Milcom NAV Webservice

**Something about structure**

An order has a “1 to 1” header (=each order has a header)

An order also has a ”1 to 1” relation to shipToAdress (each order has a shipToAdress)

An order has a ”1 to 1” relation with a orderLineList. (each order has a orderLineList)

At last a orderLineList has a ”1 to many” relation to a orderLine. (=one ordrelist kan have many orders)

You can get an overview of the web service structure by running the call below (Enter correct account number and name)

<http://83.91.84.146:7049/DynamicsNAV/WS/>5980%20ScanSport/Codeunit/ScanSport

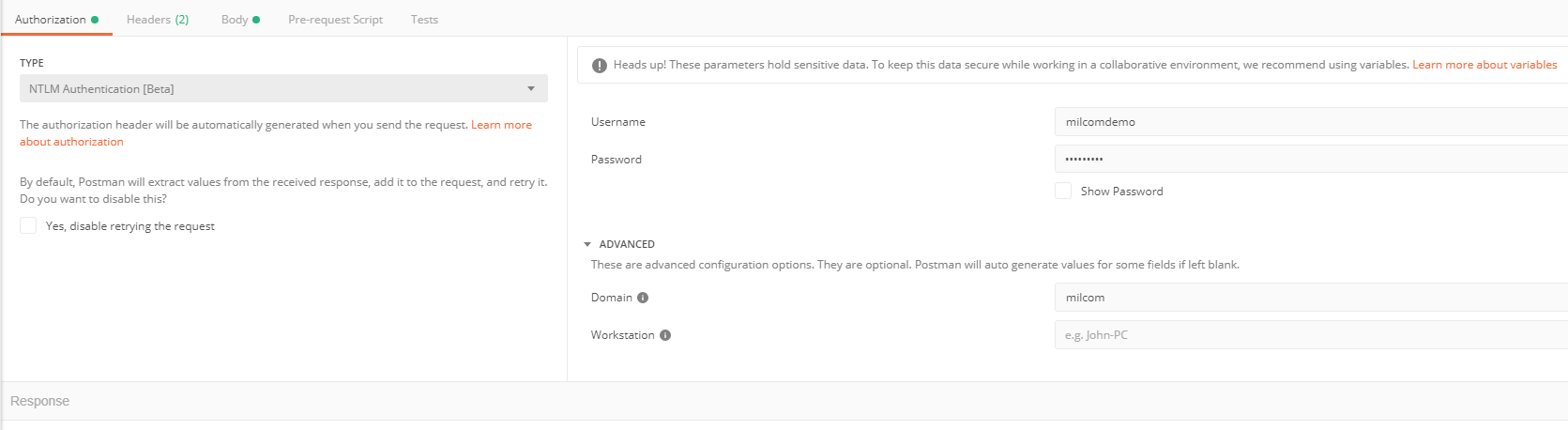
(Enter user and password)

Then you can, for example, see what an order should contain.<complexType mixed="**true**" name="**order**"><sequence><element type="**tns:header**" name="**header**" maxOccurs="**1**" minOccurs="**1**"/><element type="**tns:shipToAddress**" name="**shipToAddress**" maxOccurs="**1**" minOccurs="**1**"/><element type="**tns:orderLineList**" name="**orderLineList**" maxOccurs="**1**" minOccurs="**1**"/></sequence></complexType>

