# Table of Contents

1 Introduction .............................................................................................................................. 4  
1.1 Prerequisites .......................................................................................................................... 4  
1.2 WSDL and Schema.................................................................................................................. 4  
1.3 Web Service Request .............................................................................................................. 4  
1.4 Web Service Response ........................................................................................................... 5  
1.5 Web Service Criteria ................................................................................................................ 6  
1.5.1 Sample Criteria ............................................................................................................... 8  
1.6 Security .................................................................................................................................... 8  

2 Types ........................................................................................................................................ 9  
2.1 Data Types ............................................................................................................................... 9  
2.2 Occam Types ........................................................................................................................... 9  
2.3 Lookup Types .......................................................................................................................... 12  
2.3.1 Purchase order lookup types ............................................................................................ 13  
2.3.2 Receipt lookup types .......................................................................................................... 16  
2.3.3 Inventory transaction lookup types .................................................................................. 19  
2.3.4 Sales order lookup types .................................................................................................... 23  
2.3.5 Shipment lookup types ....................................................................................................... 26  
2.3.6 Wave fulfillment lookup types .......................................................................................... 30  
2.3.7 Reporting lookup types ...................................................................................................... 35  
2.3.8 Administrative lookup types .............................................................................................. 39  
2.4 Flex Field Types ..................................................................................................................... 42  

3 Operations.............................................................................................................................. 43  
3.1 authenticate ............................................................................................................................ 43  
3.2 authenticateAndGetTenantName .......................................................................................... 44  

3.3 Purchase Order Operations ................................................................................................... 44  
3.3.1 savePurchaseOrder .......................................................................................................... 44  
3.3.2 lookUpPurchaseOrders ................................................................................................... 50  
3.3.3 cancelPurchaseOrder ........................................................................................................ 51  
3.3.4 overwritePurchaseOrder ................................................................................................. 52  

3.4 Receipt Operations ................................................................................................................ 54  
3.4.1 getReceipts ..................................................................................................................... 54  
3.4.2 lookupReceipts ................................................................................................................. 54  

3.5 Inventory Operations ............................................................................................................. 55  
3.5.1 lookUpAdjustments .......................................................................................................... 55  
3.5.2 lookUpCycleCounts ......................................................................................................... 56  
3.5.3 lookUpBinTransfer ............................................................................................................ 57  
3.5.4 createStorageDocuments ............................................................................................... 57  
3.5.5 lookUpTransferShipments .............................................................................................. 58
3.5.6 lookUpTransferReceipts ................................................................. 59

3.6 Sales Order Operations ........................................................................................................
3.6.1 saveSalesOrder .............................................................................................................. 60
3.6.2 lookUpSalesOrders ......................................................................................................... 65
3.6.3 cancelSalesOrder .......................................................................................................... 66
3.6.4 overwriteSalesOrder ..................................................................................................... 67

3.7 Shipment Operations ........................................................................................................
3.7.1 getShipments ............................................................................................................... 69
3.7.2 lookupShipments .......................................................................................................... 69
3.7.3 updateParcelPostInfo ................................................................................................... 70

3.8 Wave Fulfillment Operations ............................................................................................
3.8.1 lookupPickTickets ......................................................................................................... 71
3.8.2 lookUpBOLs ................................................................................................................ 72
3.8.3 closeDocuments ............................................................................................................ 73
3.8.4 shipClosedBOLPackages .......................................................................................... 74
3.8.5 commitDocuments ........................................................................................................ 75
3.8.6 consumeDocuments .................................................................................................... 77
3.8.7 updateBOLPkgAttributes ......................................................................................... 78
3.8.8 setBOLCustomFields ............................................................................................... 79

3.9 Reporting Operations ......................................................................................................
3.9.1 lookUpInventory ......................................................................................................... 81
3.9.2 lookUpBillingItems ..................................................................................................... 82
3.9.3 lookUpItemMasterAtp .............................................................................................. 83
3.9.4 lookUpVoidedItems .................................................................................................... 84
3.9.5 lookUpActivityLines ................................................................................................. 84

3.10 Administrative Operations ............................................................................................
3.10.1 saveItemMaster .......................................................................................................... 85
3.10.2 lookUpItemMaster ..................................................................................................... 89
3.10.3 lookUpBusinessAffiliates ...................................................................................... 90
3.10.4 saveBusinessAffiliates .......................................................................................... 90
3.10.5 lookUpLocation ........................................................................................................ 94
1 Introduction
SmartTurn web services allow clients (web services invokers) to share data with SmartTurn through web service operations composed of requests and responses. This document describes the web services API and how to use it.

For an introduction to using web services with SmartTurn, see SmartTurn Web Services - Getting Started.doc.

1.1 Prerequisites
Before you start using SmartTurn web services, you must:

- Have a valid SmartTurn user account. Consult your SmartTurn administrator to set up an account. Consider creating a separate user account just for web services access.
- Have your SmartTurn tenant configured (warehouses, locations, and item masters set up in SmartTurn). Consult your SmartTurn administrator or the SmartTurn online Help for information on configuring your tenant information.
- Have a SOAP engine, such as AXIS, in which you can create and execute operations in a SOAP environment. See the section Example Development Environment, in SmartTurn Web Services - Getting Started.doc, for an example of this.

Important Note: SmartTurn does not provide support for configuring a SOAP environment or for additional development based on web services requests or responses (including configuring the requests or responses as flat files, objects, or other data structures).

1.2 WSDL and Schema
The SmartTurn web service is deployed using the document literal type. The SmartTurn web services are WS-I compliant.

The SmartTurn web service WSDL is located at: https://services.smartturn.com/occam/services/OccamService?wsdl

The referenced SmartTurn schema files are located at: https://services.smartturn.com/occam/schema

1.3 Web Service Request
Each SmartTurn web service operation is composed of a request and a response.

Each request begins with the standard SOAP envelope declaration:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

After the envelope declaration, the SOAP body begins. The SOAP body consists of the web service request operation. The web service operation should first include your credential information, and followed by any elements that are required as part of the request:

```xml
<savePurchaseOrder
  xmlns="http://www.smartturn.com/services/OccamService/purchase-order"
  inCredential>
```
Your credential information includes a username and password. This information must be a valid login to the SmartTurn system as accessed from your browser. Talk to your SmartTurn administrator about setting up a SmartTurn account specifically for web services access.

### 1.4 Web Service Response

Each SmartTurn web service operation is composed of a request and a response. Each response begins with the standard SOAP envelope declaration:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

After the envelope declaration, the SOAP body begins. The SOAP body consists of the web service operation response. In SmartTurn, the response typically includes a single element of a type specific to the operation (for example: `UploadResponse` for saving purchase orders, sales orders, or item masters; `*Result` for a lookup result):

```xml
<soapenv:Body>
<saveSalesOrderResponse xmlns="http://www.smartturn.com/services/OccamService/sales-order">
<uploadResponse>
```

Refer to [Section 3, Operations](#), for specific output message information for each operation.

**Errors.** If there is an error performing the request operation, the response will include an error message. You can use the error message to determine the nature of the problem with the
request. You should also check the validity of the request against the schema. For example, field length limits set by the schema but violated in the request will produce an error in the response.

**Timeouts.** Some tools for creating and executing web services requests have a built-in session timeout. If you are extracting a large amount of data, make sure your timeout is set to at least 2 minutes.

### 1.5 Web Service Criteria

For each SmartTurn web service request that is a lookup operation, you can choose to look up all of the records of that entity in SmartTurn, or create a filter and look up a subset of those records. The SmartTurn web service includes a simple criteria definition that can be used to build filters for this purpose.

A **Criteria** is a collection of one or more **Criterion**s, as well as information about how to sort and paginate the results of the filter specified by the **Criterion**s.

<table>
<thead>
<tr>
<th>Criteria Elements (type: name)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| Criterion: criterions OR LookUpManager: manager | For document lookups, supply the Criterion. By default, the lookup will assume the default warehouse of the user specified in the inCredential element (see Section 1.3 Web Service Request). To perform a lookup for a document in a different warehouse that the user has access to, the Criterion must contain the propertyName of warehouseName, the operator equal, and either a valid warehouse name or the string “All warehouses”.

For lookups that have managers in the desktop user interface (lookUpBillingItems, lookUpInventory, or lookUpActivityLines operations), supply either the Criterion or the LookUpManager. If the LookUpManager is supplied, the Criterion if supplied can only contain the propertyName of warehouseName, the operator equal, and either a valid warehouse name or the string “All warehouses”. This allows you to use existing managers but confine the results to one warehouse.

See table below for description of Criterion elements.

The LookUpManager contains one element:

- **managerName** – the case-sensitive name of an existing billing line manager, inventory manager, or activity line manager in SmartTurn.

| PaginationInfo: paginationInfo | If there are a large number of results, the results are paginated into smaller groups when they are retrieved (web services) or displayed (in list views in the SmartTurn user interface). This improves SmartTurn system response time.

In the SmartTurn web services, you can use the paginationInfo element to specify how many records to return per page and which page to retrieve.

The paginationInfo element contains:

- **pageNumber** – the page number from which the
records need to be retrieved. Must be a valid number within the total result pages available. Defaults to 1.

- `recordsPerPage` – the number of records to return per page. Must be <= 1000. Defaults to 1000. **Note:** for the `lookUpPurchaseOrders` and `lookUpSalesOrders` operations only, the `recordsPerPage` must be <= 100.

**SortingInfo:**

- `sortByProperty` – the field by which the records should be sorted. Must be the name of a valid field within the entity schema definition that is currently being queried.
- `sortType` – either ASCENDING or DESCENDING.

---

Each **Criterion** element itself has several required elements.

<table>
<thead>
<tr>
<th>Criterion Elements (type: name)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>string:propertyName</code></td>
<td><strong>Required.</strong> The name of property from the appropriate schema type. For example, if you are looking up receipts, one of the fields is id: <code>&lt;tns:propertyName&gt;id&lt;/tns:propertyName&gt;</code></td>
</tr>
<tr>
<td><code>CriterionOperatorType:operator</code></td>
<td><strong>Required.</strong> One of: equal, notEqual, lessThan, lessOrEqual, greaterThan, greaterOrEqual, in, notIn, null, notNull.</td>
</tr>
<tr>
<td><code>conditional value</code></td>
<td><strong>Required.</strong> For the conditional value, select one of <code>stringValue</code>, <code>doubleValue</code>, <code>intValue</code>, <code>dateValue</code>, <code>booleanValue</code>, <code>arrayValue</code>. For the value of the element, enter the filter information, based on the <code>propertyName</code> field. For example, if you are looking up receipts, one of the fields is id: <code>&lt;tns:propertyName&gt;id&lt;/tns:propertyName&gt;</code>. Because <code>id</code> is a string field, use <code>stringValue</code> and enter the string to filter for: <code>&lt;tns:stringValue&gt;RCPT-000008&lt;/tns:stringValue&gt;</code> Note that if the <code>operator</code> is null or notNull, this element is ignored.</td>
</tr>
<tr>
<td><code>negate</code></td>
<td><strong>Optional.</strong> Boolean value indicating whether to negate the operator. Defaults to false (do not negate).</td>
</tr>
</tbody>
</table>

When a **Criteria** contains multiple **Criterion**s, the **Criterion**s are ANDed together.
1.5.1 Sample Criteria

Example 1: This is a criteria that can be used to pull a particular receipt from the SmartTurn system.

```xml
<s0:inCriteria>
  <s0:Criterion>
    <tns:propertyName>id</tns:propertyName>
    <tns:operator>equals</tns:operator>
    <tns:stringValue>RCPT-000008</tns:stringValue>
  </s0:Criterion>
</s0:inCriteria>
```

Example 2: This is a more involved criteria that can be used to retrieve the first several sales orders from the SmartTurn system, sorted by Customer. Note that in this case the paginationInfo is not strictly necessary, since based on the Criterion only 9 records will be returned.

```xml
<s0:inCriteria>
  <s0:Criterion>
    <tns:propertyName>id</tns:propertyName>
    <tns:operator>equals</tns:operator>
    <tns:stringValue>SO-000010</tns:stringValue>
  </s0:Criterion>
  <s0:paginationInfo>
    <tns:pageNumber>1</tns:pageNumber>
    <tns:recordsPerPage>9</tns:recordsPerPage>
  </s0:paginationInfo>
  <s0:sortingInfo>
    <tns:sortByProperty>customerName</tns:sortByProperty>
    <tns:sortType>DESCENDING</tns:sortType>
  </s0:sortingInfo>
</s0:inCriteria>
```

1.6 Security

Each web service operation requires the SmartTurn Access Web Services permission.

Additionally, each web service operation requires one or more permissions in order to successfully execute, depending on the data being accessed or the operation being performed.

The user login performing the web service operation must have both the Access Web Services permission and the specific permission(s) needed for the operation.

In the SmartTurn user interface, permissions are assigned to roles and each user login has one or more roles within the system. Work with your SmartTurn administrator to ensure that the user login performing the web service operations has a role with the appropriate permissions enabled.
2 Types

Each SmartTurn entity used in a web service operation has its own type. For example, there is a Receipt type for SmartTurn receipts, a PurchaseOrderItem type for SmartTurn purchase order items, etc. More information about particular entity-specific types is discussed in the relevant part of Section 3, Operations, below. This section details “building block” types and types that are common across multiple entities.

The complete type definitions for all types are in the schema files available at https://services.smartturn.com/occam/schema.

2.1 Data Types

Data types are used throughout SmartTurn web services as building blocks for other types.

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Base Type and Value Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>String150Type</td>
<td>string, max length: 150</td>
</tr>
<tr>
<td>NameType</td>
<td>string, max length: 50</td>
</tr>
<tr>
<td>CommentType</td>
<td>string, max length: 200</td>
</tr>
<tr>
<td>DateTimeType</td>
<td>dateTime</td>
</tr>
<tr>
<td>ContactType</td>
<td>string, max length: 50</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>string, max length: 50</td>
</tr>
<tr>
<td>NumberType</td>
<td>string, max length: 50</td>
</tr>
<tr>
<td>IdType</td>
<td>string, max length: 50</td>
</tr>
<tr>
<td>DescriptionType</td>
<td>string, max length: 255</td>
</tr>
<tr>
<td>VendorItemIdType</td>
<td>string, max length: 50</td>
</tr>
<tr>
<td>ManufacturerItemIdType</td>
<td>string, max length: 100</td>
</tr>
<tr>
<td>UpcCodeType</td>
<td>string, max length: 100</td>
</tr>
<tr>
<td>EanCodeType</td>
<td>string, max length: 20</td>
</tr>
<tr>
<td>ManufacturerIdType</td>
<td>string, max length: 100</td>
</tr>
<tr>
<td>CustomFieldType</td>
<td>string, max length: 50</td>
</tr>
<tr>
<td>GeneralLedgerCodeType</td>
<td>string, max length: 30</td>
</tr>
<tr>
<td>PositiveDouble</td>
<td>double, minimum value: 1 (rounded at 12 digits, as in most of the desktop interface)</td>
</tr>
<tr>
<td>PositiveInteger</td>
<td>int, minimum value: 1</td>
</tr>
<tr>
<td>RecordsPerPageIntegerType</td>
<td>int, range: 1-1000, inclusive</td>
</tr>
<tr>
<td>CustomFieldStringType</td>
<td>string, max length: 100</td>
</tr>
<tr>
<td>CustomFieldMemoType</td>
<td>string, max length: 1000</td>
</tr>
</tbody>
</table>

