|  |
| --- |
| STELLAE INTERNATIONAL INC. |
| WEB SERVICE |
| DOCUMENTATION |

|  |
| --- |
| Don  Monday, March 16, 2015 |

# Stellae Web Services

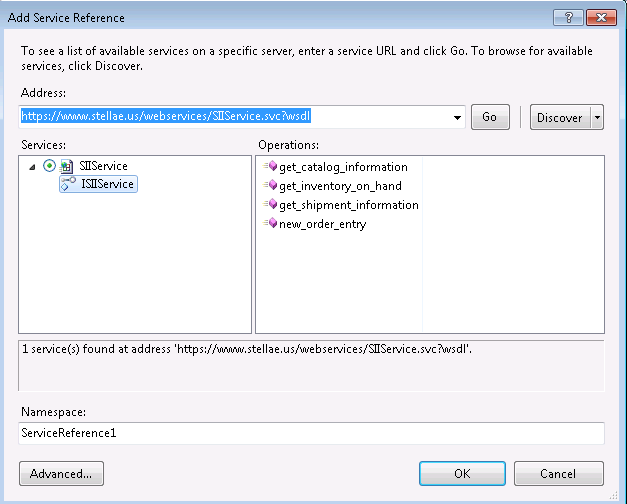
## Web Address

<https://www.stellae.us/webservices/SIIService.svc>

Please use this for testing:

https://www.stellae.us/webservicestest/SIIService.svc

Note: there have been some connection issues, and if you are using Microsoft Visual Studio, we are using a service reference, not a web reference. I believe this is correct (this is Visual Studio 2010) although I have not tested this method:



PHP developers:

1. We suggest you review this web site to understand some of the procedures that can be used to connect to Stellae and its WCF based web services.
   1. <https://social.msdn.microsoft.com/Forums/vstudio/en-US/daf37585-fcdb-4a6e-8d7b-5f990c1c49bd/how-to-consume-wcf-service-from-php-client-?forum=wcf>
   2. We have a URL that supports 2 different bindings, many users report success in connecting to the BasicHttpBinding\_ISIIService
   3. The wsdl https://www.stellae.us/webservicesmulti/SIIService.svc?wsdl
2. Alternatively:
   1. Stellae uses WS-\* Web Service, therefore you can't connect with PHP directly to the Stellae Web Service because PHP doesn't support WS-Security policy, WS-Security and WS-Secure.
   2. To circumvent this restriction, you will have to install on the Web Server the WSO2 Framework ( [http://wso2.com/products/web-services-framework/php/](http://wso2.com/products/web-services-framework/php/" \t "_blank) ). The installation is a bit tricky please follow the installation instructions carefully. We have heard from some developers that this is not being supported any longer, so (1) above should be investigated first.

## Order Import

The purpose of this method is to create an order on the Stellae system to allow the fulfillment of the order.

Input message:

User name

Order Information

### User Name

|  |  |
| --- | --- |
| User | String |
| Password | String |

Order Information

|  |  |  |
| --- | --- | --- |
| CUSTOMER\_NAME | String(10) | Bill to customer name |
| CUSTOMER CODE | String(10) | Bill to customer code – if blank, will assign |
| CUSTOMER\_ADDRESS\_1 | String(30) | Bill to customer address1 |
| CUSTOMER\_ADDRESS\_2 | String(30) | Bill to customer address2 |
| CUSTOMER\_ADDRESS\_3 | String(30) | Bill to customer address3 |
| CUSTOMER\_ADDRESS\_CITY | String(30) | Bill to customer city |
| CUSTOMER\_ADDRESS\_STATE | String(4) | Bill to customer state |
| CUSTOMER\_ADDRESS\_ZIP | String(10) | Bill to customer zip code |
| CUSTOMER\_ADDRESS\_COUNTRY | String(40) | Bill to customer country code |
| CUSTOMER\_TELEPHONE | String(20) | Bill to customer telephone |
| DELIVERY\_NAME | String(40) | Bill to delivery name |
| DELIVERY\_ADDRESS\_1 | String(40) | Bill to delivery address1 |
| DELIVERY\_ADDRESS\_2 | String(40) | Bill to delivery address2 |
| DELIVERY\_ADDRESS\_3 | String(40) | Bill to delivery address3 |
| DELIVERY\_ADDRESS\_CITY | String(40) | Bill to delivery city |
| DELIVERY\_ADDRESS\_STATE | String(4) | Bill to delivery state |
| DELIVERY\_ADDRESS\_ZIP | String(10) | Bill to delivery zip code |
| DELIVERY\_ADDRESS\_COUNTRY | String(40) | Bill to delivery country code |
| DELIVERY\_TELEPHONE | String(20) | Bill to delivery telephone |
| DELIEVERY\_ID | string | (not currently used) |
| DELIVERY\_MESSAGE | String(128) | Comment |
| EMAIL | string | Used to send to fedex/ups |
| ORDER\_ID | String(10) | Order number (sending system order number). Stellae will store this information and have use a different order number internally |
| ORDER\_TYPE | String(2) | OO – new order; CM – returns |
| CARRIER | String(20) | Preferred carrier |
| SERVICE | String(10) | Carrier service code |
| FREIGHT\_ACCOUNT | String(128) | On bill to receipt, recipient’s account |
| TAXES | Numeric | Total tax amount (store in SII system in either misc1 or misc2) |
| SHIPPING\_FEES | Numeric | Shipping charges |
| DISCOUNT | Numeric | Total discount $ |
| MISC1 | Numeric | Misc charge |
| MISC1\_REASON | String | Reason for misc charge |
| MISC2 | Numeric | Misc charge |
| MISC2\_REASON | String | Reason for misc charge |
| TOTAL\_AMOUNT | Numeric | Total amount |
| CURRENCY | String(3) | Currency |
| USER1 | String(255) | Invoice url |
| USER2 | String(255) |  |
| USER3 | String(255) |  |
| USER4 | String(255) |  |
| USER5 | String(255) |  |
| Order\_Details | Array() | Items ordered |