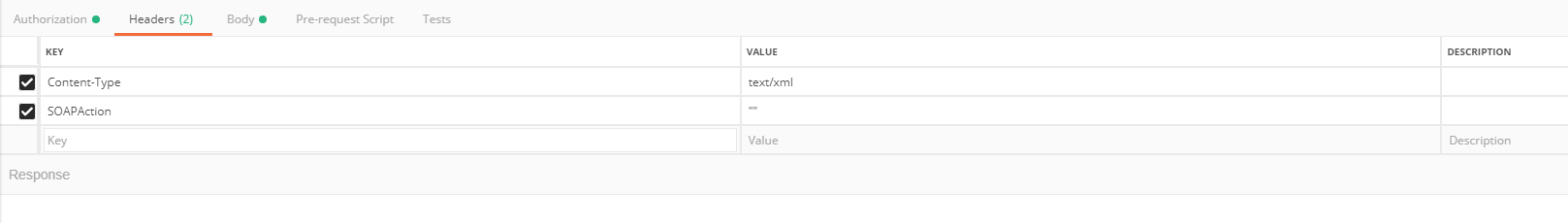
Then you can try to get through to the NAV database by running the HelloWorld function via the Postman program.

**Testing the web service through Postman.**

NTLM Authentication (not Basic Authentication) must be used.



Remember to put SOAPAction in the header.



**Call for Navision web service features.**

The SOAP header specifies the function name and parameters.

The following example is a call to the HelloWorld feature

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><HelloWorld xmlns="urn:microsoft-dynamics-schemas/codeunit/ScanSport"/> />

</soap:Body></soap:Envelope>

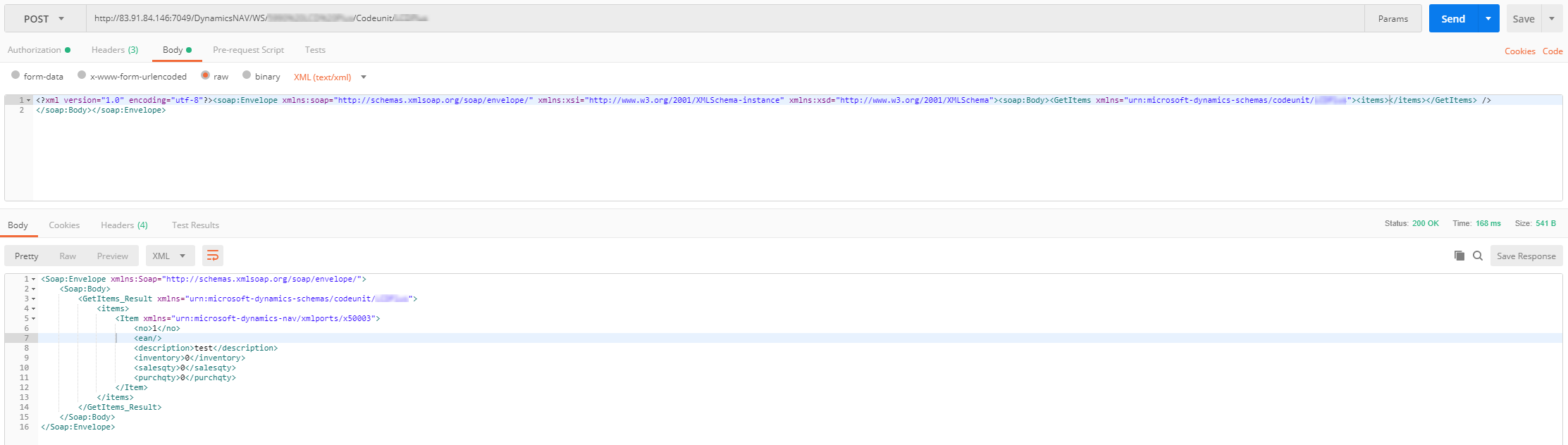


The following example is a call to the GetItems feature. Parameter is items (and always written in small print)

Insert the following into the header. (REMEMBER, enter correct accounting name)

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetItems xmlns="urn:microsoft-dynamics-schemas/codeunit/ScanSport"><items></items></GetItems> />

</soap:Body></soap:Envelope>



**Example from C # with the elements that need to be completed before a new order can be made**

”