2.2 Occam Types

Occam types are types specific to the SmartTurn business processes (as opposed to data types, above, which are more general). Like data types, occam types are used throughout SmartTurn web services as building blocks for other types.
<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>double: value string: unitAbbreviation</td>
</tr>
<tr>
<td>Address</td>
<td>string: addressLine1 string: addressLine2 string: city string: state string: country string: postalCode</td>
</tr>
<tr>
<td>Warehouse</td>
<td>NameType: localName</td>
</tr>
<tr>
<td>OccamPasswordCredential</td>
<td>conditional value: string: Password or string: notUsed; notUsed is available for internal use only. Known Issue: Because the OccamPasswordCredential uses an XML choice element here (choice of the Password and notUsed fields), some environments may require this parameter to be specified generically rather than by field name. For example in some .NET environments, use inCredential = new OccamPasswordCredential (UserID = ST_User(), Item = ST_Pass()); instead of inCredential = new OccamPasswordCredential (UserID = ST_User(), Password = ST_Pass()); string: permission (up to 100 permission elements can be specified, using the identifier (not the UI display name) of a permission in SmartTurn, to verify if this user has this permission; primarily for internal use.</td>
</tr>
<tr>
<td>CriterionOperatorType</td>
<td>string (enumeration): equal, notEqual, lessThan, lessOrEqual, greaterThan, greaterOrEqual, in, notIn, null, notNull</td>
</tr>
<tr>
<td>Criterion</td>
<td>boolean: negate string: propertyName CriterionOperatorType: operator conditional value: string: stringValue, double: doubleValue, int: intValue, dateTime: dateValue, arrayValueType: arrayValue (array of up to 100 string, double, or int; only for use with 'in' operator)</td>
</tr>
<tr>
<td>arrayValueType</td>
<td>string: stringValue double: doubleValue int: intValue</td>
</tr>
<tr>
<td>Criteria</td>
<td>Criterion: criterions (can occur 0 or more times) LookUpManager: manager (for lookUpInventory, lookUpBillingItems, or lookUpActivityLines, you can use this instead of the criterions) PaginationInfo: paginationInfo SortingInfo: sortingInfo</td>
</tr>
<tr>
<td>Type Name</td>
<td>Components (type: name)</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>On-Demand WMS (SmartTurn) Web Services - Web Services API</td>
<td><em><em>See the particular lookup</em> operations in Section 3, Operations, for more information on using Criteria.</em>*</td>
</tr>
<tr>
<td>InventoryConditionEnum</td>
<td>string (enumeration): EXCELLENT, GOOD, FAIR, POOR, DAMAGED, UNKNOWN</td>
</tr>
<tr>
<td>ShipmentStateEnum</td>
<td>string (enumeration): NEW, SAVED, APPROVED, IN_PROGRESS, CLOSED, SHIPPED (deprecated in favor of CLOSED), COMMITTED, CONSUMED, CANCELLED</td>
</tr>
</tbody>
</table>
| SaveStateEnum                | string (enumeration): NEW (deprecated – no documents will be returned with this status), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED, DONE (deprecated in favor of CLOSED)  
($SaveStateEnum is deprecated – use ShipmentStateEnum instead)                                                                                      |
| Currency                     | double: value                                                                                                                                                    |
| CreateRecord                 | string: createdBy                                                                                                                                              |
| LastModifiedRecord           | string: modifiedBy                                                                                                                                             |
| UploadResponse               | string: externalRefId                                                                                                                                             |
| ActionStatusType             | string: error                                                                                                                                                    |
|                               | string: warning                                                                                                                                                |
|                               | string: systemId                                                                                                                                                |
| ActionStatusType             | string (enumeration): SUCCESS or FAILURE                                                                                                                          |
| CurrencyTypeEnum             | string (enumeration): $                                                                                                                                                                                                        |
| InboundItemStatusEnum        | string (enumeration): NEW, SAVED, APPROVED, READY, IN_PROGRESS, ALLOCATED, SERVICE_STARTED, CLOSED, COMMITTED, CONSUMED, CANCELLED                                                                                   |
| OrderTypeEnum                | string (enumeration): INTERNAL, EXTERNAL                                                                                                                          |
| Affiliate                    | NameType: companyName                                                                                                                                           |
|                               | NameType: companyId                                                                                                                                                                                               |
| LogisticsOperator            | NameType: localname                                                                                                                                                                                                     |
| LocationName                 | NameType: name                                                                                                                                                   |
| WSDocumentIdentification     | IdType: number                                                                                                                                                    |
|                               | IdType: externalNumber                                                                                                                                             |
|                               | string: warehouseName                                                                                                                                                                                                 |
| WSDocumentIdentificationWithType | WSDocumentTypeEnum: documentType  
IdType: number  
IdType: externalNumber  
string: warehouseName                                                                                   |
### 2.3 Lookup Types

SmartTurn documents are composed of header information (information about the document itself, such as the due date, the person who created it, the document identification number, etc.), and a set of items.

Lookup types are used for the elements in the web service responses for the lookup* operations. They provide the return values for the lookup* requests. This section describes both the document and item level lookup information.

When you create a lookup request, you specify criteria for what you want to look up. See Section 1.5 Web Service Criteria.
When you receive the response to a lookup request, in addition to the document and line item information, every lookup result contains some metadata about the result being returned. That metadata is made up of some or all of the following elements:

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultMetadata</td>
<td>int: totalRecordsCount – the number of records retrieved by the lookup.</td>
</tr>
<tr>
<td></td>
<td>int: recordCountInThisPage – the number of records in the current page of lookup results.</td>
</tr>
<tr>
<td></td>
<td>int: totalPages – the number of pages of records retrieved by the lookup.</td>
</tr>
</tbody>
</table>

### 2.3.1 Purchase order lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSPurchaseOrderLookupResult</td>
<td>ResultMetadata: resultMetadata</td>
</tr>
<tr>
<td></td>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
</tr>
<tr>
<td></td>
<td>PurchaseOrder: purchaseOrders – can occur 0 or more times (up to 100); the information about each purchase order itself. Each PurchaseOrder contains (references are to Ordering&gt;Purchase Orders&gt;View Purchase Order #: PO-nnnn):</td>
</tr>
<tr>
<td></td>
<td>• NumberType: externalNumber (External #)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: date (PO date)</td>
</tr>
<tr>
<td></td>
<td>• Warehouse: warehouse (Warehouse; required only if the warehouse is different from the default warehouse for this login)</td>
</tr>
<tr>
<td></td>
<td>• string: warehouseName (Warehouse)</td>
</tr>
<tr>
<td></td>
<td>• NumberType: number (PO #)</td>
</tr>
<tr>
<td></td>
<td>• int: changeOrderNumber (unused)</td>
</tr>
<tr>
<td></td>
<td>• NameType: vendorName (Vendor&gt;Vendor name)</td>
</tr>
<tr>
<td></td>
<td>• ContactType: vendorContact (Vendor&gt;Vendor contact)</td>
</tr>
<tr>
<td></td>
<td>• PhoneType: vendorContactPhone (Vendor&gt;Vendor phone)</td>
</tr>
<tr>
<td></td>
<td>• Address: vendorAddress (Vendor&gt;Address line 1, Address line 2, City, State, Country, Postal code)</td>
</tr>
<tr>
<td></td>
<td>• NameType: buyerName (Ship To&gt;Company)</td>
</tr>
<tr>
<td></td>
<td>• ContactType: buyerContact (Ship To&gt;Ship to contact)</td>
</tr>
<tr>
<td></td>
<td>• Address: shipTo (Ship To&gt;Address line 1, Address line 2, City, State, Country, Postal code)</td>
</tr>
<tr>
<td></td>
<td>• PurchaseOrderStatusEnum: status (Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)</td>
</tr>
<tr>
<td></td>
<td>• WSSectionStatusEnum: billingStatus (Billing&gt;Billin</td>
</tr>
</tbody>
</table>
Status: only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED

- Affiliate: vendor (contains Vendor>Vendor and Code from Vendor's business affiliate record)
- CreateRecord: createRecord (deprecated in favor of createdBy and createdDate)
- string: createdBy (Created by; only populated on lookup)
- dateTime: createdDate (Date created; only populated on lookup)
- LastModifiedRecord: lastModifiedRecord (deprecated in favor of lastModifiedBy and lastModifiedDate)
- string: lastModifiedBy (not displayed in SmartTurn user interface; only populated on lookup)
- dateTime: lastModifiedDate (not displayed in SmartTurn user interface; only populated on lookup)
- CommentType: comments (Comments)
- Currency: orderedTotalAmount (Total amount; only populated on lookup)
- PurchaseOrderItem: item (from the PO Items table; required; 1 or more)
- OrderTypeEnum: type (Source; only populated on lookup)
- IdType: departmentNumber (Dept #)
- IdType: groupNumber (Group #)
- boolean: partialReceiptsAllowed (Partial Receipt Allowed)
- Affiliate: owner (deprecated)
- Affiliate: carrier (deprecated)
- Affiliate: broker (deprecated)
- dateTime: dueDate (Due Date)
- NameType: orderedBy (Ordered by)
- string: priority (Priority)
- NameType: vendorCode (Code from the Vendor's business affiliate record)
- string: ownerName (Owner)
- NameType: ownerCode (Code from the Owner's business affiliate record)
- string: carrierName (Carrier)
- NameType: carrierCode (Code from the Carrier's business affiliate record)
- string: brokerName (Broker)
- NameType: brokerCode (Code from the Broker's business affiliate record)
- CustomFields: poCustomFields (fields from the Custom tab; see Section 2.4, Flex Field Types)
### PurchaseOrderItem

- **int**: `lineNumber` *(Line #)*
- **DescriptionType**: `description` *(Description)*
- **DetailType**: `details` *(Details)*
- **dateTime**: `dueDate` *(Due Date, from the purchase order header information)*
- **Currency**: `price` *(Unit Cost)*
  - It is strongly recommended that `orderedTotalAmount`, which reports the total inventory value of the line, is used rather than `price`, due to details of implementation and how these values are persisted.
- **Currency**: `orderedTotalAmount` *(Total Item Cost)*
- **NameType**: `group` *(Group)*
- **CommentType**: `contractInfo` *(not displayed in SmartTurn user interface)*
- **Quantity**: `orderedQuantity` *(Ordered Qty)*
- **Quantity**: `pendingQuantity` *(deprecated)*
- **Quantity**: `receivedQuantity` *(Received qty)*
- **IdType**: `itemMasterId` *(Item #)*
- **InboundItemStatusEnum**: `status` *(Item Status)*
- **CommentType**: `receivingInstructions` *(not displayed in SmartTurn user interface)*
- **VendorItemIdType**: `vendorItemId` *(Product ID)*
- **DescriptionType**: `manufacturer` *(Manufacturer)*
- **ManufacturerItemIdType**: `manufacturerItemId` *(Manufacturer Item ID)*
- **UpcCodeType**: `upcCode` *(UPC)*
- **ManufacturerIdType**: `manufacturerId` *(Manufacturer ID)*
- **NumberType**: `lotNumber` *(Lot #)*
- **IdType**: `externalRefNumber` *(External #, from the purchase order)*
- **Quantity**: `weightPerUoM` *(Dim Wt per UoM: the dim. weight of this line item per unit of measure. For example, if the Dim Wt is 10 lb and the line item is for 4 cs, then the weightPerUom is 2.5 lb)*
- **Quantity**: `weightTotal` *(Dim Wt: the total dim. weight of this line item as entered by the user)*
- **Quantity**: `cubePerUom` *(Cube per UoM: the cube or volume of this line item, per unit of measure)*
- **Quantity**: `cubeTotal` *(Cube: the total cube or volume of this line item as entered by the user)*
- **integer**: `pallets` *(Pallets: the total number of separate pallets for this line item, as entered by the user)*
- **PositiveDouble**: `palletsFilled` *(Pallets Filled: the total number of pallets filled for this line item, calculated by the system based on the quantity)*
- **PositiveDouble**: `palletPackingRatio` *(Pallet Packing Ratio)*
Ratio: the quantity divided by the possible quantity given the number of pallets. It is a measure of how full the pallets are. For example, if 10 cs fit in one pallet, and we have 12 cs and 2 pallets, the pallet packing ratio is $\frac{12}{(10 \times 2)} = 0.6$. But if we have 20 cs and 2 pallets, the pallet packing ratio is $\frac{20}{(10 \times 2)} = 1.0$.

- CustomFields: `poItemCustomFields` (inventory type custom fields for the PO Items)

### 2.3.2 Receipt lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReceiptLookupResult</td>
<td>ResultMetadata: resultMetadata</td>
</tr>
<tr>
<td></td>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
</tr>
<tr>
<td></td>
<td>Receipt: receipts – can occur 0 or more times (up to 1000); the information about each receipt itself. Each Receipt contains (references are to the Receiving&gt;View Receipts&gt;View Receipt #: RCPT-nnnn screen):</td>
</tr>
<tr>
<td></td>
<td>• IdType: id (Receipt #)</td>
</tr>
<tr>
<td></td>
<td>• SaveStateEnum: saveState (deprecated in favor the WSSectionStatusEnum status)</td>
</tr>
<tr>
<td></td>
<td>• WSSectionStatusEnum: status (Status)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: scheduledDate (Scheduled Receipt Date)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: date (Receipt Date)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: dateDue (Due Date)</td>
</tr>
<tr>
<td></td>
<td>• NameType: receivedByName (Assigned To)</td>
</tr>
<tr>
<td></td>
<td>• Warehouse: warehouse (Warehouse)</td>
</tr>
<tr>
<td></td>
<td>• string: warehouseName (Warehouse)</td>
</tr>
<tr>
<td></td>
<td>• Affiliate: supplier (Vendor)</td>
</tr>
<tr>
<td></td>
<td>• NameType: supplierName (Vendor name)</td>
</tr>
<tr>
<td></td>
<td>• IdType: dockNumber (Receipt Details&gt;Dock #)</td>
</tr>
<tr>
<td></td>
<td>• IdType: bolNumber (Receipt Details&gt;Bill of lading)</td>
</tr>
<tr>
<td></td>
<td>• IdType: containerNumber (Receipt Details&gt;Container #)</td>
</tr>
<tr>
<td></td>
<td>• IdType: shipmentRefNumber (Receipt Details&gt;Shipment reference)</td>
</tr>
<tr>
<td></td>
<td>• CommentType: comments (Comments)</td>
</tr>
<tr>
<td></td>
<td>• CreateRecord: createRecord (contains Date created and Created by)</td>
</tr>
<tr>
<td></td>
<td>• ReceiptItem: item (from Receipt Items table; can occur 1 or more times)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: createdDate (Date created)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: lastModifiedDate (not displayed in SmartTurn user interface)</td>
</tr>
</tbody>
</table>
|                         | • IdType: poExternalRefNumber (the external reference number, External #, for this receipt’s purchase order, if this receipt was
created from an external (to SmartTurn) purchase order

- `string:priority (Priority)`
- `NameType:supplierCode (Code, from the Vendor's business affiliate record)`
- `string:ownerName (Owner)`
- `NameType:ownerCode (Code, from the Owner's business affiliate record)`
- `string:carrierName (Carrier)`
- `NameType:carrierCode (Code, from the Carrier's business affiliate record)`
- `string:brokerName (Broker)`
- `NameType:brokerCode (Code, from the Broker's business affiliate record)`
- `dateTime:completeDate (Date received)`
- `WSSectionStatusEnum:billingStatus (Billing>Billing Status; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)`
- `CustomFields:receiptCustomFields (fields in the Custom tab; see Section 2.4, Flex Field Types)`

