference="456789"
yourreference="852456">
<![CDATA[<xmldata></xmldata>]]>
</document>
<document type="invoice" date="2013-04-03 12:58:47" initialreference="1234567" reference="123456"
yourreference="654321">
<![CDATA[<xmldata></xmldata>]]>
</document>
</response>
```

The XMLdata is the same as in the XML-files sent by email.

Initialreference contains our original reference number, reference is our document number, yourreference is the reference number as you specified it with the order.