Getting Started with Web Services

Understanding Our Web Services

Our web services are XML-based and can be called in either GET, POST, or SOAP protocols. We recommend GET or POST, or a combination of the two and would only resort to SOAP if absolutely necessary.

Our Web Services have three distinct sections:

- **Inventory** – Obtaining Product Data
- **Orders** – Creating and Submitting Orders
- **Invoices** – Obtaining Post-Invoicing Data, such as invoice, credit and tracking details

If you are a third-party provider and will have multiple Sports South clients, it is recommended to maintain a single database of our product data that you in turn feed to your clients. Our product data is the same for all clients, with the exception of pricing (customer specific). So, you'll need a way to store pricing on a per-customer level if you choose to go this route.

We recommend you update prices each time you run an inventory update (quantity on hand), due to the possibility of price changes more often than product data. It is for this reason we include pricing in the "Onhand" methods.

Credentials For Testing

UserName: 99994
CustomerNumber: 99994
Password: 99999
Source: *

Live Credentials

UserName: Sports South Account Number
CustomerNumber: Sports South Account Number
Password: Sports South Web Services Password
Source: *

** If you are a third-party provider, the source should be an abbreviation of your company or software name and no greater than six characters. If you are not a provider, use your Sports South account number for the source.*

When you are ready to go live, you may need to contact our help desk and request credentials. Email help.desk@sportssouth.biz for these requests.

Inventory

http://webservices.theshootingwarehouse.com/smart/inventory.asmx

The link above will list all available methods for this web service section. Please keep in mind that some of these may be deprecated or outdated. As you can see, many of these methods appear to be duplicated with a "DS" suffix. This simply means this method will return a structured Data Set instead of the standard XML. Also, many of the method descriptions will have a "Click Here" link that will display its schema.

- DailyItemUpdate is the main method for obtaining product data. It is meant to be called once a day to match our update frequency. The "**LastUpdate**" field is a date field meant for the last day that you called the method. The return will only include products added/modified since that date. To get a full update, use *1/1/1990* for the date. By default, this method will return 1000 products at a time. The "**LastItem**" field is meant for the last item number in your previous return so that the next return will know where to start. If you want to bypass the 1000 per call limit, use -1 and you will get the full return.

- CategoryUpdate is used in conjunction with DailyItemUpdate to determine product categories and attributes. Link these two with the "**CATID**" field. The attribute values are returned with each product in the DailyItemUpdate while the attribute labels are returned at the per category level in the CategoryUpdate method. "**ATTR1"** goes with "**ITATR1"** and so on.

- BrandUpdate is used to get the Brand Names. DailyItemUpdate's "**ITBRDNO**" matches with BrandUpdate's "**BRDNO**".

- GetText is used to get the long paragraph descriptions. Use the "**TXTREF**" field in DailyItemUpdate when calling GetText.

- ListNewText is used to get long paragraph description updates. It uses a date field like DailyItemUpdate.

- IncrementalOnhandUpdate is the recommended main method for obtaining inventory numbers. It is also designed to only return the delta but allows for a date/time input down to the minute. We update these records every five minutes, so your frequency can be anything greater than that, although we do recommend you do five minutes. Example of formatting for the **SinceDateTime** field: "*2020-12-12T15:05:00.00+00.00*" without the quotes. All our Onhand methods return two prices along with the onhand data. The "C" price is the price we will be charging on the invoice, so this should be used as the base prices before markups are added. As stated above, it is recommended to update pricing via these methods rather than in the DailyItemUpdate method.

- OnhandInquiry is used to return inventory for one product at a time. Not recommended if you have more than one item you need inventory for.