### ReceiptItem

- `int:linenumber (Line #)`
- `IdType:itemMasterId (Item #)`
- `PurchaseOrderItemRef:poItemInfo (information about the purchase order item that this receipt item was created from, including External # for the purchase order, PO # for the purchase order, Line # from the purchase order, and External # from the purchase order item)`
- `DescriptionType:description (Description)`
- `DetailType:details (Details)`
- `Quantity:quantity (Received qty)`
- `Quantity:previouslyReceivedQuantity (deprecated)`
- `Quantity:orderedQuantity (Ordered qty)`
- `VendorItemIdType:vendorItemId (Product ID)`
- `DescriptionType:manufacturer (Manufacturer)`
- `ManufacturerIdType:manufacturerId (Manufacturer ID)`
- `ManufacturerItemIdType:manufacturerItemId (Manufacturer Item ID)`
- `UpcCodeType:upcCode (UPC)`
- `NumberType:serialNumber (Serial #)`
- `boolean:isInspectionRequired (not displayed in SmartTurn user interface)`
- `NameType:assignedLocation (Location)`
- `NameType:owner (Owner)`
- `NumberType:customerReferenceNumber (Customer ref #)`
- `NumberType:lessee (deprecated)`
- `NumberType:leaseNumber (deprecated)`
- `NumberType:lotNumber (Lot #)`
- **InventoryConditionEnum**: `condition` (Condition)
- **dateTime**: `expireDate` (Expire date)
- **WSActivityNodeStatusEnum**: `status` (Item Status)
- **IdType**: `licenseNumber` (License #)
- **Currency**: `costTotal` (the total cost for the line, that is, inventory item cost * Received qty)
- **Quantity**: `weightPerUom` (`Dim Wt per UoM`: the dim. weight of this line item per unit of measure. For example, if the `Dim Wt` is 10 lb and the line item is for 4 cs, then the `weightPerUom` is 2.5 lb)
- **Quantity**: `weightTotal` (`Dim Wt`: the total dim. weight of this line item as entered by the user)
- **Quantity**: `cubePerUom` (`Cube per UoM`: the cube or volume of this line item, per unit of measure)
- **Quantity**: `cubeTotal` (`Cube`: the total cube or volume of this line item as entered by the user)
- **integer**: `pallets` (`Pallets`: the total number of separate pallets for this line item, as entered by the user)
- **PositiveDouble**: `palletsFilled` (`Pallets Filled`: the total number of pallets filled for this line item, calculated by the system based on the quantity)
- **PositiveDouble**: `palletPackingRatio` (`Pallet Packing Ratio`: the quantity divided by the possible quantity given the number of pallets. It is a measure of how full the pallets are. For example, if 10 cs fit in one pallet, and we have 12 cs and 2 pallets, the pallet packing ratio is 12 / (10 * 2) = 0.6. But if we have 20 cs and 2 pallets, the pallet packing ratio is 20 / (10 * 2) = 1.0.)
- **NameType**: `putawayUoMAbbreviation` (`Putaway Unit`: the abbreviation for the unit of measure in which the item is put away in the warehouse; may be different from the unit of measure in which the item is being received)
- **Quantity**: `putawayWeightTotal` (`Putaway Dim Wt`: the dim. weight of the item being put away, as entered by the user)
- **Quantity**: `putawayWeightPerUom` (`Putaway Dim Wt Per UoM`)
- **Quantity**: `putawayCubeTotal` (`Putaway Cube`: the cube or volume of the item being put away, as entered by the user)
- **Quantity**: `putawayCubePerUom` (`Putaway Cube Per UoM`)
- **integer**: `putawayPallets` (`Putaway Pallets`: the total number of separate pallets that the item is being put away on, as entered by the user)
- **PositiveDouble**: `putawayPalletsFilled` (`Putaway Pallets Filled`)
- **PositiveDouble**: `putawayPalletsPackingRatio` (`Putaway Pallet Packing Ratio`)
- **CustomFields**: `receiptItemCustomFields` (inventory type custom fields for the Receipt Items)
### 2.3.3 Inventory transaction lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdjustmentsLookupResult</td>
<td>ResultMetadata: resultMetadata</td>
</tr>
<tr>
<td></td>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
</tr>
<tr>
<td></td>
<td>WSAdjustment: adjustments – can occur 0 or more times (up to 1000); the information about each adjustment itself. Each WSAdjustment contains (references are to Inventory&gt;Adjustments&gt;View):</td>
</tr>
<tr>
<td></td>
<td>• dateTime: adjustDate (Transaction Date)</td>
</tr>
<tr>
<td></td>
<td>• NameType: adjustedBy (Transacted By)</td>
</tr>
<tr>
<td></td>
<td>• CommentType: comment (Comments)</td>
</tr>
<tr>
<td></td>
<td>• IdType: id (Adjustment #)</td>
</tr>
<tr>
<td></td>
<td>• Warehouse: warehouse (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• NameType: warehouseName (Warehouse)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: createdDate (Date created)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: completeDate (Complete Date)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: lastModifiedDate (not displayed in SmartTurn user interface)</td>
</tr>
<tr>
<td></td>
<td>• NameType: createdBy (Created by)</td>
</tr>
<tr>
<td></td>
<td>• WSSectionStatusEnum: status (Status; for adjustments, will be one of SAVED, CLOSED, CANCELLED)</td>
</tr>
<tr>
<td></td>
<td>• WSSectionStatusEnum: billingStatus (Billing&gt;Billing Status; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)</td>
</tr>
<tr>
<td></td>
<td>• WSAdjustmentItem: items (from Adjust items table; can occur 1 or more times)</td>
</tr>
<tr>
<td>WSAdjustmentItem</td>
<td>• NumberType: lineNumber (Line #)</td>
</tr>
<tr>
<td></td>
<td>• IdType: itemmasterId (Item #)</td>
</tr>
<tr>
<td></td>
<td>• DescriptionType description (Description)</td>
</tr>
<tr>
<td></td>
<td>• DescriptionType: manufacturer (Manufacturer)</td>
</tr>
<tr>
<td></td>
<td>• NameType: ownerName (Owner)</td>
</tr>
<tr>
<td></td>
<td>• NameType: ownerCode (Code from the Owner’s business affiliate record)</td>
</tr>
<tr>
<td></td>
<td>• IdType: licenseTag (License #)</td>
</tr>
<tr>
<td></td>
<td>• IdType: serialNumber (Serial #)</td>
</tr>
<tr>
<td></td>
<td>• IdType: LotNumber (Lot #)</td>
</tr>
<tr>
<td></td>
<td>• Quantity: quantityInLocation (deprecated; use previousQuantity)</td>
</tr>
<tr>
<td></td>
<td>• NameType: location (Location)</td>
</tr>
<tr>
<td></td>
<td>• Quantity: newQuantity (New Quantity)</td>
</tr>
</tbody>
</table>
CycleCountsLookupResult

ResultMetadata: resultMetadata
string: error – contains any error messages that resulted when processing the lookup

WSCycleCount: cyclecounts – can occur 0 or more times (up to 1000); the information about each cycle count itself. Each WSCycleCount contains (references are to Inventory>Cycle Counts>View):

- IdType: id (Cycle Count #)
- dateTime: createdDate (Created date)
- NameType: createdBy (Created by)
- NameType: warehouseName (Warehouse)
- dateTime: cycleCountDate (Transaction Date)
- NameType: cycleCountedBy (Transacted By)
- CommentType: comment (Comments)
- dateTime: completeDate (Complete Date)
- dateTime: lastModifiedDate (not displayed in SmartTurn user interface)
- WSSectionStatusEnum: status (Status; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)
- WSSectionStatusEnum: billingStatus (Billing>Billing Status; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)
- string: priority
- string: ownerName (Owner)
- string: inventoryManagerName (Inventory Manager)
- WSCycleCountItem: items (from Cycle Count Items table; can occur 1 or more times)

WSCycleCountItem

- NumberType: lineNumber (Line #)
- IdType: itemmasterId (Item #)
- DescriptionType: description (Description)
- DescriptionType: manufacturer (Manufacturer)
- IdType: licenseTag (License #)
- IdType: serialNumber (Serial #)
- IdType: lotNumber (Lot #)
- Quantity: quantityInLocation (Previous Quantity; the amount of the item in the location before the

- NameType: reasonName (Adjustment Reason)
- Currency: costTotal (the inventory cost for the line, inventory item cost * New Quantity)
- Quantity: previousQuantity (Current qty)
- Currency: previousCostTotal (the previous inventory cost for the line, inventory item cost * Current qty)
cycle count)

- **NameType**: `locationName` *(Location)*
- **Quantity**: `newQuantity` *(Counted Quantity; the amount of the item the user counts in the location; note that this is only the new quantity in the location if the Item Status is Closed)*
- **NameType**: `reasonName` *(deprecated)*
- **NameType**: `ownerName` *(Owner)*
- **NameType**: `ownerCode` *(Code from the Owner’s business affiliate record)*
- **boolean**: `counted` *(deprecated)*
- **IdType**: `previousCycleCountId` *(deprecated)*
- **Currency**: `costTotal` *(the inventory cost for the line, inventory item cost * Counted Quantity)*
- **dateTime**: `dateCounted` *(Date Counted; the timestamp of when this line became counted and the inventory adjustment was made)*
- **NameType**: `countedBy` *(Counted By; the username of the user who counted this line)*
- **WSActivityNodeStatusEnum**: `itemStatus` *(Item Status; for cycle counts, will be SAVED, CLOSED, or CANCELLED)*

<table>
<thead>
<tr>
<th>WSBinTransferLookupResult</th>
<th>ResultMetadata: <code>resultMetadata</code> string: <code>error</code> – contains any error messages that resulted when processing the lookup</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSBinTransfer: <code>binTransfers</code> – can occur 0 or more times (up to 1000); the information about each bin transfer itself. Each WSBinTransfer contains (references are to Inventory&gt;Bin Transfers&gt;View):</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>dateTime</strong>: <code>transferredDate</code> <em>(Transaction Date)</em></td>
</tr>
<tr>
<td></td>
<td><strong>NameType</strong>: <code>transferredBy</code> <em>(Transacted By)</em></td>
</tr>
<tr>
<td></td>
<td><strong>CommentType</strong>: <code>comments</code> <em>(Comments)</em></td>
</tr>
<tr>
<td></td>
<td><strong>IdType</strong>: <code>id</code> <em>(Bin Transfer #)</em></td>
</tr>
<tr>
<td></td>
<td><strong>string</strong>: <code>warehouseName</code> <em>(Warehouse)</em></td>
</tr>
<tr>
<td></td>
<td><strong>dateTime</strong>: <code>createdDate</code> <em>(Date created)</em></td>
</tr>
<tr>
<td></td>
<td><strong>dateTime</strong>: <code>completeDate</code> <em>(Complete Date)</em></td>
</tr>
<tr>
<td></td>
<td><strong>dateTime</strong>: <code>lastModified</code> <em>(not displayed in SmartTurn user interface)</em></td>
</tr>
<tr>
<td></td>
<td><strong>NameType</strong>: <code>createdBy</code> <em>(Created by)</em></td>
</tr>
<tr>
<td></td>
<td><strong>WSSectionStatusEnum</strong>: <code>billingStatus</code> <em>(Billing&gt;Billing Status; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)</em></td>
</tr>
<tr>
<td></td>
<td><strong>WSSectionStatusEnum</strong>: <code>status</code> <em>(Status; for bin transfers, will be SAVED, CLOSED, CANCELLED)</em></td>
</tr>
<tr>
<td></td>
<td>*<em>WSBinTransferItemList: <code>items</code> (from the Transfer Items table; required; must have exactly one list of</em></td>
</tr>
</tbody>
</table>
On-Demand WMS (SmartTurn) Web Services - Web Services API

**WSBinTransferItemList** contains at least one and up to 1000 **WSBinTransferItem**s:

- integer: **lineNumber** (Line #)
- IdType: **itemMasterId** (Item #)
- DescriptionType: **description** (Description)
- DateTimeType: **expireDate** (Expire Date)
- string: **licenseNumber** (License #)
- string: **lotNumber** (Lot #)
- IdType: **serialNumber** (Serial #)
- NameType: **currentLocationName** (Original Location)
- Quantity: **currentQuantity** (Current qty)
- Quantity: **transferQuantity** (Transfer qty)
- NameType: **toLocationName** (Location to)
- **WSBinTransferReasonEnum**: **transferReason** (Transfer reason; can be TRANSFER, REPLENISHMENT, CYCLECOUNT, PRODUCTION)
- string: **ownerName** (Owner)
- NameType: **ownerCode** (Code from the Owner’s business affiliate record)
- string: **productId** (Product ID)
- NameType: **itemmasterGroupName** (Group)
- dateTime: **receiptDate** (Item Receipt Date)
- NameType: **details** (Details)
- string: **vendorName** (Vendor)
- NameType: **vendorCode** (Code from the Vendor’s business affiliate record)
- string: **brokerName** (Broker)
- NameType: **brokerCode** (Code from the Broker’s business affiliate record)
- InventoryConditionEnum: **condition** (Condition)
- NameType: **customerRefNumber** (Customer ref #)
- Currency: **unitCost** (the inventory item cost for the unit of measure being transferred; not displayed in the SmartTurn user interface)
  - It is strongly recommended that **totalCost**, which reports the total inventory value of the line, is used rather than **unitCost**, due to details of implementation and how these values are persisted.
- Currency: **totalCost** (the inventory item cost for the unit of measure being transferred * Transfer qty; not displayed in the SmartTurn user interface)
- CustomFields: **binTransferItemCustomFields** (inventory custom fields for the Transfer Items)
## 2.3.4 Sales order lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSSalesOrderLookupResult</td>
<td>ResultMetadata: resultMetadata</td>
</tr>
<tr>
<td></td>
<td>string: error – contains any error messages that resulted</td>
</tr>
<tr>
<td></td>
<td>when processing the lookup</td>
</tr>
<tr>
<td></td>
<td>SalesOrder: salesOrders – can occur 0 or more times (up to 100); the information about</td>
</tr>
<tr>
<td></td>
<td>each sales order itself. Each</td>
</tr>
<tr>
<td></td>
<td>SalesOrder contains (references are to Ordering&gt;Orders&gt;View&gt;View Sales Order #: SO-nnnn):</td>
</tr>
<tr>
<td></td>
<td>• IdType: number (SO #)</td>
</tr>
<tr>
<td></td>
<td>• IdType: externalNumber (External #)</td>
</tr>
<tr>
<td></td>
<td>• SalesOrderTypeEnum: type (Source; one of INTERNAL or EXTERNAL)</td>
</tr>
<tr>
<td></td>
<td>• PositiveInteger: changedOrderNumber (unused)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: date (SO date)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: dateDue (Due date)</td>
</tr>
<tr>
<td></td>
<td>• Warehouse: warehouse (Warehouse; required only if the warehouse is different from the</td>
</tr>
<tr>
<td></td>
<td>warehouse is different from the default warehouse for this login)</td>
</tr>
<tr>
<td></td>
<td>• string: warehouseName (Warehouse)</td>
</tr>
<tr>
<td></td>
<td>• NameType: shipFromName (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• ContactType: shipFromContact (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• PhoneType: shipFromContactPhone (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• Address: shipFrom (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• NameType: customerName (Bill To&gt;Bill to customer)</td>
</tr>
<tr>
<td></td>
<td>• ContactType: customerContact (Bill To&gt;Bill to contact)</td>
</tr>
<tr>
<td></td>
<td>• PhoneType: customerContactPhone (Bill To&gt;Bill to phone)</td>
</tr>
<tr>
<td></td>
<td>• Address: customerAddress (Bill To&lt;Address line 1, Address line 2, City, State, Country, Postal Code)</td>
</tr>
<tr>
<td></td>
<td>• NameType: salesRep (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• boolean: useShipToAsBillAddress (Bill To&gt;Same as Ship To; if true, customer name</td>
</tr>
<tr>
<td></td>
<td>provided, will be overwritten by the ship to info)</td>
</tr>
<tr>
<td></td>
<td>• NameType: shipToName (Ship To&gt;Ship to customer)</td>
</tr>
<tr>
<td></td>
<td>• ContactType: shipToContact (Ship To&gt;Ship to contact)</td>
</tr>
<tr>
<td></td>
<td>• PhoneType: shipToContactPhone (Ship To&gt;Ship to phone)</td>
</tr>
<tr>
<td></td>
<td>• Address: ShipTo (Ship To&lt;Address line 1, Address line 2, City,}</td>
</tr>
</tbody>
</table>
### SalesOrder

- State, Country, Postal Code
- Affiliate: customer (Ship To: Customer)
- SalesOrderStatusEnum: status (Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)
- SalesOrderItem: item (from the Items table; required, 1 or more)
- CommentType: comments (Comments)
- boolean: allowPartialShipment (Partial Shipment Allowed)
- Affiliate: owner (deprecated)
- Affiliate: carrier (deprecated)
- Affiliate: broker (deprecated)
- NameType: orderedBy (Ordered by)
- string: priority (Priority)
- NameType: customerCode (Code from the Customer’s business affiliate record)
- string: ownerName (Owner)
- NameType: ownerCode (Code from the Owner’s business affiliate record)
- string: carrierName (Carrier)
- NameType: carrierCode (Code from the Carrier’s business affiliate record)
- string: brokerName (Broker)
- NameType: brokerCode (Code from the Broker’s business affiliate record)
- dateTime: createdDate (Date created; only populated on lookup)
- dateTime: lastModifiedDate (not displayed in SmartTurn user interface; only populated on lookup)
- WSSectionStatusEnum: billingStatus (Billing> Billing Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)
- IdType: waveNumber (Wave #; only populated on lookup)
- CustomFields: soCustomFields (fields from the Custom tab; see Section 2.4, Flex Field Types)