Order Details:

|  |  |  |
| --- | --- | --- |
| SKU | String(18) | UPC |
| QUANTITY | Numeric | Quantity ordered |
| SIZE | String | (e.g. small). Not used |
| PRICE | Numeric |  |
| FLAGS | String(15) | Flags concerning item |
|  |  |  |

Output message:

|  |  |  |
| --- | --- | --- |
| Status |  |  |
| Sii\_order\_number |  |  |
| Detail\_status | String() | Per line errors |

An output status of 0001 means that the order was accepted.

Note, in the Stellae system, there isn’t a field for total tax amount (it is kept on a line item basis), so any non zero amount is moved to either misc1 or misc2 (if misc1 has a value). As a result, you cannot fill in taxes and both miscellaneous charge fields. The total amount field should be the total amount of the order, however, at the current time this is not validated. If currency is not filled in, it will be based on system parameters internal to the Stellae system.

The USER1 field should contain an URL that can be used to generate a PDF. We use the (Microsoft) HttpUtility.UrlDecode function prior to using the url, which will translate / to %2F etc. We have had some issues when the strings weren’t UrlEncoded prior to transmission.

The service code is passed to the carrier, example service codes are:

|  |  |  |
| --- | --- | --- |
| CARRIERCODE | SERVICECODE | SERVICENAME |
| FEDEX | 01 | PRIORITY OVERNIGHT |
| FEDEX | 01 | INTERNATIONAL PRIORITY |
| FEDEX | 03 | 2DAY |
| FEDEX | 03 | INTERNATIONAL ECONOMY |
| FEDEX | 05 | STANDARD OVERNIGHT |
| FEDEX | 06 | FIRST OVERNIGHT |
| FEDEX | 06 | INTERNATIONAL FIRST |
| FEDEX | 20 | EXPRESS SAVER |
| FEDEX | 70 | INTERNATIONAL PRIORITY FREIGHT |
| FEDEX | 70 | FDX 1DAY FREIGHT |
| FEDEX | 80 | FDX 2DAY FREIGHT |
| FEDEX | 83 | FDX 3DAY FREIGHT |
| FEDEX | 86 | INTERNATIONAL ECONOMY FREIGHT |
| FEDEX | 90 | FDX GROUND RESIDENTIAL |
| FEDEX | 92 | FDX GROUND COMMERCIAL |
| UPS | 01 | UPS Next Day Air US Domestic |
| UPS | 02 | UPS Second Day Air US Domestic |
| UPS | 03 | UPS Ground US Domestic |
| UPS | 07 | UPS World Express - Originating in US |
| UPS | 08 | UPS Worldwide Expedited |
| UPS | 11 | UPS Standard |
| UPS | 12 | UPS 3 Day Select US Domestic |
| UPS | 13 | UPS Next Day Air Saver US Domestic |
| UPS | 14 | UPS Next Day Air Early AM US Domestic |
| UPS | 54 | UPS Worldwide Express PlusSM |
| UPS | 59 | UPS Second Day Air AM US Domestic |
| UPS | 65 | UPS Saver US Domestic |

Sample VB code:

