

SWAPI

(Softwear Web API)

Version: 1.4
Date: 20-06-2011
Author: Jeroen de Vries

Version history

- 21-10-2010: 1.4
 - Added barcode parameter to getStock
- 21-10-2010: 1.2
 - Added multiple-article capability to getStock
- 04-10-2010: 1.1
 - Added BLOCKCLIENT function
- 31-08-2010: 1.0.a
 - Added pwd parameter to SETCLIENT function
- 10-08-2010: 1.0
 - Changed all XML encoding from **utf-8** to **windows-1252**
 - GETCLIENT can now be called with id and/or email address
 - SETCLIENT now returns id of newly created client
 - Added GETINVOICE function
- 07-06-2010: Draft 0.9
 - Added SETCLIENT function
- 31-05-2010: Draft 0.8
 - Added GETCLIENT function
- 30-05-2010: Draft 0.7
 - Added SENDTRANSIT function
- 23-05-2010: Draft 0.6
 - Added perShop switch to GETSTOCK function
- 23-04-2010: Draft 0.5
 - Added GETSTATUS function
 - Added GETCOUPONS function
 - Added REDEEMCOUPON function
 - Added ADDCOUPON function
- 05-01-2010: Draft 0.4
 - Added RefID to SENDORDER function
 - Added CANCELORDER function
- 27-11-2009: Draft 0.3
 - Added Shipping and discount to SENDORDER function
- 20-11-2009: Draft 0.2
 - Added SENDORDER function
- 31-10-2009: Draft 0.1

The Softwear Web API (SWAPI) allows developers to interact with the Softwear ERP and POS systems.

The API is HTTP-based

All methods in SWAPI require a GET or POST request. HTTP response codes are not (yet) meaningful.

The API is a RESTful resource

SWAPI attempts to conform to the design principles of Representational State Transfer (REST). Most methods take the optional **format** parameter that allows you to get results in the format of your choice.

SWAPI currently supports the following data formats: XML, HTML and SOAP.

Test v.s production

For testing, <http://testapi.softwear.nl>.

For production purposes, use <http://api.softwear.nl>

If the testapi is used, return values will be the same as the production api however, no changes are made in the database.

Please note that the performance of the testapi is significantly less than the production api.

DATAFEEDS

In addition to the api, several datafeeds can be downloaded upon request. Datafeeds are available for article information and client information. Datafeeds are refreshed automatically every 24 hrs. All datafeeds are in CSV format.

The content of the datafeeds can vary per project, for more information contact us at info@softwear.nl or 0900-SOFTWEAR.

The example Datafeed with the shop Id=246 can be reached at:

<http://www.softwear.nl/ism/bebees/osws.csv>

function getStock.

This method is used to retrieve the actual stock of an article.

Syntax 1:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]getstock&[token=]<shopid>[&articlecode=<articlecode>][&barcode=<barcode>][&format=<xml/html>][&switches=<logical/detail/perShop>]`

Syntax 2: `http://testapi.softwear.nl/getstock/<shopid>/articlecode__<articlecode>/barcode__<barcode>[/format__<xml/html>][/switches__<logical/detail/perShop>]`

elements in [] are optional

elements in <> should be replaced by literals

Example:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?getstock&token=246&articlecode=0000002&format=xml&switches=logical`

parameter: token:string

3-digit shopid

parameter: articlecode:string

Articlecode of article for which stock should be returned.

Articlecode is case-sensitive.

Articlecode can be either a single articlecode or a comma separated list of articlecodes.

parameter: barcode:string

Barcode of article for which stock should be returned.

Barcode is case-sensitive.

Barcode can be either a single barcode or a comma separated list of barcodes.

Can be any barcode of article, stock for all variations of article will be returned.

parameter: format:string

xml returns result as XML (default)

soap returns result in SOAP envelope

html returns result as HTML

parameter: switches:string

comma separated string of switches.

logical returns result as 1 or 0. 1 Means that there is at least one piece of that article in stock, regardless of size and color. The logical option can only be used when stock is retrieved for a single articlecode.

detail returns result per SKU (default)

perShop returns results per shop (implies detail)

function sendOrder.

This method is used to place an order in a Softwear backoffice.