### SalesOrderItem

- int: lineNumber (Line #)
- IdType: itemMasterId (Item #)
- DescriptionType: description (Description)
- DetailType: details (Details)
- Quantity: shippedQuantity (deprecated)
- Quantity: orderedQuantity (Ordered Qty)
- dateTime: customerRequestDate (Requested ship date)
- Currency: price (Unit Price; note that if the user enters an Override Price and saves the sales order, that Override Price becomes the Unit Price)
<table>
<thead>
<tr>
<th>DescriptionType: manufacturer</th>
<th>(Manufacturer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VendorItemIdType: vendorItemId</td>
<td>(Product ID; if the item master is set for Pick on item attributes, this value can be provided in the saveSalesOrder request, and if provided must match the productId element; if the item master is not set for Pick on item attributes, this value is ignored in the saveSalesOrder request and is only provided on lookup)</td>
</tr>
<tr>
<td>ManufacturerItemIdType: manufacturerItemId</td>
<td>(Manufacturer Item ID)</td>
</tr>
<tr>
<td>UpcCodeType: upcCode</td>
<td>(UPC)</td>
</tr>
<tr>
<td>ManufacturerIdType: manufacturerId</td>
<td>(Manufacturer ID)</td>
</tr>
<tr>
<td>IdType: externalRefNumber</td>
<td>(External #, from the sales order)</td>
</tr>
<tr>
<td>Quantity: weightPerUoM</td>
<td>(Dim Wt per UoM: the dim. weight of this line item per unit of measure. For example, if the Dim Wt is 10 lb and the line item is for 4 cs, then the weightPerUom is 2.5 lb)</td>
</tr>
<tr>
<td>Quantity: weightTotal</td>
<td>(Dim Wt: the total dim. weight of this line item as entered by the user)</td>
</tr>
<tr>
<td>Quantity: cubePerUom</td>
<td>(Cube per UoM: the cube or volume of this line item, per unit of measure)</td>
</tr>
<tr>
<td>Quantity: cubeTotal</td>
<td>(Cube: the total cube or volume of this line item as entered by the user)</td>
</tr>
<tr>
<td>integer: pallets</td>
<td>(Pallets: the total number of separate pallets for this line item, as entered by the user)</td>
</tr>
<tr>
<td>PositiveDouble: palletsFilled</td>
<td>(Pallets Filled: the total number of pallets filled for this line item, calculated by the system based on the quantity)</td>
</tr>
<tr>
<td>PositiveDouble: palletPackingRatio</td>
<td>(Pallet Packing Ratio: the quantity divided by the possible quantity given the number of pallets. It is a measure of how full the pallets are. For example, if 10 cs fit in one pallet, and we have 12 cs and 2 pallets, the pallet packing ratio is 12 / (10 * 2) = 0.6. But if we have 20 cs and 2 pallets, the pallet packing ratio is 20 / (10 * 2) = 1.0.)</td>
</tr>
<tr>
<td>string: lotNumber</td>
<td>(Lot #)</td>
</tr>
<tr>
<td>string: serialNumber</td>
<td>(Serial #)</td>
</tr>
<tr>
<td>string: productId</td>
<td>(Product ID; if the item master is set for Pick on item attributes, this value can be provided in the saveSalesOrder request, and if provided must match the vendorItemId element; if the item master is not set for Pick on item attributes, this value is ignored in the saveSalesOrder request and is only provided on lookup)</td>
</tr>
<tr>
<td>DateTimeType: expireDate</td>
<td>(Expire Date)</td>
</tr>
<tr>
<td>string: licenseNumber</td>
<td>(License #)</td>
</tr>
<tr>
<td>CustomFields: soItemCustomFields</td>
<td>(inventory type custom fields for the Items table)</td>
</tr>
</tbody>
</table>
### 2.3.5 Shipment lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShipmentLookupResult</td>
<td>ResultMetadata: resultMetadata</td>
</tr>
<tr>
<td></td>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
</tr>
<tr>
<td></td>
<td>Shipment: shipments – can occur 0 or more times (up to 1000); the information about each shipment itself. Each Shipment contains (references are to the Fulfillment&gt;View Shipments&gt;View Shipment #: SImnnn screen):</td>
</tr>
<tr>
<td></td>
<td>• CommentType: comments (Comments)</td>
</tr>
<tr>
<td></td>
<td>• NameType: customerName (Ship To&gt;Ship to customer; note that on the Add or Edit Shipment screen, this field is labelled Ship To&gt;Override Ship to customer; in both cases it is the overridden value of the Ship To&gt;Customer field)</td>
</tr>
<tr>
<td></td>
<td>• ContactType: customerContact (Ship To&gt;Ship to contact)</td>
</tr>
<tr>
<td></td>
<td>• PhoneType: customerPhone (Ship To&gt;Ship to phone)</td>
</tr>
<tr>
<td></td>
<td>• Affiliate: customer (Ship To&gt;Customer)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: date (Shipment date)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: dueDate (Due date)</td>
</tr>
<tr>
<td></td>
<td>• IdType: id (Shipment #)</td>
</tr>
<tr>
<td></td>
<td>• boolean: hasVoidedItem (Voided; true if the shipment has one or more completely or partially voided shipment line items)</td>
</tr>
<tr>
<td></td>
<td>• ShipmentItem: item (from Ship Items table; can occur 0 or more times)</td>
</tr>
<tr>
<td></td>
<td>• SaveStateEnum: saveState (depreceated)</td>
</tr>
<tr>
<td></td>
<td>• Address: shipFrom (Ship From&gt;Address line 1, Address line 2, City, State, Country, Postal code)</td>
</tr>
<tr>
<td></td>
<td>• NameType: shippedByName (Ship From&gt;Shipped via)</td>
</tr>
<tr>
<td></td>
<td>• Address: shipTo (Ship To&gt;Address line 1, Address line 2, City, State, Country, Postal code)</td>
</tr>
<tr>
<td></td>
<td>• ShipmentStateEnum: state (Status)</td>
</tr>
<tr>
<td></td>
<td>• Currency: tax (not displayed in SmartTurn user interface)</td>
</tr>
<tr>
<td></td>
<td>• Currency: total (Total amount)</td>
</tr>
<tr>
<td></td>
<td>• Warehouse: warehouse (depreceated)</td>
</tr>
<tr>
<td></td>
<td>• string: priority (Priority)</td>
</tr>
<tr>
<td></td>
<td>• NameType: warehouseName (Warehouse)</td>
</tr>
<tr>
<td></td>
<td>• CreateRecord: createRecord (contains Date created and Created by)</td>
</tr>
<tr>
<td></td>
<td>• IdType: loadReference (Load reference)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: createdDate (Date created)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: lastModifiedDate (not displayed in SmartTurn user interface)</td>
</tr>
<tr>
<td></td>
<td>• dateTime: dateShipped (Date shipped)</td>
</tr>
<tr>
<td>Field Type</td>
<td>Field Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IdType: soExternalRefNumber</td>
<td>(the external reference number, External #, for this shipment's sales order, if this shipment was created from an external (to SmartTurn) sales order)</td>
</tr>
<tr>
<td>string: shippingContactEmail</td>
<td>(from the Ship To&gt;Customer’s business affiliate record, the Email of the Shipment contact)</td>
</tr>
<tr>
<td>NameType: customerCode</td>
<td>(Code, from the Customer’s business affiliate record)</td>
</tr>
<tr>
<td>string: ownerName</td>
<td>(Owner)</td>
</tr>
<tr>
<td>NameType: ownerCode</td>
<td>(Code, from the Owner’s business affiliate record)</td>
</tr>
<tr>
<td>string: carrierName</td>
<td>(Carrier)</td>
</tr>
<tr>
<td>NameType: carrierCode</td>
<td>(Code, from the Carrier’s business affiliate record)</td>
</tr>
<tr>
<td>string: brokerName</td>
<td>(Broker)</td>
</tr>
<tr>
<td>NameType: brokerCode</td>
<td>(Code, from the Broker’s business affiliate record)</td>
</tr>
<tr>
<td>DescriptionType: billToCustomerName</td>
<td>(Bill To&gt;Bill to customer)</td>
</tr>
<tr>
<td>DescriptionType: billToCustomerContact</td>
<td>(Bill To&gt;Bill to contact)</td>
</tr>
<tr>
<td>DescriptionType: billToCustomerPhone</td>
<td>(Bill To&gt;Bill to phone)</td>
</tr>
<tr>
<td>Address: billToAddress</td>
<td>(Bill To&gt;Address line 1, Address line 2, City, State, Country, Postal code)</td>
</tr>
<tr>
<td>WSSectionStatusEnum: billingStatus</td>
<td>(Billing&gt;Billing Status; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)</td>
</tr>
<tr>
<td>Quantity: totalCube</td>
<td>(Total Cube)</td>
</tr>
<tr>
<td>Quantity: totalDimWeight</td>
<td>(Total Dim. Weight)</td>
</tr>
<tr>
<td>integer: totalPallets</td>
<td>(Total Pallets)</td>
</tr>
<tr>
<td>PositiveDouble: totalPalletsFilled</td>
<td>(Total pallets filled)</td>
</tr>
<tr>
<td>CustomFields: shipmentCustomFields</td>
<td>(fields in the Custom tab; see Section 2.4, Flex Field Types)</td>
</tr>
<tr>
<td>DescriptionType: parcelCompanyName</td>
<td>(Parcel Post&gt;Parcel Company)</td>
</tr>
<tr>
<td>DescriptionType: parcelContact</td>
<td>(Parcel Post&gt;Parcel Contact)</td>
</tr>
<tr>
<td>DescriptionType: parcelContactPhone</td>
<td>(Parcel Post&gt;Parcel Phone)</td>
</tr>
<tr>
<td>Address: parcelAddress</td>
<td>(Parcel Post&gt;Address line 1, Address line 2, City, State, Country, Postal Code)</td>
</tr>
<tr>
<td>WSShipRushCarrierTypeEnum: parcelCarrier</td>
<td>(Parcel Post&gt;Parcel Carrier; can be one of: FEDEX, DHL, USPS, UPS, ENDICIA, MANUAL, SHIPMENT_CARRIER)</td>
</tr>
<tr>
<td>DescriptionType: parcelAccountNumber</td>
<td>(Parcel Post&gt;Parcel Account #)</td>
</tr>
<tr>
<td>WSParcelPostStatusEnum: parcelPostStatus</td>
<td>(Parcel Post&gt;Parcel Post Status; the status of the shipment in relation to the ShipRush external application; can be one of: NOT_POSTED, PENDING_POST, POSTED)</td>
</tr>
</tbody>
</table>
### On-Demand WMS (SmartTurn) Web Services - Web Services API

**DescriptionType: parcelServiceType** (Parcel Post>Service Type)

**PositiveInteger: parcelNumberOfPackages** (Parcel Post>Num of Packages)

**DescriptionType: parcelActualWeight** (Parcel Post>Actual Weight)

**DescriptionType: parcelTrackingNumber** (Parcel Post>Tracking #)

**string: parcelTrackingURL** (Parcel Post>Tracking URL)

**Currency: parcelTotalEstimatedCharges** (Parcel Post>Total Est Charges)

**string: parcelManualCarrier** (Parcel Post>Parcel Carrier; only populated when the initial entry, parcelCarrier, is set to MANUAL)

**IdType: waveNumber** (Wave #)

### ShipmentItem

**int: linenumber** (Line #)

**InventoryConditionEnum: condition** (Condition)

**IdType: customerRefNumber** (Customer ref #)

**DescriptionType: description** (Description)

**DetailType: details** (Details)

**NameType: fromLocation** (Location)

**IdType: itemMasterId** (Item #)

**IdType: leaseNumber** (deprecated)

**IdType: lessee** (deprecated)

**IdType: lotNumber** (Lot #)

**DescriptionType: manufacturer** (Manufacturer)

**ManufacturerIdType: manufacturerId** (Manufacturer ID)

**ManufacturerItemIdType: manufacturerItemId** (Manufacturer Item ID)

**NameType: owner** (Owner)

**IdType: serialNumber** (Serial #)

**Currency: price** (Unit Price; note that if the user enters an Override Price and saves the shipment, that Override Price becomes the Unit Price)

  - It is strongly recommended that total, which reports the total price of the line, is used rather than price, due to details of implementation and how these values are persisted.

**Currency: total** (Current Unit Price * Qty Shipped, that is, the total price for the line item)

**UpcCodeType: upcCode** (UPC)

**VendorItemIdType: vendorItemId** (Product ID)

**Quantity: shippedQuantity** (Qty Shipped)

**Quantity: voidedQuantity** (Qty Voided: the total amount, from all voided items, that is voided against this shipment item)