Dim client As New SIIServiceClient()

Dim authorization As New UserName()

Dim order As New Order\_Header\_New

authorization.user\_name = "TEST"

authorization.user\_password = "1234"

order.ORDER\_TYPE = "OO"

order.CUSTOMER\_CODE = "802170"

order.CUSTOMER\_NAME = "BARNEY'S INC"

order.CUSTOMER\_ADDRESS\_1 = "P.O BOX 422"

order.CUSTOMER\_ADDRESS\_3 = "ATT: ACCOUNT PAYABLE"

order.CUSTOMER\_ADDRESS\_CITY = "LYNDHURST"

order.CUSTOMER\_ADDRESS\_STATE = "NJ"

order.CUSTOMER\_ADDRESS\_ZIP = "07071"

order.CUSTOMER\_ADDRESS\_COUNTRY = "US"

order.CARRIER = "FEDEX"

order.SERVICE = "02"

order.DELIVERY\_NAME = "Customer Name"

order.DELIVERY\_ADDRESS\_1 = "RECEIVING DEPARTMENT"

order.DELIVERY\_ADDRESS\_2 = "1201 Street Road"

order.DELIVERY\_ADDRESS\_CITY = "Philadelpia"

order.DELIVERY\_ADDRESS\_STATE = "NJ"

order.DELIVERY\_ADDRESS\_ZIP = "07022"

order.DELIVERY\_ADDRESS\_COUNTRY = "US"

order.DELIVERY\_TELEPHONE = "2015551212"

Dim order\_detail As New Order\_Detail\_New

Dim order\_line\_items() As Order\_Detail\_New

ReDim order\_line\_items(1)

order\_detail.SKU = "9999000159072"

order\_detail.QUANTITY = 2

order\_detail.PRICE = 235

order\_line\_items(0) = order\_detail

order\_detail = New Order\_Detail\_New

order\_detail.SKU = "9999000159089"

order\_detail.QUANTITY = 4

order\_detail.PRICE = 240

order\_line\_items(1) = order\_detail

order.Order\_Details = order\_line\_items

Dim order\_response As NewOrderRequestResponse

order\_response = client.new\_order\_entry(authorization, order)

## Inventory Request

This returns the current on hand inventory quantity less any orders that have been received but not picked. It currently does not return any products that have zero on hand inventory. Please note that the array of inventory values has an unknown limit, but less than 100.

Input message:

User name

Requested UPC’s

|  |  |  |
| --- | --- | --- |
| Upc | String(20) | Upc/ean13/sku |
| Warehouse | String(4) | Stellae warehouse code |

If UPC is blank returns all inventory for the warehouse. If warehouse is blank, returns all inventory for the UPC regardless of the warehouse.

Output message

|  |  |  |
| --- | --- | --- |
| Status | String |  |
| Inventory\_values | array |  |

Inventory values

|  |  |  |
| --- | --- | --- |
| UPC | String(18) |  |
| Warehouse | String(4) |  |
| On\_hand | decimal |  |
| Uom | String | Unit of measure |
| Lot | String(50) | If serialized/lots |

Sample VB code:

Dim client As New SIIServiceClient()

Dim authorization As New UserName()

Dim upcs() As UPC\_Inventory\_Request

ReDim upcs(1)

Dim upc As New UPC\_Inventory\_Request

upc.warehouse = "NYC"

upc.upc = "9999000014319"

upcs(0) = upc

upc = New UPC\_Inventory\_Request

upc.warehouse = "NYC"

upc.upc = "9999000091655"

upcs(1) = upc

authorization.user\_name = "TEST"

authorization.user\_password = "1234"

Dim response = Client.get\_inventory\_on\_hand(authorization, upcs)

For Each oh As UPC\_Inventory\_Response In response.Inventory\_values

' do something

Next

# Shipment Information

This supplies information about shipment and returns. Please note, due to the heavy processing required by this request, we limit to a maximum of 3 days if you don’t specify an order number.