Syntax 1:

```
http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]sendorder&[token=]<shopid>&custnum=<custnum>&email=<email>&title=<title>&firstname=<firstname>&middlename=<middlename>&lastname=<lastname>&street=<street>&houenum=<houenum>&houseadd=<houseadd>&zipcode=<zipcode>&city=<city>&countrycode=<countrycode>&remarks=<remarks>&shipping=<shipping>&discount=<discount>[&refid=<refid>][&articlecode=<articlecode>&color=<color>&size=<size>&qty=<qty>&price=<price>]*
```

elements in [] are optional

elements in <> should be replaced by literals

Example:

```
http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?sendorder&token=246&custnum=1234&email=jeroen@softwear.nl&title=Dhr&firstname=Jeroen&middlename=de&lastname=Vries&street=Veerdijk&houenum=42&houseadd=e&zipcode=15312MS&city=WORMER&countrycode=NL&remarks=Niet%20op%20maandag%20leveren&shipping=495&discount=1000&refid=65748392&articlecode=12345&color=200/600&size=68&qty=1&price=2995&articlecode=12345&color=200/600&size=74&qty=2&price=3995
```

parameter: token:string

3-digit shopid

parameter: custnum:string

Customernumber of client. Returning customers should be addressed by their existing emailadres, not necessarily their customernumber. Otherwise, a new customers will be created in Softwear.

parameter: email:string

emailaddress of customer

parameter: title:string

Title of customer. Usually something like Mr/Mrs/Ms etc...

parameter: firstname:string

Customers firstname or initials

parameter: middlename:string

Customers middlename

parameter: lastname:string

Customers last- or familyname

parameter: street:string

Deliveryaddress streetname

parameter: houenum:string

Deliveryaddress house number

parameter: houseadd:string

Deliveryaddress housenumber addition

parameter: zipcode:string

Deliveryaddress zip or postal code

parameter: city:string

Deliveryaddress city

parameter: countrycode:string

Deliveryaddress countrycode.

Coded against ISO 3166: http://www.iso.org/iso/english_country_names_and_code_elements

parameter: remarks:string

Remarks

parameter: shipping:number

Shipping fee in eurocents or the actual fee times 100

Eg. € 4,95 would be represented as 495.

Shipping includes VAT

parameter: discount:number

Discount amount in eurocents or the actual amount times 100.

Eg. € 6,25 would be represented as 625.

This discount applies to the whole order, not to the individual articles listed below. Discount includes VAT

parameter: refid:string(10)

Reference order number.

This number uniquely identifies the order. If this number is supplied with this function, the same number can be used with the **cancelorder** function to cancel an order in the backoffice.

parameter: articlecode:string

Articlecode as found in datafeed.

parameter: color:string

Colorcode as found in datafeed.

parameter: size:string

Size as found in datafeed. When mailcolor/sucolor or mailcolor/cup are both used, concatenate them both with a slash (/) in between .

parameter: qty:integer

Nr. Pieces sold of most recently mentioned article

parameter: price:integer

Price of most recently mentioned article times 100

Price includes VAT

The previous 5 parameters can be repeated as many times as needed to express the complete salesorder.

The order of these parameters is significant: articlecode, color, size, qty, price.

Return values

Results are returned in HTML.

On success, OK is returned, otherwise a human readable error message is returned.

Remarks:

Optionally, the sendOrder API can be configured to accept amounts excluding VAT.

function cancelOrder.

This method is used to cancel an existing order in a Softwear backoffice.

Syntax 1:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]cancelorder&[token=]<shopid>&refid=<refid>`

elements in [] are optional

elements in <> should be replaced by literals

Example:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?cancelorder&token=246&refid=65748392`

parameter: token:string

3-digit shopid

parameter:refid:string(10)

Reference order number.

This number uniquely identifies the order. It is the same number originally supplied as **refid** when the order was created with the **sendorder** function.

function getStatus.

This method is used to retrieve the historical transactions of a customer.

Syntax 1:

http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]getstatus&[token=]<shopid>&email=<email>[&format=<xml/html>]

Syntax 2: http://testapi.softwear.nl/getstatus/<shopid>/email__<email>[/format__<xml/html>]

elements in [] are optional

elements in <> should be replaced by literals

Example:

http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?getstatus&token=246&email=jeroen@softwear.nl&format=xml

parameter: token:string

3-digit shopid

parameter: email:string

Emailaddress of client.

parameter: format:string

xml returns result as XML (default)

html returns result as HTML

function getCoupons.

This method is used to retrieve the coupons status of a customer.