**IdType: licenseNumber** (License #)
<table>
<thead>
<tr>
<th>Field Type</th>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutBoundIdListType</td>
<td>outBoundIdList</td>
<td>Outbound IDs</td>
</tr>
<tr>
<td>SalesOrderItemRef</td>
<td>soItemInfo</td>
<td>Information about the sales order item that this shipment item was created from, including External # for the sales order, SO # for the sales order, Line # from the sales order, and External # from the sales order item</td>
</tr>
<tr>
<td>Quantity</td>
<td>weight</td>
<td>(Qty Shipped * {Weight Unit / Number of Eaches in Unit of Measure for Qty Shipped}), that is, the total weight of the line item</td>
</tr>
<tr>
<td>DateTime</td>
<td>expireDate</td>
<td>Expire Date</td>
</tr>
<tr>
<td>Currency</td>
<td>costTotal</td>
<td>(Qty Shipped * {inventory item cost per each / Number of Eaches in Unit of Measure for Qty Shipped}), that is, the total inventory value for the line, based on the cost at which the inventory was brought into the warehouse; not displayed in SmartTurn user interface</td>
</tr>
<tr>
<td>Quantity</td>
<td>weightPerUom</td>
<td>(Dim Wt per UoM: the dim. weight of this line item per unit of measure. For example, if the Dim Wt is 10 lb and the line item is for 4 cs, then the weightPerUom is 2.5 lb)</td>
</tr>
<tr>
<td>Quantity</td>
<td>weightTotal</td>
<td>(Dim Wt: the total dim. weight of this line item as entered by the user)</td>
</tr>
<tr>
<td>Quantity</td>
<td>cubePerUom</td>
<td>(Cube per UoM: the cube or volume of this line item, per unit of measure)</td>
</tr>
<tr>
<td>Integer</td>
<td>pallets</td>
<td>(Pallets: the total number of separate pallets for this line item, as entered by the user)</td>
</tr>
<tr>
<td>PositiveDouble</td>
<td>palletsFilled</td>
<td>(Pallets Filled: the total number of pallets filled for this line item, calculated by the system based on the quantity)</td>
</tr>
<tr>
<td>PositiveDouble</td>
<td>palletPackingRatio</td>
<td>(Pallet Packing Ratio: the quantity divided by the possible quantity given the number of pallets. It is a measure of how full the pallets are. For example, if 10 cs fit in one pallet, and we have 12 cs and 2 pallets, the pallet packing ratio is 12 / (10 * 2) = 0.6. But if we have 20 cs and 2 pallets, the pallet packing ratio is 20 / (10 * 2) = 1.0.)</td>
</tr>
<tr>
<td>NameType</td>
<td>pickUoMAbbreviation</td>
<td>(Pick Unit: the abbreviation for the unit of measure in which the item was picked for shipment; may be different from the unit of measure in which the item is being shipped)</td>
</tr>
<tr>
<td>Quantity</td>
<td>pickWeightTotal</td>
<td>(Pick Dim Wt)</td>
</tr>
<tr>
<td>Quantity</td>
<td>pickWeightPerUom</td>
<td>(Pick Dim Wt Per UoM)</td>
</tr>
<tr>
<td>Quantity</td>
<td>pickCubeTotal</td>
<td>(Pick Cube)</td>
</tr>
<tr>
<td>Quantity</td>
<td>pickCubePerUom</td>
<td>(Pick Cube Per UoM)</td>
</tr>
<tr>
<td>Integer</td>
<td>pickPallets</td>
<td>(Pick Pallets)</td>
</tr>
<tr>
<td>PositiveDouble</td>
<td>pickPalletsFilled</td>
<td>(Pick Pallets Filled)</td>
</tr>
<tr>
<td>PositiveDouble</td>
<td>pickPalletsPackingRatio</td>
<td>(Pick Pallet Packing Ratio)</td>
</tr>
<tr>
<td>CustomFields</td>
<td>shipmentItemCustomFields</td>
<td>(inventory type custom fields for the Ship Items)</td>
</tr>
</tbody>
</table>
### 2.3.6 Wave fulfillment lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
</table>
| WSPickTicketLookupResult | ResultMetadata: resultMetadata  
string: error – contains any error messages that resulted when processing the lookup  
WSPickTicket: pickTicket – can occur 0 or more times (up to 1000); the information about each pick ticket itself. Each WSPickTicket contains (references are to Wave Fulfillment>Pick Tickets>View Pick Ticket #: PT-nnnn):  
• IdType: pickTicketNumber (PT #)  
• string: warehouseName (Warehouse)  
• dateTime: createdDate (Created date)  
• string: createdBy (Created by)  
• WSSectionStatusEnum: status (Status from the header of the pick ticket)  
• dateTime: completeDate (Complete date)  
• string: pickedBy (Transacted by)  
• dateTime: pickedDate (Transacted date)  
• dateTime: dueDate (Due date)  
• dateTime: scheduledDate (Scheduled date)  
• string: priority (Priority)  
• string: ownerName (Owner)  
• string: ownerCode (Code from the Owner’s business affiliate record)  
• IdType: waveNumber (Wave #)  
• string: transitLocationName (Transit Location)  
• string: comments (Comments)  
• WSPickTicketItem: pickTicketItem (from Pick Ticket Items table, Full View; can occur 0 or more times)  
• WSPickTicketAnomalyItem: pickTicketAnomalyItem (from Rejected Picks tab; can occur 0 or more times) |
| WSPickTicketItem | The WSPickTicketItem reports the pick ticket item as seen from the Full View of the pick ticket in the application (Wave Fulfillment>Pick Tickets>View Pick Ticket #: PT-nnnn, Full View from the Pick Ticket Items table drop-down).  
• int: lineNumber (Line #)  
• IdType: itemMasterId (Item #)  
• DescriptionType: description (Description)  
• string: salesOrderLineNumber (SO # / Line #)  
• string: licenseNumber (Lic #)  
• string: ownerName (Owner)  
• string: ownerCode (Code from the Owner’s business affiliate record)  
• string: serialNumber (Serial #) |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lotNumber</code></td>
<td>Lot #</td>
</tr>
<tr>
<td><code>expireDate</code></td>
<td>Expire Date</td>
</tr>
<tr>
<td><code>itemReceiptDate</code></td>
<td>Item receipt date</td>
</tr>
<tr>
<td><code>productId</code></td>
<td>Product ID</td>
</tr>
<tr>
<td><code>outBoundIdList</code></td>
<td>Outbound IDs</td>
</tr>
<tr>
<td><code>quantityToPick</code></td>
<td>Qty to Pick</td>
</tr>
<tr>
<td><code>quantityPicked</code></td>
<td>Qty Picked</td>
</tr>
<tr>
<td><code>fromLocationName</code></td>
<td>(Original Location)</td>
</tr>
<tr>
<td><code>currentLocationName</code></td>
<td>(the location of the inventory item)</td>
</tr>
<tr>
<td><code>bolPackageLineNumber</code></td>
<td>(BOL # / Pkg # / Line #)</td>
</tr>
<tr>
<td><code>pickedBy</code></td>
<td>Picked by</td>
</tr>
<tr>
<td><code>datePicked</code></td>
<td>Date picked</td>
</tr>
<tr>
<td><code>itemStatus</code></td>
<td>Status</td>
</tr>
<tr>
<td><code>itemMasterId</code></td>
<td>Item #</td>
</tr>
<tr>
<td><code>description</code></td>
<td>Description</td>
</tr>
<tr>
<td><code>lotNumber</code></td>
<td>Lot #</td>
</tr>
<tr>
<td><code>serialNumber</code></td>
<td>Serial #</td>
</tr>
<tr>
<td><code>licenseNumber</code></td>
<td>Lic #</td>
</tr>
<tr>
<td><code>voidedQuantity</code></td>
<td>Qty Rejected</td>
</tr>
<tr>
<td><code>originalQuantity</code></td>
<td>the amount of the Qty Picked</td>
</tr>
<tr>
<td><code>locationName</code></td>
<td>(Original Location of the pick ticket item)</td>
</tr>
<tr>
<td><code>voidingDate</code></td>
<td>Voiding Date, visible only on the Explore screen; the date the item was rejected</td>
</tr>
<tr>
<td><code>lineNumber</code></td>
<td>Line # from the Rejected Picks tab</td>
</tr>
<tr>
<td><code>originalLineNumber</code></td>
<td>Line # of the pick ticket item rejected, that is, Pick Ticket Line # from the Rejected Picks tab</td>
</tr>
<tr>
<td><code>isInventoryAffected</code></td>
<td>(Returned to inventory)</td>
</tr>
<tr>
<td><code>voidedTotalValue</code></td>
<td>(Total Cost, not available in UI)</td>
</tr>
<tr>
<td><code>customFields</code></td>
<td>(inventory type custom fields)</td>
</tr>
</tbody>
</table>

**WSPickTicketAnomalyItem**

The `WSPickTicketAnomalyItem` reports the rejected picks (Wave Fulfillment>Pick Tickets>View Pick Ticket #: PT-nnnn, the Rejected Picks tab)

- `itemMasterId` (Item #)
- `description` (Description)
- `lotNumber` (Lot #)
- `serialNumber` (Serial #)
- `expireDate` (Expire Date)
- `licenseNumber` (Lic #)
- `voidedQuantity` (Qty Rejected)
- `originalQuantity` (the amount of the Qty Picked from the pick ticket item that has not yet been rejected)
- `locationName` (Original Location of the pick ticket item)
- `voidingDate` (Voiding Date, visible only on the Explore screen; the date the item was rejected)
- `lineNumber` (Line # from the Rejected Picks tab)
- `originalLineNumber` (Line # of the pick ticket item rejected, that is, Pick Ticket Line # from the Rejected Picks tab)
- `isInventoryAffected` (Returned to inventory)
- `voidedTotalValue` (Total Cost, not available in UI)
- `customFields` (inventory type custom fields)

**WSBOLLookupResult**

`resultMetadata` (resultMetadata)

- `error` contains any error messages that resulted when processing the lookup
WSBOL: `bol` – can occur 0 or more times (up to 1000); the information about each bill of lading itself. Each `WSBOL` contains (references are to Wave Fulfillment>Bills of Lading>View Bill of Lading #: BOL-nnnn):

- **IdType**: `bolNumber` *(BOL #)*
- **string**: `warehouseName` *(Warehouse)*
- **dateTime**: `createdDate` *(Created date)*
- **string**: `createdBy` *(Created by)*
- **WSSectionStatusEnum**: `status` *(Status)*
- **dateTime**: `completeDate` *(Complete date)*
- **IdType**: `waveNumber` *(Wave #)*
- **string**: `transactedBy` *(Transacted by)*
- **dateTime**: `transactionDate` *(Date)*
- **string**: `carrierName` *(Carrier)*
- **string**: `carrierCode` *(Code from the Carrier's business affiliate record)*
- **string**: `customerName` *(Customer)*
- **string**: `customerCode` *(Code from the Customer's business affiliate record)*
- **string**: `ownerName` *(Owner)*
- **string**: `ownerCode` *(Code from the Owner's business affiliate record)*
- **string**: `comments` *(Comments)*
- **Address**: `shipToAddress` *(address fields on Ship To tab)*
- **string**: `shipToName` *(Override Customer on Ship To tab)*
- **Address**: `shipFromAddress` *(address fields on Ship From tab)*
- **string**: `shipFromName` *(Warehouse on Ship From tab)*
- **Address**: `remitToAddress` *(address fields on Remit To tab)*
- **string**: `remitToName` *(Remit To on Remit To tab)*
- **CustomFields**: `customFields` *(custom fields with Business Process of Outbound; see Section 2.4, Flex Field Types)*

- **WSBOLPackage**: `bolPackage`. Each `bolPackage` includes:
  - **WSActivityNodeStatusEnum**: `status` *(Status)*
  - **integer**: `packageNumber` *(Package #)*
  - **string**: `trackingNumber` *(Tracking #)*
  - **string**: `trackingUrl` *(Tracking URL)*
  - **Currency**: `totalEstimatedCharges` *(Total Estimated Charges)*
  - **Quantity**: `cube` *(Cube)*
  - **Quantity**: `dimWeight` *(Dim Wt)*
  - **integer**: `pallets` *(Pallets)*
  - **HazardousFlagEnum**: `hazardous` *(Hazardous)*
  - **string**: `comments` *(Comments)*
- **WSPackItem**: `packItem` *(these are the BOL items, from the Pack View of bill of lading in the application, that are in this*
### WSPackItem

The **WSPackItem** reports the BOL items from the **Pack View** of the bill of lading in the application (Wave Fulfillment>Bills of Lading>View Bill of Lading #: BOL-nnnn, Pack View from the Pack Items table drop-down)

- integer: **lineNumber** (Line #)
- IdType: **itemMasterId** (Item #)
- DescriptionType: **description** (Description)
- WSActivityNodeStatusEnum: **itemStatus** (Item Status)
- Quantity: **qtyToPack** (Qty to Pack)
- Quantity: **qtyAvailToPack** (Qty Avail to Pack)
- Quantity: **qtyPacked** (Qty Packed)
- Quantity: **weightPerUoM** (Dim Wt per UoM)
- Quantity: **weightTotal** (Dim Wt)
- Quantity: **cubePerUoM** (Cube per UoM)
- Quantity: **cubeTotal** (Cube)
- integer: **pallets** (Pallets)
- positiveDouble: **palletsFilled** (Pallets Filled)
- positiveDouble: **palletPackingRatio** (Pallet Packing Ratio)
- CustomFields: **customFields** (inventory type custom fields)

### WSDemandItem

The **WSDemandItem** reports the BOL items from the **Demand View** of the bill of lading in the application (Wave Fulfillment>Bills of Lading>View Bill of Lading #: BOL-nnnn, Demand View from the Pack Items table drop-down). The Demand View shows what was ordered. Only the part of the demand item that is a part of the parent **WSPackItem** element is reported.

- integer: **lineNumber** (Line #)
- string: **salesOrderLineNumber** (SO # / Line #)
- IdType: **itemMasterId** (Item #)
- DescriptionType: **description** (Description)
- DetailType: **details** (Details)
- Quantity: **orderedQuantity** (Qty to Pack)
- Quantity: **orderedQuantityWhole** (Ordered Qty)
- dateTime: **customerRequestDate** (Requested ship date from the sales order line)
- Currency: **price** (Unit price)
- DescriptionType: **manufacturer** (Manufacturer)
- VendorItemIdType: **vendorItemId** (Product ID)
- ManufacturerItemIdType: **manufacturerId** (Manufacturer ID)
- IdType: **externalRefNumber** (External # from the sales order, if any)
- Quantity: **weightPerUoM** (Dim Wt per UoM)
- Quantity: **weightTotal** (Dim Wt)
- Quantity: **cubePerUoM** (Cube per UoM)
### On-Demand WMS (SmartTurn) Web Services - Web Services API

**WSFulfillmentItem**

The WSFulfillmentItem reports the BOL items from the Fulfillment View of the bill of lading in the application (Wave Fulfillment>Bills of Lading>View Bill of Lading #: BOL-nnnn, Fulfillment View and Rejected View from the Pack Items table drop-down). The Fulfillment View shows what was picked. The Rejected View shows what was picked and also rejected by the packer (for example, if it was broken, damaged, or otherwise not suitable for sending to the customer). Only the part of the fulfillment item that is a part of the parent WSDemandItem element is reported.

- WSLineTypeEnum: lineType (Line Type, show in the application only in the activity line manager)
- int: lineNumber (Line #)
- IdType: itemMasterId (Item #)
- DescriptionType: description (Description)
- string: licenseNumber (Lic #)
- string: ownerName (Owner)
- string: ownerCode (Code from the Owner’s business affiliate record)
- string: serialNumber (Serial #)
- string: lotNumber (Lot #)
- dateTime: expireDate (Expire Date)
- dateTime: itemReceiptDate (Item receipt date)
- string: productId (Product ID)
- OutBoundIdListType: outBoundIdList (Outbound IDs)
- Quantity: quantityToPick (Qty to Pack, that is, the amount of this fulfillment item that needs to be packed for this WSDemandItem)
- Quantity: qtyAvailToPack (Qty Avail to Pack, that is, the amount of this fulfillment item that has been picked already for this WSDemandItem)
- string: fromLocationName (Original Location)
- string: currentLocationName (the location of the inventory item: Original Location if the item Status is Saved or Cancelled, Transit Location if the item Status is Started, Destination Location if the item Status is Closed; visible in the application as Location on any inventory item report)
- string: destinationLocationName (Destination Location)
2.3.7 Reporting lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>InventoryLookupResult</td>
<td>ResultMetadata: resultMetadata</td>
</tr>
<tr>
<td></td>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
</tr>
<tr>
<td></td>
<td>InventoryItem: inventoryItems – can occur 0 or more times (up to 1000); the information about each item itself. Each InventoryItem contains (references are to an inventory item record, for instance Reports&gt;Inventory by Item&gt;View):</td>
</tr>
<tr>
<td></td>
<td>• IdType: itemMasterId (Item #)</td>
</tr>
<tr>
<td></td>
<td>• DescriptionType: description (Description)</td>
</tr>
<tr>
<td></td>
<td>• DescriptionType: details (Details)</td>
</tr>
<tr>
<td></td>
<td>• IdType: serialNumber (Serial #)</td>
</tr>
<tr>
<td></td>
<td>• LocationName: location (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• NameType: locationName (Location)</td>
</tr>
<tr>
<td></td>
<td>• NameType: warehouseName (Warehouse)</td>
</tr>
<tr>
<td></td>
<td>• Quantity: quantityOnHand (Qty on hand)</td>
</tr>
<tr>
<td></td>
<td>• DateTimeType: expireDate (Expire date)</td>
</tr>
<tr>
<td></td>
<td>• DateTimeType: receiptDate (Item receipt date)</td>
</tr>
<tr>
<td></td>
<td>• IdType: productId (Product ID)</td>
</tr>
<tr>
<td></td>
<td>• NameType: owner (Owner)</td>
</tr>
<tr>
<td></td>
<td>• NameType: ownerCode (Code from the Owner’s business affiliate record)</td>
</tr>
<tr>
<td></td>
<td>• NameType: lessee (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• NameType: lease (deprecated)</td>
</tr>
<tr>
<td></td>
<td>• NameType: customerRefNumber (Customer ref #)</td>
</tr>
<tr>
<td></td>
<td>• NameType: lot (Lot #)</td>
</tr>
<tr>
<td></td>
<td>• InventoryConditionEnum: condition (Condition)</td>
</tr>
<tr>
<td></td>
<td>• DateTimeType: lastCycleCountDate (Last Cycle Count Date)</td>
</tr>
<tr>
<td></td>
<td>• ItemClassificationEnum: itemClassification (Classification)</td>
</tr>
<tr>
<td></td>
<td>• Currency: valueEach (inventory item cost [value] for one “each”, as defined in the item master under Units of)</td>
</tr>
</tbody>
</table>