Input message:

Username

Requested orders

|  |  |  |
| --- | --- | --- |
| Order Number | String |  |
| Invoice date from | Date |  |
| Invoice date to | Date |  |

Output message

|  |  |  |
| --- | --- | --- |
| Status |  |  |
| Orders |  |  |

Orders:

|  |  |  |
| --- | --- | --- |
| Order\_id | String |  |
| Invoice\_date | Date |  |
| Invoice\_number | String |  |
| Invoice\_type | String | (IN – shipment, CM return) |
| Cartons | array |  |

Cartons

|  |  |  |
| --- | --- | --- |
| CARRIER | String |  |
| SERVICE | String |  |
| TRACKING | String | Outbound tracking number |
| RETURN\_TRACKING | String | Return label tracking number |
| UCC128 | String |  |
| Items | array |  |

Items

|  |  |  |
| --- | --- | --- |
| SKU | String |  |
| Size | String | Not used |
| Quantity | Decimal |  |
| Price | Decimal |  |

Example code:

Dim shipments As New Shipment\_Request

Dim shipmentresponse As NewShipmentRequestResponse

shipments.date\_from = #6/21/2011#

shipmentresponse = client.get\_shipment\_information(authorization, shipments)

# Catalog information

This supplies information about items in the database.

Input message:

Username

Requested styles:

|  |  |  |
| --- | --- | --- |
| Season\_code | String |  |
| SKU | string |  |
| Style |  |  |

Output message:

Status (0001 is success)

Upcs()

Returns:

|  |  |  |
| --- | --- | --- |
| SKU | String |  |
| Description | String |  |
| Style | String |  |
| Color\_code | String |  |
| Color | String |  |
| Size\_code | String |  |
| Season\_code | String |  |
| Fabric | string |  |

Sample vb code:

Dim catalog As New Catalog\_items\_request

Dim items As Catalog\_items\_response

catalog.style = "TD\*"

items = client.get\_catalog\_information(authorization, catalog)

# Import Line List

This allows the import of new items to the line list:

Input message:

Username:

Array of items to be imported – note that data will generally be translated to uppercase and truncated to the maximum allowed length. The upc can be a non numeric SKU, please discuss with Stellae.

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | MAX SIZE | \*=required | EXAMPLE DATA |
| upc | 18 | string\* | 3262113300102 |
| Description | 40 | string\* | TOOTHBRUSHES |
| Content | 60 | string | (HBR) 100 PLASTIC |
| Style | 20 | string\* | CONA-TOOT-HBR |
| Color Code | 18 | string\* | 001 |
| Color Name | 30 | string | WHITE |
| size code | 18 | string\* | S |
| size name | 30 | string | SMALL |
| size order |  | Number | 3 |
| Activity code | 18 | string | CONA |
| Activity Name | 50 | string | SLG |
| Country of Origin | 30 | string | JAPAN |
| Season Code | 4 | string | 17 |
| Alternate UPC | 30 | string | CONA-TOOT-HBR/001/S |
| RTL Price |  | Number | 6 |
| Price Group | 20 | string | LIST |
| WHLS Price |  | Number | 3 |
| Cost |  | Number | 0.8 |
| Currency\_code | 3 | string | USD |
| Fabric\_code | 18 | string | HBR |
| Image\_url |  | string |  |

Returns: 0001 for success

Example code:

Dim ll(0) As Line\_list\_row

ll(0) = New Line\_list\_row

With ll(0)

.upc = "1234567890123"

.description = "Sample Product"

.content = "ghi"

.style = "ghi"

.color\_code = "001"

.color = "white"

.size\_code = "S"

.size = "Small"

.size\_order = 1

.activity\_code = "ghi"

.country\_of\_origin = "Italy"

.season\_code = "12SS"

.alternate\_code = "ghi"

.retail\_price = 3

.price\_group = "LIST"

.wholesale\_price = 2

.cost = 1

.currency\_code = "USD"

.activity = "ghi"

.fabric\_code = "ghi"

End With

Dim response\_line\_list = client.import\_line\_list(authorization, ll)

Console.WriteLine("response " & response\_line\_list.status)

# Advance Ship notice

This message provides the needed information to receive goods into inventory. This should be sent instead of a purchase order.

Input message:

User name

Order Information

Returns:

Status

Stallae Purchase Order transaction number

|  |  |  |
| --- | --- | --- |
| poref1 | String | normally invoice number |
| poref2 | String | normally purchase order number |
| vendorname | String | vendor name |
| vendorcode | String | vendor code |
| invoice\_date | Date | invoice date |
| ship\_date | Date | ship date |
| asn\_lines() | Asn\_Line | items |

Asn\_lines:

|  |  |  |
| --- | --- | --- |
| cartonno | String | carton number |
| cartonref | String | carton reference |
| sku | String | upc/sku |
| style | String | style |
| color\_code | String | color |
| size\_code | String | size |
| quantity | Decimal | quantity expected |
| cost | Decimal | unit ocst |
| composition | String | composition |
| country\_origin | String | country of origin |
| description | String | description |
| comment | String | comment |
| season | String | season |
| sales\_order | String | sales order number |
| customer\_code | String | customer code |
| store\_code | String | store/door code |

The style, color\_code, size\_code are primarily for verification, and composition, country of origin, description, season, are if the sku doesn’t exist in the database, an entry in the product table can be created. However, at the current time, the sku must exist in the product table, so these fields are not used.