Syntax 1:

http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]getcoupons&[token=]<shopid>&email=<email>[&format=<xml/html>]

Syntax 2: http://testapi.softwear.nl/getcoupons/<shopid>/email__<email>[/format__<xml/html>]

elements in [] are optional

elements in <> should be replaced by literals

Example:

http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?getcoupons&token=246&email=jeroen@softwear.nl&format=xml

parameter: token:string

3-digit shopid

parameter: email:string

Emailaddress of client.

parameter: format:string

xml returns result as XML (default)

soap returns result in SOAP envelope

html returns result as HTML

function redeemCoupon.

This method is used to redeem (part of) an existing coupon.

Syntax 1:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]redeemcoupon&[token=]
<shopid>&couponid=<couponid> [&value=<value>][&format=<xml/html>]`

Syntax 2: `http://testapi.softwear.nl/redeemcoupon/<shopid>/couponid__<couponid>
/[value__<value>][format__<xml/html>]`

elements in [] are optional

elements in <> should be replaced by literals

Example:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?redeemcoupon&token=246&email=j
eroen@softwear.nl&format=xml`

parameter: token:string

3-digit shopid

parameter: couponid:string

unique id of existing coupon.

parameter: value:string

Value to subtract from coupon in eurocents.

Of value is omitted, the entire coupon is redeemed.

Value can only be specified for monetary coupons, not for percentage coupons. Percentage coupons can only be redeemed entirely.

parameter: format:string

xml returns result as XML (default)

soap returns result in SOAP envelope

html returns result as HTML

function addCoupon.

This method is used to create a new coupon for an existing customer.

Syntax 1:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]addcoupon&[token=]<shopid>&email=<email>&couponid=<couponid>&value=<value>[&format=<xml/html>]`

Syntax 2:

`http://testapi.softwear.nl/addcoupon/<shopid>/email__<email>/couponid__<couponid>/value__<value>[/format__<xml/html>]`

elements in [] are optional

elements in <> should be replaced by literals

Example:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?addcoupon&token=246&email=jeroen@softwear.nl&couponid=1234567890/value=1000&format=xml`

parameter: token:string

3-digit shopid

parameter: email:string

Emailaddress of an existing client.

parameter: couponid:string

unique id for new coupon.

parameter: value:string

Coupon value in eurocents.

parameter: format:string

xml returns result as XML (default)

soap returns result in SOAP envelope

html returns result as HTML

function sendTransit.

This method is used to create a transit order in a Softwear backoffice.

Syntax 1:

```
http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]sendtransit&[token=]<shopid>&from=<sourceshop>&to=<destinationshop>&articlecode=<articlecode>&color=<color>&size=<size>&qty=<qty>
```

elements in [] are optional

elements in <> should be replaced by literals

Example:

```
http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?sendtransit&token=246&from=Amsterdam&to=Wormer&articlecode=12345&color=200/600&size=74&qty=2
```

parameter: token:string

3-digit shopid

parameter: from:string

Shopname where transit is originating. Shopname is case sensitive and should be obtained from the getstock function used with the perShop switch

parameter: to:string

Shopname where transit is going to. Shopname is case sensitive and should be obtained from the getstock function used with the perShop switch

parameter: articlecode:string

Articlecode as found in datafeed.

parameter: color:string

Colorcode as found in datafeed.

parameter: size:string

Size as found in datafeed.

parameter: qty:integer

Nr. Pieces sold of most recently mentioned article

Return values

Results are returned in HTML.

On success, OK is returned, otherwise a human readable error message is returned.

function getClient.

This method is used to receive the address information of a customer.

Syntax 1:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]getClient&[token=]<shopid>[&email=<email>][&id=<id>][&format=<xml/html>][&switches=<list>]`

Syntax 2:

`http://testapi.softwear.nl/getclient/<shopid>[email__<email>][format__<xml/html>][switches__<logical>]`

elements in [] are optional

elements in <> should be replaced by literals

Example:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?getClient&token=246&email=jeroen@softwear.nl&format=xml`

parameter: token:string

3-digit shopid

parameter: email:string

Emailaddress of client.

parameter: format:string

xml returns result as XML (default)

html returns result as HTML

parameter: switches:string

comma separated string of switches.

list returns a list emailaddresses for all active clients

detail returns details for client specified by email address parameter (default)

function setClient.

This method is used to add or update a client order in a Softwear backoffice.