• string: bolPackageLineNumber (BOL # / Pkg # / Line #)  
• string: pickedBy (Picked by)  
• dateTime: datePicked (Date picked)  
• WSActivityNodeStatusEnum: itemStatus (Status)  
• CustomFields: customFields (custom fields with Business Process of Inventory)
<table>
<thead>
<tr>
<th>Measure: not exposed in SmartTurn user interface</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• WSMethodAcquired: methodAcquired (Method Acquired; one of Purchased or Produced)</td>
<td></td>
</tr>
<tr>
<td>• DescriptionType: manufacturer (Manufacturer)</td>
<td></td>
</tr>
<tr>
<td>• WSLicense: licenses (from the License Numbers table for this inventory item record; one WSLicense element for each license tag of the inventory item; each WSLicense contains a licenseNumber and a quantity)</td>
<td></td>
</tr>
<tr>
<td>• CustomFields: inventoryCustomFields (fields from the Custom tab; see Section 2.4, Flex Field Types)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WSBillingItemsLookupResult</th>
<th>ResultMetadata: resultMetadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
<td></td>
</tr>
<tr>
<td>WSBillingLineItem: billingItems – can occur 0 or more times; the information about each billing line itself. Each WSBillingLineItem contains (references are to the Billing tab of any billable document, and to other fields available on the (View any document)&gt;Edit Billing screen):</td>
<td></td>
</tr>
<tr>
<td>• IdType: documentId (Document #)</td>
<td></td>
</tr>
<tr>
<td>• WSDocumentTypeEnum: documentType (Document Type)</td>
<td></td>
</tr>
<tr>
<td>• WSMetricNodeStatusEnum: billingItemStatus (the Billing Status for document this billing line is from; displayed at the line level as Billing Line Status for example at Administration&gt;Billing Line Manager&gt;View)</td>
<td></td>
</tr>
<tr>
<td>• dateTime: createdDate (Created date for the document)</td>
<td></td>
</tr>
<tr>
<td>• NameType: createdBy (Created by for the document)</td>
<td></td>
</tr>
<tr>
<td>• dateTime: lastModifiedDate (the most recent time that the billing line item was modified; in UI, only exposed in Administration&gt;Billing Line Manager, as Last modified date)</td>
<td></td>
</tr>
<tr>
<td>• NameType: warehouseName (Warehouse)</td>
<td></td>
</tr>
<tr>
<td>• IdType: externalRefNumber (External # from the document, if there is one)</td>
<td></td>
</tr>
<tr>
<td>• NameType: ownerName (Owner)</td>
<td></td>
</tr>
<tr>
<td>• string: ownerCode (the Code of the Business Affiliate Owner, if any, for this business affiliate)</td>
<td></td>
</tr>
<tr>
<td>• IdType: generalLedgerCode (General Ledger Code of the Charge Type for this billing line)</td>
<td></td>
</tr>
<tr>
<td>• WSBillingBasisEnum: billingBasis (Billing Basis for this billing line; can be perUoM, perLine, perDocument, or adHoc)</td>
<td></td>
</tr>
<tr>
<td>• PositiveDouble: quantityValue (Qty for this billing line)</td>
<td></td>
</tr>
<tr>
<td>• NameType: uomName (Unit of Measure for this billing line)</td>
<td></td>
</tr>
<tr>
<td>• DescriptionType: description (Description for this billing line)</td>
<td></td>
</tr>
<tr>
<td>• Currency: billingRate (Rate for this billing line)</td>
<td></td>
</tr>
<tr>
<td>• Currency: billingTotal (the Total for this billing line, that is, the Qty * Rate for this billing line)</td>
<td></td>
</tr>
</tbody>
</table>
| WSItemMasterAtpLookupResult | ResultMetadata: resultMetadata
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
<td></td>
</tr>
</tbody>
</table>
| WSItemMasterAtp: items – can occur 0 or more times (up to 1000); the information about each item master itself. Each WSItemMasterAtp contains (references are to Reports>Available to Promise):
| • idType: id (Item #)
| • DescriptionType: description (Description)
| • Quantity: untaggedQuantityOnHand (Untagged Qty on Hand)
| • Quantity: taggedQuantityOnHand (Tagged Qty on Hand)
| • Quantity: quantityOnHand (Total Qty on Hand)
| • Quantity: pendingKitCreateQuantity (Pending Kit Creation)
| • Quantity: pendingKitConsumptionQuantity (Pending Kit Consumption)
| • Quantity: savedPurchases (Saved Purchases)
| • Quantity: savedSales (Saved Sales)
| • Quantity: pendingPurchases (Pending Purchases)
| • Quantity: pendingSales (Pending Sales)
| • Quantity: atp (Available to Promise)
| • Quantity: overStock (Over (Under) Stock)
| • Quantity: reorderPoint (Reorder Point)
| • Currency: value (Inventory Value; the total value of the quantityOnHand)
| • ItemTypeEnum: type (Admininstration>Item Masters>Item Type)
| • Currency: salesPrice (Administration>Item Masters>Sales/Fulfillment>Sales Price per Shipping Unit)
| • Currency: purchaseCost (Administration>Item Masters>PO/Receiving>Purchase Cost per Receiving Unit)
| • ItemClassification: classification (Administration>Item Masters>Material Handling>Classification)
| • string: vendorItemId (Administration>Item Masters>PO/Receiving>Product ID)
| • string: groupName (Administration>Item Masters>Group)
| • string: preferredVendor (Administration>Item Masters>PO/Receiving>Preferred Vendor)
| • string: preferredVendorCode (Administration>Item Masters>PO/Receiving>Preferred Vendor Code)
| • string: UPC (Administration>Item Masters>General>UPC)
| • string: busAffOwnerName (Administration>Item Masters>General>Owner)
| • string: busAffOwnerCode (the Code of the business)
### WSVoidedItemsLookupResult

<table>
<thead>
<tr>
<th>ResultMetadata: resultMetadata</th>
<th>string: error — contains any error messages that resulted when processing the lookup</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSVoidedItem: items — can occur 0 or more times (up to 1000); the information about each voided (anomaly) item itself. Each WSVoidedItem contains:</td>
<td></td>
</tr>
</tbody>
</table>

- **IdType: documentId** (the document with the voided (anomaly) item: Shipment # or Pick Ticket #)
- **IdType: itemmasterId** (Item #)
- **DescriptionType: description** (Description)
- **IdType: lotNumber** (Lot #)
- **IdType: serialNumber** (Serial #)
- **dateTime: expireDate** (Expire Date)
- **string: licenseNumber** (License #)
- **Quantity: voidedQuantity** (Qty Voided or Qty Rejected)
- **Quantity: originalQuantity** (this is the amount of the original line item that has not yet been voided)
- **NameType: locationName** (Location from the line in the Voided or Rejected Items table)
- **dateTime: voidingDate** (not exposed in SmartTurn user interface)
- **CustomFields: customFields** (inventory type custom fields for the inventory item from the line in the Voided Items table)
- **PositiveInteger: lineNumber** (Line # from the line in the Voided or Rejected Items table)
- **PositiveInteger: originalLineNumber** (Shipment Line # or Pick Ticket Line # from the line in the Voided or Rejected Items table)
- **boolean: isInventoryAffected** (true if inventoryMovement is INBOUND or OUTBOUND, which happens when a voided shipment item’s Return to Inventory field is checked, or when a rejected pick item’s Remove from Inventory field is checked; false if inventoryMovement is NO_EFFECT)
- **WSValueMovementEnum: businessProcess** (Business Process, shown in the Activity Line Manager)
- **Currency: voidedTotalValue** (the inventory value of the Qty Voided or Qty Rejected for the line in the Voided Items table; not exposed in SmartTurn user interface)
- **IdType: externalRefNumber** (the External # from the sales order that this shipment is for, if any)

### WSAactivityLinesLookupResult

<table>
<thead>
<tr>
<th>ResultMetadata: resultMetadata</th>
<th>string: error — contains any error messages that resulted when processing the lookup</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSAactivityLine: items — can occur 0 or more times (up to</td>
<td></td>
</tr>
</tbody>
</table>
1000); the information about each activity line itself. Each WSAActivityLine contains (references are to Administration>Activity Line Manager>View):

- integer: lineNumber (Line #)
- IdType: itemMasterId (Item #)
- string: description (Description)
- WSContentTypeEnum: contentType (Document Type)
- IdType: documentId (Document #)
- WSAActivityNodeStatusEnum: status (Item Status)
- Quantity: quantity (Qty)
- dateTime: date (Business effective date)
- dateTime: createdDate (Date created)
- string: createdBy (Created by)
- dateTime: dueDate (Due Date)
- dateTime: plannedDate (Scheduled date)
- dateTime: completeDate (Complete date)
- string: warehouseName (Warehouse)
- string: ownerName (Owner)
- string: ownerCode (the Code of the Business Affiliate Owner, if any, for this business affiliate)
- string: brokerName (Broker)
- string: carrierName (Carrier)
- string: vendorName (Vendor)
- string: customerName (Customer)
- string: priority (Priority)
- WSLineTypeEnum: lineType (Line Type)
- WSValueMovementEnum: businessProcess (Business Process)
- string: serialNumber (Serial #)
- string: lotNumber (Lot #)
- Currency: costTotal (Total cost, the total inventory value of the line)
- Currency: priceTotal (Total price, the total price of the line; only relevant on sales orders and shipments)
- CustomFields: customFields (inventory custom fields for the line items referenced)

### 2.3.8 Administrative lookup types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ItemMasterLookupResult</td>
<td>ResultMetadata: resultMetadata&lt;br&gt;string: error – contains any error messages that resulted when processing the lookup&lt;br&gt;ItemMaster: itemmasters – can occur 0 or more times</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>IdType: id</td>
<td>(Item #)</td>
</tr>
<tr>
<td>DescriptionType: description</td>
<td>(Description)</td>
</tr>
<tr>
<td>ItemTypeEnum: type</td>
<td>(Item Type)</td>
</tr>
<tr>
<td>Group: group</td>
<td>(Group)</td>
</tr>
<tr>
<td>boolean: lotControlled</td>
<td>(General&gt;Lot Controlled)</td>
</tr>
<tr>
<td>boolean: expirationDateControlled</td>
<td>(General&gt;Expiration Date Controlled)</td>
</tr>
<tr>
<td>boolean: receiptDateControlled</td>
<td>(not used in application currently)</td>
</tr>
<tr>
<td>NameType: manufacturer</td>
<td>(General&gt;Manufacturer)</td>
</tr>
<tr>
<td>ManufacturerIdType: manufacturerId</td>
<td>(General&gt;Manufacturer ID)</td>
</tr>
<tr>
<td>ManufacturerItemIdType: manufacturerItemId</td>
<td>(General&gt;Manufacturer Item ID)</td>
</tr>
<tr>
<td>boolean: serialControlled</td>
<td>(deprecated)</td>
</tr>
<tr>
<td>boolean: serialUnique</td>
<td>(deprecated)</td>
</tr>
<tr>
<td>SerialControlEnum: serialControlType</td>
<td>(one of NONE, UNIQUE, SIMPLE_LABEL, OPEN_CONTAINER, CLOSED_CONTAINER)</td>
</tr>
<tr>
<td>UpcCodeType: upc</td>
<td>(General&gt;UPC)</td>
</tr>
<tr>
<td>EanCodeType: ean</td>
<td>(General&gt;EAN)</td>
</tr>
<tr>
<td>GeneralLedgerCodeType: generalLedgerCode</td>
<td>(General&gt;General Ledger Code)</td>
</tr>
<tr>
<td>HazardousFlagEnum: hazardous</td>
<td>(one of YES, NO; Material Handling&gt;Hazardous)</td>
</tr>
<tr>
<td>ItemClassificationEnum: classification</td>
<td>(one of A, B, C; Material Handling&gt;Classification)</td>
</tr>
<tr>
<td>Currency: purchaseCost</td>
<td>(PO/Receiving&gt;Purchase Cost per Receiving Unit)</td>
</tr>
<tr>
<td>Currency: salesCost</td>
<td>(Sales/Fulfillment&gt;Sales Price per Shipping Unit)</td>
</tr>
<tr>
<td>boolean: sellable</td>
<td>(Sales/Fulfillment&gt;Sellable)</td>
</tr>
<tr>
<td>Unit: standardUnit</td>
<td>(Units of Measure&gt;Standard Unit)</td>
</tr>
<tr>
<td>Unit: measurementUnit</td>
<td>(Units of Measure&gt;Measurement Unit)</td>
</tr>
<tr>
<td>Unit: shippingUnit</td>
<td>(Units of Measure&gt;Shipping Unit)</td>
</tr>
<tr>
<td>Unit: receivingUnit</td>
<td>(Units of Measure&gt;Receiving Unit)</td>
</tr>
<tr>
<td>Unit: weightUnit</td>
<td>(Units of Measure&gt;Weight Unit)</td>
</tr>
<tr>
<td>Unit: billingUnit</td>
<td>(Units of Measure&gt;Billing Unit)</td>
</tr>
<tr>
<td>VendorItemIdType: vendorItemId</td>
<td>(PO/Receiving&gt;Product ID)</td>
</tr>
<tr>
<td>boolean: outBoundIdControlled</td>
<td></td>
</tr>
<tr>
<td>ItemmasterWarehouse: itemMasterWarehouses (an entry in the table on the Warehouses tab)</td>
<td>outBoundIdVerificationStatusEnum: outBoundIdVerificationStatus (one of NO_CHECK (General&gt;Outbound ID Length Not Checked), CHECK_FROM_ENTRY (General&gt;Outbound ID Length Checked From Entry), FIXED_LENGTH (General&gt;Outbound ID Length Fixed))</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>boolean: allowConsolidationOfPallets (Material Handling&gt;Allow Consolidation of Pallets)</td>
<td>boolean: basedOnMeasurementUnit (Units of Measure&gt;Based on Measurement Unit)</td>
</tr>
<tr>
<td>WSMultiAttributeFulfillmentEnum: pickOnItemAttributes (one of NONE, CORE, CORE_AND_FLEX; Sales/Fulfillment&gt;Pick on item attributes)</td>
<td>ItemmasterWarehouse: itemMasterWarehouses (an entry in the table on the Warehouses tab)</td>
</tr>
<tr>
<td>CustomFields: itemMasterCustomFields (fields in the Custom tab; see Section 2.4, Flex Field Types)</td>
<td>string: preferredVendor (PO/Receiving&gt;Preferred Vendor)</td>
</tr>
<tr>
<td>WSBAStatus: customerStatus (Types&gt;Customer Status)</td>
<td>string: error – contains any error messages that resulted when processing the lookup</td>
</tr>
<tr>
<td>WSBAStatus: ownerStatus (Types&gt;Owner Status)</td>
<td>WSBusinessAffiliate: businessAffiliates – can occur 0 or more times (up to 100); the information about each business affiliate itself. Each WSBusinessAffiliate contains (references are to the Administration&gt;Business Affiliates&gt;View screen):</td>
</tr>
<tr>
<td>WSBAStatus: carrierStatus (Types&gt;Carrier Status)</td>
<td>string: companyName (Name)</td>
</tr>
<tr>
<td>WSBAStatus: brokerStatus (Types&gt;Broker Status)</td>
<td>string: companyCode (Code)</td>
</tr>
<tr>
<td>string: ownerName (the Name of the Business Affiliate Owner, if any, for this business affiliate)</td>
<td>string: ownerCode (the Code of the Business Affiliate Owner, if any, for this business affiliate)</td>
</tr>
<tr>
<td>Address: mainAddress (Main&gt;Address line 1, Address line 2, City, State, Country, Postal code)</td>
<td></td>
</tr>
</tbody>
</table>
2.4 Flex Field Types

SmartTurn flex fields (called "Custom Fields" in the SmartTurn user interface) allow you to customize SmartTurn for business processes that are unique to your operation. The SmartTurn web services flex field types allow you to retrieve custom field data using the web services lookup* operations.

The `CustomFields` type contains one or more `CustomField` elements:

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomFields</td>
<td>CustomField: field (occurs 1 or more times)</td>
</tr>
</tbody>
</table>

The `CustomField` element contains the information about the custom field as it has been configured in SmartTurn. In the SmartTurn user interface, you can find the names, types, and business processes of custom fields under Administration>Custom Fields.

<table>
<thead>
<tr>
<th>CustomField Element</th>
<th>Components (type: name)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

On-Demand WMS (SmartTurn) Web Services - Web Services API
3 Operations

3.1 authenticate

Overview: This web service operation verifies the login information (username and password) of the SmartTurn account that the client (web service invoker) is using.

This operation requires the permission “Web Service Access” under Administration>My Account.

Input message: authenticate

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |

Output message: authenticateResponse

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>string: response</td>
<td>• The response string contains “authentication – success” if the login was successful, or an error message if the login failed.</td>
</tr>
</tbody>
</table>

Notes:

While the authenticate operation can be used as a standalone request/response, the inCredential element must also be embedded in the request of every other operation. For example:

```
<savePurchaseOrder xmlns="http://www.smartturn.com/services/OccamService/purchase-order">
  <inCredential>
```

Conditional value

Choose one based on the particular custom field, then enter the corresponding value:

- CustomFieldStringType: textValue
- double: numberValue
- CustomFieldMemoType: memoValue
- string: listDisplayValue
- YesNoEnum: yesnoValue
- Currency: currencyValue
- Quantity: quantityValue

For example:

```
<tns:listDisplayValue>Summer</tns:listDisplayValue>
```
3.2 authenticateAndGetTenantName

**Overview:** This web service operation verifies the login information (username and password, and optionally particular permissions) of the SmartTurn account that the client (web service invoker) is using. This operation is an alternative to the authenticate operation that additionally validates that the user has the specific permissions passed in as part of the request.