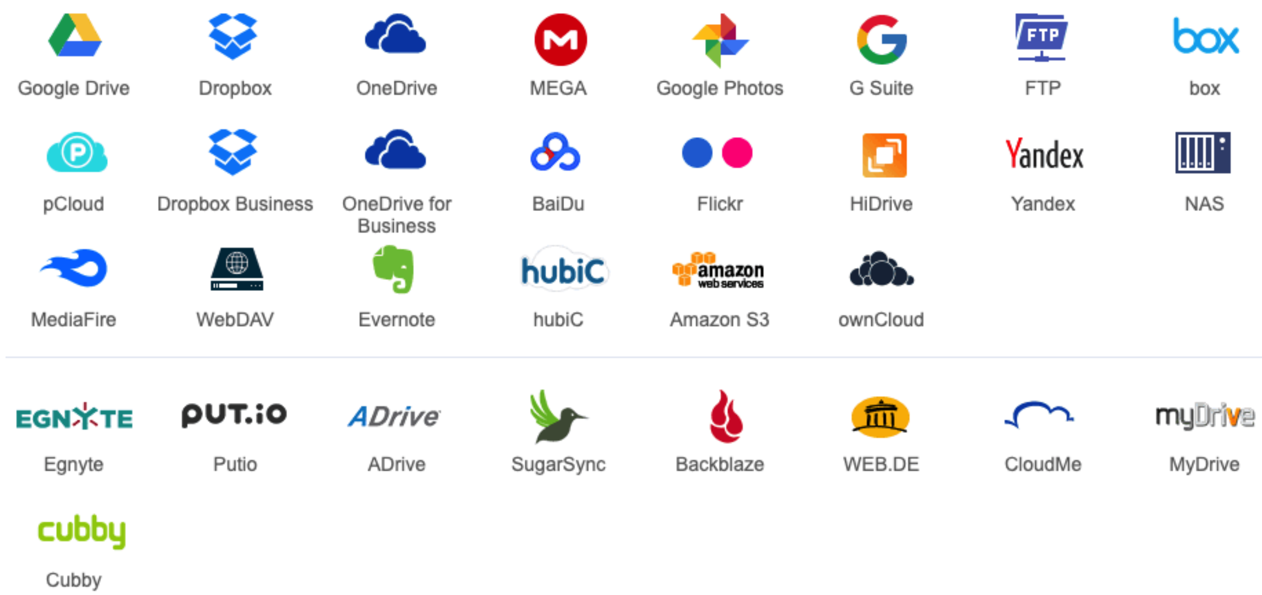
- OnhandUpdateByCSV is used when you have a handful of items. You pass them through the **CSVItems** field separated by commas. This is recommended to check the inventory levels for items in a user's shopping cart.

- OnhandUpdate will return the inventory for our full set of products. This is only recommended for cleanup purposes. Some developers like to set this up on a "daily" or a "couple of times a week" routine to make sure there were no hiccups in the IncrementalOnhandUpdate deltas.

Images

You may come across web service methods for obtaining images, please do not use these as they are all deprecated. We currently have two ways to obtain our product images:

1) We can setup a routine to sync the images to an FTP or a cloud service (Dropbox, Drive, S3, etc.) on your side. This could be setup to run a couple of times a week and would add/update images. We will need access credentials for the FTP or cloud service if you opt for this solution. Available connections:



2) You can use the field **PICREF** from DailyItemUpdate and download the images directly from our media server. If no **PICREF** is available for a product, fall back to the **ITEMNO**. We currently have the following image sizes:

High-Res PNG - *https://media.server.theshootingwarehouse.com/hires/PICREF.png*
Large JPG - *https://media.server.theshootingwarehouse.com/large/PICREF.jpg*
Small JPG - *https://media.server.theshootingwarehouse.com/small/PICREF.jpg*
Thumbnail JPG - *https://media.server.theshootingwarehouse.com/thumbnail/PICREF.jpg*

As you can see from the link formats above, you simply swap out the “PICREF” text for each item. Please notice the High-Res images are .png.

Orders

http://webservices.theshootingwarehouse.com/smart/orders.asmx

Creating an Order

1) Call the AddHeader method:

Fulfillment (dropship) – Non-Firearms Only ***

CustomerNumber	Required	Sports South Account Number
UserName	Required	Sports South Account Number
Password	Required	Sports South Web Services Password
Source	Required	Sports South Account Number
PO	Required	Unique Purchase Order Number
OrderNumber	Optional	Your System's Order Number
SalesMessage	Optional	Message to your Sales Rep
ShipVia	Required	One of the following: (*empty*, G, 2, N) *
ShipToName	Required	Customer or Company Name
ShipToAttn	Required	2nd ShipToName, If Not Needed, Copy ShipToName from Above
ShipToAddr1	Required	Customer Address Line 1
ShipToAddr2	Optional	Customer Address Line 2
ShipToCity	Required	Customer City
ShipToState	Required	Customer State
ShipToZip	Required	Customer Zip Code
ShipToPhone	Required	Customer Telephone Number
AdultSignature	Required	"True" or "False" – If True, Adult Signature Required **
Signature	Required	"True" or "False" – If True, Signature Required **
Insurance	Required	"True" or "False" – If True, Insured Amount Defaults to Invoice Total **

Ship to Store (address on file)

CustomerNumber	Required	Sports South Account Number
UserName	Required	Sports South Account Number
Password	Required	Sports South Web Services Password
Source	Required	Sports South Account Number
PO	Required	Unique Purchase Order Number
OrderNumber	Optional	Your System's Order Number
SalesMessage	Optional	Message to your Sales Rep
ShipVia	Required	One of the following: (*empty*, G, 2, N) *
ShipToName -> ShipToState	Required	Leave Empty
ShipToZip	Required	0
ShipToPhone	Required	Leave Empty
AdultSignature	Required	"True" or "False" – If True, Adult Signature Required **
Signature	Required	"True" or "False" – If True, Signature Required **
Insurance	Required	"True" or "False" – If True, Insured Amount Defaults to Invoice Total **

** See Shipping Code Chart below for more information.*

*** Additional Charges will apply. See "Order Fulfillment Overview" document for more information.*

**** Firearm Fulfillment requires additional approval and adjustments to order placement procedures. Please contact help.desk@sportssouth.biz for details.*

2) If the order was successfully created, a Sports South Web Order number will be returned:

<int>8000000</int>

If the order was not created, an error will be returned.