      MilcomNav.header header = new MilcomNav.header();

            header.currency = "DKK";

            header.email = ["Test@test.dk"](mailto:Test@test.dk);

            header.orderDate = "08202017";  //MMDDYYYY

            header.orderNo = "test146";

            header.phoneNo = "22665544";

            header.sellToCustomerNo = "TestCustomer";

            header.shippingAgent = "GLS"; //Get list of valid values from Milcom

            header.yourReference = "customerReference";

            header.externalDocNo = "OrderRef1";

            MilcomNav.shipToAddress stAddress = new MilcomNav.shipToAddress();

            stAddress.address1 = "Roskildevej 17";

            stAddress.address2 = "";

            stAddress.city = "Roskilde";

            stAddress.contactName = "Peter Hansen";

            stAddress.country = "DK";

            stAddress.county = "";

            stAddress.name = "Company x";

            stAddress.postalNo = "4000";

            stAddress.pakkeShopID = "";

            MilcomNav.orderLineList lineList = new MilcomNav.orderLineList();

            lineList.orderLine = new MilcomNav.orderLine[2];

            lineList.orderLine[0] = new MilcomNav.orderLine();

            lineList.orderLine[0].itemNo = "test1";

            lineList.orderLine[0].itemName = "Test item";

            lineList.orderLine[0].lineType = "Item"; //Values are "G/L" for item charge. Otherwise choose Item

            lineList.orderLine[0].price = "110";

            lineList.orderLine[0].quantity = "1";

            lineList.orderLine[0].total = "100"; //Total line amount. If quantity \* price is not equal to this a line discount will be applied

            lineList.orderLine[1] = new MilcomNav.orderLine();

            lineList.orderLine[1].itemNo = "test1";

            lineList.orderLine[1].itemName = "Test item";

            lineList.orderLine[1].lineType = "Item"; //Values are "G/L" for item charge. Otherwise choose Item

            lineList.orderLine[1].price = "110";

            lineList.orderLine[1].quantity = "1";

            lineList.orderLine[1].total = "100"; //Total line amount. If quantity \* price is not equal to this a line discount will be applied

            order.header = header;

            order.shipToAddress = stAddress;

            order.orderLineList = lineList;

”

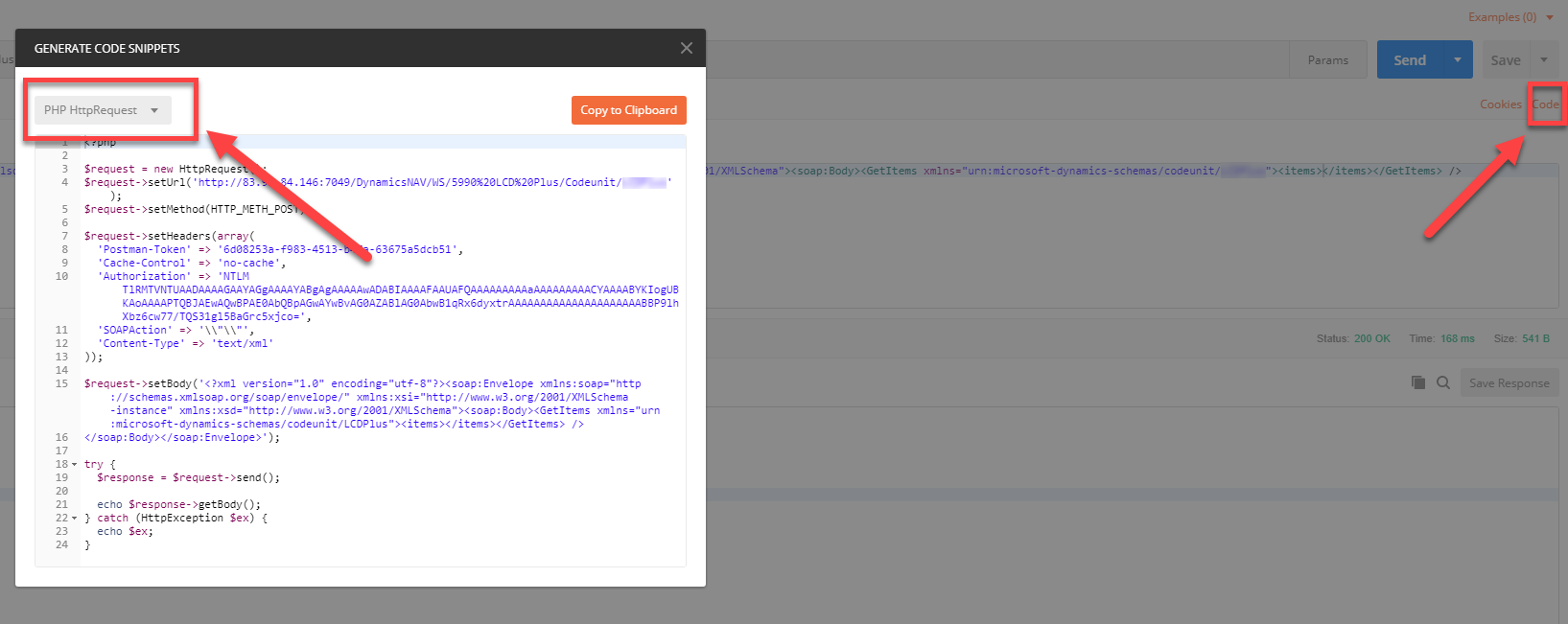
**Code examples in different web development languages.**