This operation requires the permission “Web Service Access” under Administration>My Account.

**Input message:** authenticateAndGetTenantNameRequest

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required)  
• string: permission (optional; up to 100; the identifier(s) of specific permissions in SmartTurn) |

**Output message:** authenticateAndGetTenantNameResponse

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| WSAuthenticationResult: result | Exactly one WSAuthenticationResult, which contains:  
• boolean: authorized (whether or not the user's credentials are valid, and the user has any of the permissions specified in the inCredential element)  
• string: error (an error message, if any)  
• string: tenantName (Company name)  
• string: warehouseName (the default Warehouse of this user) |

3.3 Purchase Order Operations

3.3.1 savePurchaseOrder

**Overview:** This web service operation saves (creates) purchase orders to SmartTurn. The operation requires a username/password and a collection of one or more purchase orders. A single operation can save (create) 100 purchase orders.
A SmartTurn purchase order contains: (1) purchase order header information; (2) a collection of one or more purchase order items. Each purchase order can contain up to 1000 purchase order items.

This operation requires the permission “Add/Create New” under Purchase Orders.

**Input message:** savePurchaseOrder

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | - string: UserId (required)  
- choice: string: Password (required) |
| PurchaseOrder: inPurchaseOrders | - NumberType: externalNumber (External #)  
- dateTime: date (PO date)  
- Warehouse: warehouse (Warehouse; required only if the warehouse is different from the default warehouse for this login)  
- string: warehouseName (Warehouse)  
- NumberType: number (PO #)  
- int: changeOrderNumber (unused)  
- NameType: vendorName (Vendor>Vendor name)  
- ContactType: vendorContact (Vendor>Vendor contact)  
- PhoneType: vendorContactPhone (Vendor>Vendor phone)  
- Address: vendorAddress (Vendor>Address line 1, Address line 2, City, State, Country, Postal code)  
- NameType: buyerName (Ship To>Company)  
- ContactType: buyerContact (Ship To>Ship to contact)  
- Address: shipTo (Ship To>Address line 1, Address line 2, City, State, Country, Postal code)  
- PurchaseOrderStatusEnum: status (Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)  
- WSSectionStatusEnum: billingStatus (Billing>Billing Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)  
- Affiliate: vendor (contains Vendor>Vendor and Code from Vendor’s business affiliate record)  
- CreateRecord: createRecord (deprecated in favor of createdBy and createdDate)  
- string: createdBy (Created by; only populated on lookup)  
- dateTime: createdDate (Date created; only populated on lookup)  
- LastModifiedRecord: lastModifiedRecord (deprecated in favor of lastModifiedBy and lastModifiedDate)  
- string: lastModifiedBy (not displayed in SmartTurn user interface; only populated on lookup)  
- dateTime: lastModifiedDate (not displayed in SmartTurn user interface; only populated on lookup) |
On-Demand WMS (SmartTurn) Web Services - Web Services API

- **CommentType**: comments (Comments)
- **Currency**: orderedTotalAmount (Total amount; only populated on lookup)
- **PurchaseOrderItem.item** (from the PO Items table; required; 1 or more)
- **OrderTypeEnum.type** (Source; only populated on lookup)
- **IdType.departmentNumber** (Dept #)
- **IdType.groupNumber** (Group #)
- **boolean.partialReceiptsAllowed** (Partial Receipt Allowed)
- **Affiliate.owner** (deprecated)
- **Affiliate.carrier** (deprecated)
- **Affiliate.broker** (deprecated)
- **dateTime.dueDate** (Due Date)
- **NameType.orderedBy** (Ordered by)
- **string.priority** (Priority)
- **NameType.vendorCode** (Code from the Vendor’s business affiliate record)
- **string.ownerName** (Owner)
- **NameType.ownerCode** (Code from the Owner’s business affiliate record)
- **string.carrierName** (Carrier)
- **NameType.carrierCode** (Code from the Carrier’s business affiliate record)
- **string.brokerName** (Broker)
- **NameType.brokerCode** (Code from the Broker’s business affiliate record)
- **CustomFields:poCustomFields** (fields from the Custom tab; see Section 2.4, Flex Field Types)

**Output message**: savePurchaseOrderResponse

**Elements (type: name)**

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| UploadResponse: uploadResponse | • string:externalRefId – The alphanumeric identifier used by the client (web service invoker) to identify the purchase order. This identifier is part of the input message: it is the required element externalNumber in the PurchaseOrder. It is returned as part of the output message.  
• string:systemId – The unique system identifier created by SmartTurn during the operation.  
• ActionStatusType:status – The result of the web service request: SUCCESS or FAILURE.  
• string:error – The error message. Only present if status is not SUCCESS.  
• string:warning – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown. |
Notes:

The `inPurchaseOrders : externalNumber` element is an alphanumeric identifier used by the client (web service invoker) to identify the purchase order. It persists within SmartTurn as the `External #` when the purchase order is saved to SmartTurn, and is returned as part of the `savePurchaseOrderResponse`. It must be unique: no other SmartTurn purchase order can use this identifier. If you attempt to use an existing purchase order’s `externalNumber`, the web service response will give you a corresponding error message. (There is one exception to this: if the purchase order in SmartTurn has been cancelled, another purchase order can be created with the same `External #`.)

The `inPurchaseOrders : <business affiliate>Name` and `<business affiliate>Code` elements must match a valid, active business affiliate of that type. For example, you can create a purchase order by supplying just the `ownerCode` or just the `ownerName`. However, the `Name` and `Code`, if they are both supplied, must match the same valid, active business affiliate of that type and must be the `Name` and `Code` of a valid, active business affiliate of `Type Owner`.

The `inPurchaseOrders : item : itemMasterId` is the `Item #` of the item master (master record for a type of inventory item) within SmartTurn. The item master must already be configured within SmartTurn before the `savePurchaseOrder` web service can be invoked.

The `inPurchaseOrders : item : orderedQuantity` element is of type `Quantity`, described above in Section 2.2, Occam Types. An element of type `Quantity` contains both a `value` (number) and a `unitAbbreviation` (abbreviation for unit of measure). For example:

```
<ns26:orderedQuantity>
  <ns29:value xmlns:ns29="http://www.smartturn.com/services/occamtypes">10.0</ns29:value>
  <ns30:unitAbbreviation xmlns:ns30="http://www.smartturn.com/services/occamtypes">ea</ns30:unitAbbreviation>
</ns26:orderedQuantity>
```

The unit of measure and abbreviation must already be configured with the SmartTurn item master before the `savePurchaseOrder` web service can be invoked.

The `uploadResponse : systemId` element is a unique, system-wide identifier created by SmartTurn (the `PO #`). It is returned in the `savePurchaseOrderResponse`.

### 3.3.1.1 Sample savePurchaseOrder request

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <savePurchaseOrder
      xmlns="http://www.smartturn.com/services/OccamService/purchase-order">
      <inCredential>
        <ns1:UserId xmlns:ns1="http://www.smartturn.com/services/occamtypes">testuser</ns1:UserId>
        <ns2:Password xmlns:ns2="http://www.smartturn.com/services/occamtypes">t3stp-d</ns2:Password>
      </inCredential>
    </savePurchaseOrder>
  </soapenv:Body>
</soapenv:Envelope>
```
</inCredential>

</inPurchaseOrders>

<ns3:externalNumber xmlns:ns3="http://www.smartturn.com/services/purchase-order-types">Ext - 0003</ns3:externalNumber>

<ns4:date xmlns:ns4="http://www.smartturn.com/services/purchase-order-types">2007-03-20T00:01:50.664Z</ns4:date>

<ns5:vendorName xmlns:ns5="http://www.smartturn.com/services/purchase-order-types">v name</ns5:vendorName>

<ns6:vendorContact xmlns:ns6="http://www.smartturn.com/services/purchase-order-types">v contact</ns6:vendorContact>

<ns7:vendorContactPhone xmlns:ns7="http://www.smartturn.com/services/purchase-order-types">000-000-0000</ns7:vendorContactPhone>

<ns8:vendorAddress xmlns:ns8="http://www.smartturn.com/services/purchase-order-types">
  <ns9:addressLine1 xmlns:ns9="http://www.smartturn.com/services/occamtypes">1000</ns9:addressLine1>
  <ns10:addressLine2 xmlns:ns10="http://www.smartturn.com/services/occamtypes">broadway</ns10:addressLine2>
</ns8:vendorAddress>

<ns15:buyerContact xmlns:ns15="http://www.smartturn.com/services/purchase-order-types">
  <ns16:shipTo xmlns:ns16="http://www.smartturn.com/services/purchase-order-types">
    <ns17:addressLine1 xmlns:ns17="http://www.smartturn.com/services/occamtypes">1000</ns17:addressLine1>
    <ns18:addressLine2 xmlns:ns18="http://www.smartturn.com/services/occamtypes">broadway</ns18:addressLine2>
  </ns16:shipTo>
</ns15:buyerContact>
</ns16:shipTo>
<ns24:comments xmlns:ns25="http://www.smartturn.com/services/purchase-order-types">this is comments</ns25:comments>
<ns25:item xmlns:ns26="http://www.smartturn.com/services/purchase-order-types">
<ns25:description>description</ns26:description>
<ns25:details>details</ns26:details>
<ns25:dueDate>2007-03-20T00:01:50.664Z</ns26:dueDate>
<ns25:price>
<ns26:value xmlns:ns27="http://www.smartturn.com/services/occamtypes">1.0</ns27:value>
</ns25:price>
<ns25:group>All</ns26:group>
<ns25:contractInfo>contract info</ns26:contractInfo>
<ns25:orderedQuantity>
<ns28:value xmlns:ns29="http://www.smartturn.com/services/occamtypes">10.0</ns29:value>
</ns25:orderedQuantity>
<ns25:itemMasterId>BB-10</ns26:itemMasterId>
<ns25:receivingInstructions>recv instructions</ns26:receivingInstructions>
<ns25:vendorItemId>v itm id</ns26:vendorItemId>
<ns25:manufacturerItemId>manf item id</ns26:manufacturerItemId>
<ns25:manufacturerId>manf id</ns26:manufacturerId>
</ns25:item>
</inPurchaseOrders>
</savePurchaseOrder>
</soapenv:Body>
</soapenv:Envelope>
3.3.1.2 Sample savePurchaseOrder response

```xml
<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <savePurchaseOrderResponse
     xmlns="http://www.smartturn.com/services/OccamService/purchase-order">
      <uploadResponse>
        <ns1:externalRefId
         xmlns:ns1="http://www.smartturn.com/services/occamtypes">Ext -
0003</ns1:externalRefId>
        <ns2:status
         xmlns:ns2="http://www.smartturn.com/services/occamtypes">SUCCESS</ns2:status>
        <ns3:systemId
         xmlns:ns3="http://www.smartturn.com/services/occamtypes">PO-
000001</ns3:systemId>
      </uploadResponse>
    </savePurchaseOrderResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

3.3.2 lookUpPurchaseOrders

**Overview:** This web service operation retrieves purchase order documents from SmartTurn based on the criteria used. This operation requires a username/password and optionally a criteria for selecting purchase order documents. This operation does NOT guarantee that the orders retrieved are unchanged. That is, the orders could still be altered within SmartTurn after they are retrieved through this operation.

This operation requires the permission "View Detail" under *Purchase Orders.*

**Input message:** lookUpPurchaseOrder

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
   • choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria is passed, the first 100 purchase orders available in the system are returned, sorted by purchase order number. For this operation, the Criteria can include any PurchaseOrder elements except item, warehouse (but warehouseName can be included), vendorAddress, shipTo, vendor, comments, owner, broker, carrier, and poCustomFields.  
   • See Section 1.5, Web Service Criteria, for information on criteria. |
### Output message: `lookUpPurchaseOrderResponse`

#### Elements (name: type)

<table>
<thead>
<tr>
<th>Element</th>
<th>Components</th>
</tr>
</thead>
</table>
| WSPurchaseOrderLookupResult: result | • resultMetadata — The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.1 Purchase order lookup types, for more information on lookup results.  
• error — The error message(s) that resulted when processing the lookup criteria.  
• purchaseOrders — The collection of PurchaseOrder elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |

### 3.3.3 `cancelPurchaseOrder`

#### Overview:
This web service operation cancels the selected purchase order document from SmartTurn. This operation requires a username/password and either the SmartTurn purchase order number (PO #) or the external reference number (External #), or both, and optionally, the warehouse of the purchase order. This operation puts the selected purchase order into a Cancelled status. This operation fails if the selected purchase order is not already in Saved status.

This operation requires the permission “Cancel” under Purchase Orders.

#### Input message: `cancelPurchaseOrder`

#### Elements (name: type)

<table>
<thead>
<tr>
<th>OccamPasswordCredential: inCredential</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>string: UserId (required)</td>
<td></td>
</tr>
<tr>
<td>choice: string: Password (required)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IdType: externalNumber</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>The External # for the purchase order in SmartTurn. Either externalNumber or number must be specified in the input message. If both are specified, they must refer to the same purchase order, or else there is an error and the operation will not complete.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IdType: number</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>The PO # for the purchase order in SmartTurn. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same purchase order, or else there is an error and the operation will not complete.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>string: warehouseName</th>
<th>Notes</th>
</tr>
</thead>
</table>
| The Warehouse for the purchase order in SmartTurn. The warehouseName is optional, but is required in two business situations:  
• If the UserId in the inCredential for this request has ‘All Warehouses’ as the default warehouse context, then the warehouseName is required.  
• If the UserId in the inCredential for this request as a particular warehouse as the default warehouse context, but has access to a second warehouse and needs to cancel a purchase order from the second warehouse, then the |
On-Demand WMS (SmartTurn) Web Services - Web Services API

Output message: `cancelPurchaseOrderResponse`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| UploadResponse: uploadResponse | • `string': externalRefId` — This identifier is optionally part of the input message; it is the element `externalNumber`. It is returned as part of the output message.  
• `string': systemId` — This identifier is optionally part of the input message; it is the element `number`. It is returned as part of the output message.  
• `ActionStatusType': status` — The result of the web service request: `SUCCESS` or `FAILURE`.  
• `string': error` — The error message. Only present if `status` is not `SUCCESS`.  
• `string': warning` — Warning messages, if any. The web service request will be successfully processed even if a warning is thrown. |

### 3.3.4 `overwritePurchaseOrder`

**Overview:** This web service operation overwrites all information in the selected purchase order documents from SmartTurn. A single operation can overwrite all information in up to 100 purchase orders. This operation requires a username/password and either the SmartTurn purchase order number (PO #) or the external reference number (External #), or both, for each purchase order. This operation fails if the selected purchase orders are not already in Saved status.

A SmartTurn purchase order contains: (1) purchase order header information; (2) a collection of one or more purchase order items. Each purchase order can contain up to 1000 purchase order items.

This operation requires the permission “Edit” under *Purchase Orders*.