Dim ahn As Asn\_Header\_New

ahn = New Asn\_Header\_New

ahn.poref1 = "Test1"

ahn.poref2 = "Test2"

ahn.invoice\_date = Now.Date

ahn.ship\_date = Now.Date

ahn.vendorcode = "V00001"

ahn.vendorname = "TEST VENDOR"

Dim asn\_line As Asn\_Line

Dim asn\_lines(2) As Asn\_Line

asn\_line = New Asn\_Line

asn\_line.cartonno = 1

asn\_line.cartonref = "123456"

asn\_line.sku = "3585810017723"

asn\_line.color\_code = "-"

asn\_line.size\_code = "50E"

asn\_line.style = "VROS50E"

asn\_line.quantity = 1

asn\_line.sales\_order = "1234"

asn\_line.cost = 20

asn\_line.comment = "test"

asn\_line.customer\_code = "BREWINGMAR"

asn\_lines(0) = asn\_line

asn\_line = New Asn\_Line

asn\_line.cartonno = 1

asn\_line.cartonref = "123456"

asn\_line.sku = "3585810017723"

asn\_line.color\_code = "-"

asn\_line.size\_code = "50E"

asn\_line.style = "VROS50E"

asn\_line.quantity = 4

asn\_line.sales\_order = "4321"

asn\_line.cost = 20

asn\_line.comment = "test"

asn\_line.customer\_code = "100270"

asn\_lines(1) = asn\_line

asn\_line = New Asn\_Line

asn\_line.cartonno = 1

asn\_line.cartonref = "123456"

asn\_line.sku = "3585810017860"

asn\_line.color\_code = "-"

asn\_line.size\_code = "50E"

asn\_line.style = "VROS50E"

asn\_line.quantity = 4

asn\_line.sales\_order = "4321"

asn\_line.cost = 20

asn\_line.comment = "test"

asn\_line.customer\_code = "100270"

asn\_lines(2) = asn\_line

ahn.asn\_lines = asn\_lines

order\_response = client.new\_asn\_entry(authorization, ahn)

# Purchase Order Receipts

This is to find out what has been received.

Input message:

Username

Receipts:

|  |  |  |
| --- | --- | --- |
| entry\_date\_from | Date | receipt date start |
| entry\_date\_to | Date | receipt date end |
| po\_ref1 | String | normally invoice number |
| po\_ref2 | String | normally purchase order number |
| po\_xact | Integer | internal stellae purchase order transaction number |

This returns rows (not hierarchical), output message:

Status

Receipt rows

Where each row contains:

|  |  |  |
| --- | --- | --- |
| poxact | Integer | stellae purchase order transaction number |
| porxact | Integer | stellae purchase receipt transaction number |
| polinenum | Integer | stellae purchase order line number |
| po\_entry\_date | Date | transaction date |
| po\_receipt\_date | Date | entered receipt date |
| warehouse | String | warehouse |
| qty\_rcvd | Decimal | qty received |
| ext\_amount | Decimal | extended amount |
| sku | String | sku/upc |
| po\_ordered | Decimal | asn quantity |
| vendor | String | vendor name |
| poref1 | String | normally invoice number from asn (poref1 on asn file) |
| poref2 | String | normally purchase order number from asn (poref2 on asn file) |
| pocomment | String | comment |
| style | String | sku style |
| color\_code | String | sku color |
| size\_code | String | sku size |
| lot\_number | String | lot number |

Dim poreceipts As purchase\_Order\_receipt\_request

poreceipts = New purchase\_Order\_receipt\_request

Dim por\_response As Purchase\_Order\_Receipt\_Response

authorization.user\_name = "xyz"

authorization.user\_password = "pdq"

poreceipts.po\_xact = 543

CType(client.Endpoint.Binding, WSHttpBinding).MaxReceivedMessageSize = 1000000

por\_response = client.get\_purchase\_order\_receipt(authorization, poreceipts)

Console.WriteLine("response {0}", por\_response.status)

# Error codes

|  |  |
| --- | --- |
| 0001 | success |
| 0002 | Bad login information |
| 0003 | Login points to multiple databases |
|  |  |
| 0005 | Invalid order type must be OO or CM |
| 0006 | Error creating order header |
| 0007 | Bad UPC/EAN13 |
| 0008 | Duplicate order – order number exists in database |
| 0009 | Problem with line item import |