3) Call the AddDetail method:

OrderNumber	**Required**	Sports South Web Order Number *(returned with AddHeader)*
CustomerNumber	**Required**	Sports South Account Number
UserName	**Required**	Sports South Account Number
Password	**Required**	Sports South Web Services Password
Source	**Required**	Sports South Account Number
SSItemNumber	**Required**	Sports South Item Number *(ITEMNO from DailyItemUpdate)*
Quantity	**Required**	Quantity Requested of This Item
OrderPrice	**Required**	Expected Price *(for reporting purposes only)*
CustomerItemNumber	**Optional**	Your System's Item Number
CustomerItemDescription	**Optional**	Your System's Item Description

4) If the item was successfully added to the order, a Boolean value of "true" will be returned:

<boolean>true</boolean>

If the item was NOT successfully added to the order, a Boolean value of "false" will be returned:

<boolean>false</boolean>

5) Repeat steps 3 & 4 for each additional item that is on the order.

6) Call the Submit method:

OrderNumber	**Required**	Sports South Web Order Number *(returned with AddHeader)*
CustomerNumber	**Required**	Sports South Account Number
UserName	**Required**	Sports South Account Number
Password	**Required**	Sports South Web Services Password
Source	**Required**	Sports South Account Number

7) If the order was successfully submitted, a Boolean value of "true" will be returned:

<boolean>true</boolean>

If the order was NOT successfully submitted, a Boolean value of "false" will be returned:

<boolean>false</boolean>

Confirming an Order

1) Call the GetDetail method:

CustomerNumber	**Required**	Sports South Account Number
UserName	**Required**	Sports South Account Number
Password	**Required**	Sports South Web Services Password
OrderNumber	**Required**	Sports South Web Order Number *(returned with AddHeader)*
CustomerOrderNumber	**Optional**	Your System's Order Number
Source	**Required**	Sports South Account Number

2) XML will be returned:


```
<Table>
  <ORDNO>1119886</ORDNO>
  <ORCUST>99994</ORCUST>
  <ORITEM>25845</ORITEM>
  <ORQTY>1</ORQTY>
  <ORPRC>1.00</ORPRC>
  <ORQTYF>1</ORQTYF>
  <ORPRCF>19.50</ORPRCF>
  <ORCUSI />
  <ORCUSD />
</Table>
```


3) Each **ORQTYF** (Quantity Filled) should match it's corresponding **ORQTY** (Quantity Ordered) for a complete order. If **ORQTYF** is less than its **ORQTY** counterpart, that item was shorted on the order. Shorts can be partial or full. We will ship what we can (fill and kill).

Shipping Code Chart

Service	ShipVia Code	
Ground Economy	*empty*	May include discount carrier programs/services. *
Ground Premium	G	Shipped via major carrier. *
2-Day Air	2	
Next-Day Air	N	

** See "Order Fulfillment Overview" document for more details and restrictions.*

Invoices

http://webservices.theshootingwarehouse.com/smart/invoices.asmx

The link above will list all available methods for this web service section. Please keep in mind that some of these may be deprecated or outdated. As you can see, many of these methods appear to be duplicated with a "DS" suffix. This simply means this method will return a structured Data Set instead of the standard XML. Also, many of the method descriptions will have a "Click Here" link that will display its schema

This section contains methods used to check invoices and obtain tracking/package details. In our system, an order becomes an invoice once it is released and sent to the warehouse.

- There are numerous methods for obtaining invoice details. These methods being with “GetBy” and have different means to lookup an invoice. Such as GetByDate, GetByOrderNumber, and GetByPONumber.
    - A sample return for these methods can be viewed here: http://webservices.theshootingwarehouse.com/smart/schemas/Invoice_GetByDate.xsl

- GetCreditsByDate is used to routinely check for credits that may be issued. It is recommended to call this method daily with yesterday’s date. If a credit is returned, you can match it up with the order via either the Invoice Number or PO Number. The details returned will show what item was credited and the reason for the credit.

- GetTrackingByPO is recommended for obtaining tracking details. It is for this reason a unique PO should be used when creating the order. Tracking is updated hourly, so it is recommended to have a routine that checks this method hourly for all POs that have not yet received tracking. Once a tracking number is returned, remove the PO from the routine.

- GetPackageContents is used to determine which products are in each package. This is only needed if GetTrackingByPO returns multiple tracking numbers.