PHP example.

When implementing in PHP, through the Postman program you can create PHP code, which can make a web service call to NAV.

Here's an example of a call to the HelloWorld function. If it works, then there is a hole through to the NAV data. (REMEMBER to create url with correct accounting number and name).

In Postman, first select “code” on the right side and then select programming language in the dropdown box.



The code is then generated.

” <?php

$request = new HttpRequest();

$request->setUrl('http://83.91.84.146:7049/DynamicsNAV/WS/5980%20ScanSport/Codeunit/ScanSport');

$request->setMethod(HTTP\_METH\_POST);

$request->setHeaders(array(

'Postman-Token' => 'ad74b2d8-a4b5-44f9-ae1c-e3ade5ac5d54',

'Cache-Control' => 'no-cache',

'Authorization' => 'NTLM TlRMTVNTUAADAAAAGAAYAGgAAAAYABgAgAAAAAwADABIAAAAFAAUAFQAAAAAAAAAaAAAAAAAAACYAAAABYKIogUBKAoAAAAPTQBJAEwAQwBPAE0AbQBpAGwAYwBvAG0AZABlAG0AbwBfnoJ8JxMTtgAAAAAAAAAAAAAAAAAAAADyLFFu727gSSUbLZ31m9Ov44k3eXHvi6I=',

'SOAPAction' => '\\"\\"',

'Content-Type' => 'text/xml'

));

$request->setBody('<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetItems xmlns="urn:microsoft-dynamics-schemas/codeunit/ScanSport"><items>1</items></GetItems> />

</soap:Body></soap:Envelope>');

try {

$response = $request->send();

echo $response->getBody();

} catch (HttpException $ex) {

echo $ex;

}

”

**Info regarding new customers**

New customers can be created through the web service. The customer number can be entered by yourself and will typically be phone number or similar.

The customer number is a maximum of 20 digits long.

**New Order Info**

When creating new orders that do not have an associated customer (customer is not created). Then NAV will try to create a new customer where the customer number will be his phone number.

When creating is new order in which the item does not exist. Then the creation will fail.

The order number identifies an order and it is a maximum of 20 digits long.

**Installation of web server solution that communicates with the web service**

Experience with server installations where the following may be necessary to gain access to communicate with the web service.

- Run the solution on a dedicated IP

- Tie soap to the IP

- (Litespeed Server) Litespeed cache needed to be cleared, firewall opened for port 7079.