Syntax 1 for use with Softwear retail systems:

[http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?\[function\]=setclient&\[token\]=<shopid>&custnum=<custnum>&email=<email>&title=<title>&firstname=<firstname>&middlename=<middlename>&lastname=<lastname>&street=<street>&houenum=<houenum>&houseadd=<houseadd>&zipcode=<zipcode>&city=<city>&pwd=<pwd>&countrycode=<countrycode>](http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function]=setclient&[token]=<shopid>&custnum=<custnum>&email=<email>&title=<title>&firstname=<firstname>&middlename=<middlename>&lastname=<lastname>&street=<street>&houenum=<houenum>&houseadd=<houseadd>&zipcode=<zipcode>&city=<city>&pwd=<pwd>&countrycode=<countrycode>)

Syntax 2 for use with Softwear wholesale systems:

[http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?\[function\]=setclient&\[token\]=<shopid>&custnum=<custnum>&email=<email>&title=&name=<name>&street=<street>&houenum=<houenum>&houseadd=<houseadd>&zipcode=<zipcode>&city=<city>&countrycode=<countrycode>](http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function]=setclient&[token]=<shopid>&custnum=<custnum>&email=<email>&title=&name=<name>&street=<street>&houenum=<houenum>&houseadd=<houseadd>&zipcode=<zipcode>&city=<city>&countrycode=<countrycode>)

elements in [] are optional

elements in <> should be replaced by literals

Example:

<http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?setclient&token=246&custnum=1234&email=jeroen@softwear.nl&title=Dhr&firstname=Jeroen&middlename=de&lastname=Vries&street=Veerdijk&houenum=42&houseadd=e&zipcode=15312MS&city=WORMER&countrycode=NL>

parameter: token:string

3-digit shopid

parameter: custnum:string

Customernumber of client. Returning customers should be addressed by their existing emailadres, not necessarily their customernumber. Otherwise, a new customers will be created in Softwear.

parameter: email:string

emailaddress of customer

parameter: title:string

Title of customer. Usually something like Mr/Mrs/Ms etc...

parameter: firstname:string

Customers firstname or initials

parameter: middlename:string

Customers middlename

parameter: lastname:string

Customers last- or familyname

parameter (wholesale):name:string

Company name

parameter (wholesale):deliveryaddress:string

Delivery address

parameter (wholesale):deliverycountrycode:string

Delivery countrycode

parameter (wholesale):deliverycity:string

Delivery city

parameter (wholesale):deliverytelephone:string

Delivery telephone

parameter (wholesale):deliveryzipcode:string

Delivery zipcode

parameter (wholesale):vatnr:string

VAT nr

parameter: street:string

Deliveryaddress streetname

parameter: housenum:string

Deliveryaddress house number

parameter: houseadd:string

Deliveryaddress housenumber addition

parameter: zipcode:string

Deliveryaddress zip or postal code

parameter: city:string

Deliveryaddress city

parameter: password:string

Password. If not supplied, Softwear will generate a password.

parameter: countrycode:string

Deliveryaddress countrycode.

Coded against ISO 3166: http://www.iso.org/iso/english_country_names_and_code_elements

function getInvoice.

This method is used to receive invoice data for a particular day from Software.

Syntax 1:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]getinvoice&[token=]<shopid> [&year]<year> [&month]<month> [&day]<day>][&format=<xml/html>]`

Syntax 2:

`http://testapi.softwear.nl/getclient/<shopid>[/year__<year>][month__<month>][day__<day>][format__<xml/html>]`

elements in [] are optional

elements in <> should be replaced by literals

Example:

`http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?getclient&token=246&year=2010&month=8&day=12&format=xml`

parameter: token:string

3-digit shopid

parameter: year:number

Year to query.

parameter: month:number

Month to query.

parameter: day:number

Day to query.

parameter: format:string

xml returns result as XML (default)

html returns result as HTML

function blockClient.

This method is used to block/unblock a client in a Softwear backoffice.

Syntax 1:

[http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?\[function=\]blockclient&\[token=\]<shopid>&custnum=<custnum>&blocked=<true/false>](http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?[function=]blockclient&[token=]<shopid>&custnum=<custnum>&blocked=<true/false>)

elements in [] are optional

elements in <> should be replaced by literals

Example:

http://testapi.softwear.nl/scripts/foxisapi.dll/sww1.wreq1.mpx?setclient&token=246&custnum=1234&blocked=true

parameter: token:string

3-digit shopid

parameter: custnum:string

Customernumber of client.

parameter: blocked:string

true to block a client, false to unblock