**Input message:** `overwritePurchaseOrder`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • `string': UserId` (required)  
• `choice': string': Password` (required) |
| PurchaseOrder: inPurchaseOrders | • `NumberType': externalNumber` (External #)  
• `dateTime': date` (PO date)  
• `Warehouse': warehouse` (Warehouse; required only if the warehouse is different from the default warehouse for this login)  
• `NumberType': number` (PO #)  
• `int': changeOrderNumber` (unused)  
• `NameType': vendorName` (Vendor>Vendor name)  
• `ContactType': vendorContact` (Vendor>Vendor contact) |
- **PhoneType**: `vendorContactPhone` (Vendor>`Vendor phone`
- **Address**: `vendorAddress` (Vendor>`Address line 1, Address line 2, City, State, Country, Postal code`
- **NameType**: `buyerName` (Ship To>`Company`
- **ContactType**: `buyerContact` (Ship To>`Ship to contact`
- **Address**: `shipTo` (Ship To>`Address line 1, Address line 2, City, State, Country, Postal code`
- **PurchaseOrderStatusEnum**: `status` (Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)
- **WSSectionStatusEnum**: `billingStatus` (Billing>`Billing Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)
- **Affiliate**: `vendor` (contains Vendor>`Vendor and Code from Vendor's business affiliate record)
- **CreateRecord**: `createRecord` (deprecated in favor of `createdBy` and `createdDate`
- **string**: `createdBy` (Created by; only populated on lookup)
- **dateTime**: `createdDate` (Date created; only populated on lookup)
- **LastModifiedRecord**: `lastModifiedRecord` (deprecated in favor of `lastModifiedBy` and `lastModifiedDate`
- **string**: `lastModifiedBy` (not displayed in SmartTurn user interface; only populated on lookup)
- **dateTime**: `lastModifiedDate` (not displayed in SmartTurn user interface; only populated on lookup)
- **CommentType**: `comments` (Comments)
- **Currency**: `orderedTotalAmount` (Total amount; only populated on lookup)
- **PurchaseOrderItem**: `item` (from the PO Items table; required; 1 or more)
- **OrderTypeEnum**: `type` (Source; only populated on lookup)
- **IdType**: `departmentNumber` (Dept #)
- **IdType**: `groupNumber` (Group #)
- **boolean**: `partialReceiptsAllowed` (Partial Receipt Allowed)
- **Affiliate**: `owner` (deprecated)
- **Affiliate**: `carrier` (deprecated)
- **Affiliate**: `broker` (deprecated)
- **dateTime**: `dueDate` (Due Date)
- **NameType**: `orderedBy` (Ordered by)
- **string**: `priority` (Priority)
- **NameType**: `vendorCode` (Code from the Vendor's business affiliate record)
- **string**: `ownerName` (Owner)
- **NameType**: `ownerCode` (Code from the Owner's business
### 3.4 Receipt Operations

#### 3.4.1 getReceipts

**Overview:** This web service operation retrieves receipts from SmartTurn. It is deprecated. Please use `lookupReceipts` instead.

#### 3.4.2 lookupReceipts

**Overview:** This web service operation retrieves receipts from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting receipts.

This operation requires the permission “View Detail” under `Receiving>All Receipts`.

**Input message:** `lookupReceipts`

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential: inCredential</td>
<td></td>
</tr>
<tr>
<td>• string: UserId (required)</td>
<td></td>
</tr>
<tr>
<td>• choice: string: Password (required)</td>
<td></td>
</tr>
</tbody>
</table>
Criteria: inCriteria

- If no Criteria is passed, the first 1000 receipts available in the system are returned, sorted by receipt number.
- See Section 1.5, Web Service Criteria, for information on criteria.

Output message: lookupReceiptsResponse

Elements (type: name) | Components
--- | ---
ReceiptLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.2 Receipt lookup types, for more information on lookup results.
• error – The error message(s) that resulted when processing the lookup criteria.
• receipts – The collection of Receipt elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.

3.5 Inventory Operations

3.5.1 lookUpAdjustments

Overview: This web service operation retrieves adjustments from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting adjustments.

This operation requires the permission “Adjust Inventory”, under Inventory>Inventory Management.

Input message: lookUpAdjustments

Elements (type: name) | Notes
--- | ---
OccamPasswordCredential: inCredential | • string: UserId (required)
• choice: string: Password (required)

Criteria: inCriteria | • If no Criteria is passed, the first 1000 adjustments available in the system are returned. For this operation, the Criteria can include any WSAdjustment elements except items, comment, and warehouse (warehouse is deprecated; you can query by its replacement, warehouseName).
• See Section 1.5, Web Service Criteria, for information on criteria.

Output message: lookUpAdjustmentsResponse

Elements (type: name) | Components
3.5.2 lookUpCycleCounts

Overview: This web service operation retrieves cycle count documents from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting cycle count documents.

This operation requires the permission "Cycle Count View", under Inventory>Inventory Management.

Input message: lookUpCycleCounts

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria is passed, the first 1000 cycle counts available in the system are returned. For this operation, the Criteria can include any of the WSCycleCount elements, except for the comments and any of the information contained in the items.  
• See Section 1.5, Web Service Criteria, for information on criteria. |

Output message: lookUpCycleCountsResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| CycleCountsLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentIndex, recordCountInThisPage, and totalPages. See Section 2.3.3 Inventory transaction lookup types, for more information on lookup results.  
• error – The error message(s) that resulted when processing the lookup criteria.  
• cyclecounts – The collection of WSCycleCount elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |
3.5.3 lookUpBinTransfer

**Overview:** This web service operation retrieves bin transfers from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting bin transfers from across the system.

This operation requires the permission “Transfer Inventory”, under *Inventory>Inventory Management*.

**Input message:** lookUpBinTransfer

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential: inCredential</td>
<td>• string: UserId (required)</td>
</tr>
<tr>
<td></td>
<td>• choice: string: Password (required)</td>
</tr>
<tr>
<td>Criteria: inCriteria</td>
<td>• If no Criteria is passed, the first 1000 bin transfers available in the system are returned. For this operation, the Criteria can include any of the WSBinTransfer elements except for items and comments.</td>
</tr>
<tr>
<td></td>
<td>• See Section 1.5, Web Service Criteria, for information on criteria.</td>
</tr>
</tbody>
</table>

**Output message:** lookUpBinTransferResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSBinTransferLookupResult: result</td>
<td>• resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.3 Inventory transaction lookup types, for more information on lookup results.</td>
</tr>
<tr>
<td></td>
<td>• error – The error message(s) that resulted when processing the lookup criteria.</td>
</tr>
<tr>
<td></td>
<td>• binTransers – The collection of WSBinTransfer elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.</td>
</tr>
</tbody>
</table>

3.5.4 createStorageDocuments

**Overview:** This web service operation creates, saves, and closes storage documents, and computes the billing lines for each storage document. The operation requires a username/password, the inventory manager, and the warehouse. The storage documents are created with a Status of Closed and with billing lines already computed. The document IDs of the new storage documents (ST-nnnnnn) are returned in the web service response. The operation fails if the inventory manager is not set to Use as storage document.

This operation requires the permission “Add Storage”, under *Billing>Storage Documents*.

**Input message:** createStorageDocuments
3.5.5 lookUpTransferShipments

**Overview:** This web service operation retrieves transfer shipments from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting transfer shipments from across the system.

**Input message:** lookUpTransferShipments

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria is passed, the first 1000 transfer shipments available in the system are returned. For this operation, the Criteria can include any of the WSTransferShipment elements except for isPushScenario, shipFromAddress, shipToAddress, transferShipmentItem, transferShipmentAnomalyItem, and comments.  
• See Section 1.5, Web Service Criteria, for information on criteria. |

**Output message:** lookUpTransferShipmentsResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| CreateTransferShipmentsResponse: response | • ActionStatusType: status – The result of the web service request: SUCCESS or FAILURE.  
• string: documentId – the collection of strings (Document #s) of all of the storage documents created in the system on success.  
• string: error or warning – The error message(s) or warning message(s) that resulted when processing the create request. |
### 3.5.6 `lookUpTransferReceipts`

**Overview:** This web service operation retrieves transfer receipts from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting transfer receipts from across the system.

**Input message:** `lookUpTransferReceipts`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential: inCredential</td>
<td>• <code>string</code>: <code>UserId</code> <em>(required)</em></td>
</tr>
<tr>
<td></td>
<td>• <code>choice</code>: <code>string</code>: <code>Password</code> <em>(required)</em></td>
</tr>
<tr>
<td>Criteria: inCriteria</td>
<td>• If no <code>Criteria</code> is passed, the first 1000 transfer receipts available in the system are returned. For this operation, the <code>Criteria</code> can include any of the <code>WSTransferReceipt</code> elements except for <code>shipToAddress</code>, <code>billingAddress</code>, <code>transferReceiptItem</code>, <code>useAsItemReceiptDate</code>, and <code>comments</code>.</td>
</tr>
<tr>
<td></td>
<td>• See Section 1.5, Web Service Criteria, for information on criteria.</td>
</tr>
</tbody>
</table>

**Output message:** `lookUpTransferReceiptsResponse`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSTransferReceiptLookupResult: result</td>
<td>• <code>resultMetadata</code> – The metadata about the result collection, including <code>totalRecordsCount</code>, <code>currentPageIndex</code>, <code>recordCountInThisPage</code>, and <code>totalPages</code>. See Section 2.3.3 Inventory transaction lookup types, for more information on lookup results.</td>
</tr>
<tr>
<td></td>
<td>• <code>error</code> – The error message(s) that resulted when processing the lookup criteria.</td>
</tr>
<tr>
<td></td>
<td>• <code>transferReceipt</code> – The collection of <code>WSTransferReceipt</code> elements that match the input criteria. If the lookup returns no results, this component is not returned, and the <code>resultMetadata</code> indicates a <code>totalRecordsCount</code> of 0.</td>
</tr>
</tbody>
</table>
3.6 Sales Order Operations

3.6.1 saveSalesOrder

Overview: This web service operation saves (creates) sales orders to SmartTurn. The operation requires a username/password and a collection of one or more sales orders. A single operation can save (create) 100 sales orders.

A SmartTurn sales order contains: (1) sales order header information; (2) a collection of one or more sales order items. Each sales order can contain up to 1000 sales order items.

This operation requires the permission “Add/Create New” under Sales Orders.

Input message: saveSalesOrder

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
| | • choice: string: Password (required) |
| SalesOrder: inSalesOrders | • IdType: number (SO #)  
| | • IdType: externalNumber (External #)  
| | • SalesOrderTypeEnum: type (Source; one of INTERNAL or EXTERNAL)  
| | • PositiveInteger: changedOrderNumber (unused)  
| | • dateTime: date (SO date)  
| | • dateTime: dateDue (Due date)  
| | • Warehouse: warehouse (Warehouse; required only if the warehouse is different from the default warehouse for this login)  
| | • string: warehouseName (Warehouse)  
| | • NameType: shipFromName (deprecated)  
| | • ContactType: shipFromContact (deprecated)  
| | • PhoneType: shipFromContactPhone (deprecated)  
| | • Address: shipFrom (deprecated)  
| | • NameType: customerName (Bill To>Bill to customer)  
| | • ContactType: customerContact (Bill To>Bill to contact)  
| | • PhoneType: customerContactPhone (Bill To>Bill to phone)  
| | • Address: customerAddress (Bill To>Address line 1, Address line 2, City, State, Country, Postal Code)  
| | • NameType: salesRep (deprecated)  
| | • boolean: useShipToAsBillAddress (Bill To>Same as Ship To; if true, customer name and customer contact info, if provided, will be overwritten by the ship to info)  
| | • NameType: shipToName (Ship To>Ship to customer)  
| | • ContactType: shipToContact (Ship To>Ship to contact)  
| | • PhoneType: shipToContactPhone (Ship To>Ship to phone)  
| | • Address: ShipTo (Ship To>Address line 1, Address line 2, City,
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State, Country, Postal Code</td>
<td></td>
</tr>
<tr>
<td>Affiliate: customer</td>
<td>(Ship To&gt;Customer)</td>
</tr>
<tr>
<td>SalesOrderStatusEnum: status</td>
<td>(Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)</td>
</tr>
<tr>
<td>SalesOrderItem: item</td>
<td>(from the Items table; required, 1 or more)</td>
</tr>
<tr>
<td>CommentType: comments</td>
<td>(Comments)</td>
</tr>
<tr>
<td>boolean: allowPartialShipment</td>
<td>(Partial Shipment Allowed)</td>
</tr>
<tr>
<td>Affiliate: owner</td>
<td>(deprecated)</td>
</tr>
<tr>
<td>Affiliate: carrier</td>
<td>(deprecated)</td>
</tr>
<tr>
<td>Affiliate: broker</td>
<td>(deprecated)</td>
</tr>
<tr>
<td>NameType: orderedBy</td>
<td>(Ordered by)</td>
</tr>
<tr>
<td>string: priority</td>
<td>(Priority)</td>
</tr>
<tr>
<td>NameType: customerCode</td>
<td>(Code from the Customer's business affiliate record)</td>
</tr>
<tr>
<td>string: ownerName</td>
<td>(Owner)</td>
</tr>
<tr>
<td>NameType: ownerCode</td>
<td>(Code from the Owner's business affiliate record)</td>
</tr>
<tr>
<td>string: carrierName</td>
<td>(Carrier)</td>
</tr>
<tr>
<td>NameType: carrierCode</td>
<td>(Code from the Carrier's business affiliate record)</td>
</tr>
<tr>
<td>string: brokerName</td>
<td>(Broker)</td>
</tr>
<tr>
<td>NameType: brokerCode</td>
<td>(Code from the Broker's business affiliate record)</td>
</tr>
<tr>
<td>dateTime: createdDate</td>
<td>(Date created; only populated on lookup)</td>
</tr>
<tr>
<td>dateTime: lastModifiedDate</td>
<td>(not displayed in SmartTurn user interface; only populated on lookup)</td>
</tr>
<tr>
<td>WSSectionStatusEnum: billingStatus</td>
<td>(Billing&gt;Billing Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)</td>
</tr>
<tr>
<td>IdType: waveNumber</td>
<td>(Wave #; only populated on lookup)</td>
</tr>
<tr>
<td>CustomFields: soCustomFields</td>
<td>(fields from the Custom tab; see Section 2.4, Flex Field Types)</td>
</tr>
</tbody>
</table>

**Output message:** saveSalesOrderResponse

| Elements (type: name) | Components |
UploadResponse:
uploadResponse

- string: externalRefId – The alphanumeric identifier used by the client (web service invoker) to identify the sales order. This identifier is part of the input message: it is the required element externalNumber in the SalesOrder. It is returned as part of the output message.
- string: systemId – The unique system identifier created by SmartTurn during the operation.
- ActionStatusType: status – The result of the web service request: SUCCESS or FAILURE.
- string: error – The error message. Only present if status is not SUCCESS.
- string: warning – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown.

Notes:
The inSalesOrders : externalNumber element is an alphanumeric identifier used by the client (web service invoker) to identify the sales order. It persists within SmartTurn as the External # when the sales order is saved to SmartTurn, and is returned as part of the saveSalesOrderResponse. It must be unique: no other SmartTurn sales order can use this identifier. If you attempt to use an existing sales order's externalNumber, the web service response will give you a corresponding error message. (There is one exception to this: if the sales order in SmartTurn has been cancelled, another sales order can be created with the same External #.)

The inSalesOrders : <business affiliate>Name and <business affiliate>Code elements must match a valid, active business affiliate of that type. For example, you can create a sales order by supplying just the ownerCode or just the ownerName. However, the Name and Code, if they are both supplied, must match the same valid, active business affiliate of that type and must be the Name and Code of a valid, active business affiliate of Type Owner.

The inSalesOrders : shipTo elements specify the customer that the sales order should be shipped to. They correspond to the fields in the Ship To tab on the View Sales Order screen in the SmartTurn user interface. The inSalesOrders : customer elements specify the customer that the sales order should be billed to. They correspond to the fields in the Bill To tab on the View Sales Order screen in the SmartTurn user interface. If these customers should be the same, you can set inSalesOrders : useShipToAsBillAddress to true and leave the customer elements blank.

The inSalesOrders : item : itemMasterId is the Item # of the item master (master record for a type of inventory item) within SmartTurn. The item master must already be configured within SmartTurn before the saveSalesOrder web service can be invoked.

The inSalesOrders : item : orderedQuantity element is of type Quantity, as described in Section 2.2, Occam Types. An element of type Quantity contains both a value (number) and a unitAbbreviation (abbreviation for unit of measure). For example:

```xml
<ns23:orderedQuantity>
  <ns24:value xmlns:ns24="http://www.smartturn.com/services/occamtypes">10.0</ns24:value>
</ns23:orderedQuantity>
```
The unit of measure and abbreviation must already be configured with the SmartTurn item master before the `saveSalesOrder` web service can be invoked.

The `uploadResponse : systemId` element is a unique, system-wide identifier created by SmartTurn (the SO #). It is returned in the `saveSalesOrderResponse`.

### 3.6.1.1 Sample `saveSalesOrder` request

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <saveSalesOrder
      xmlns="http://www.smartturn.com/services/OccamService/sales-order">
      <inCredential>
        <ns1:UserId
          xmlns:ns1="http://www.smartturn.com/services/occamtypes">testuser</ns1:UserId>
        <ns2:Password
          xmlns:ns2="http://www.smartturn.com/services/occamtypes">t3stp-d</ns2:Password>
      </inCredential>
      <inSalesOrders>
        <ns3:externalNumber xmlns:ns3="http://www.smartturn.com/services/sales-order-types">Ext-id-00502</ns3:externalNumber>
        <ns4:type xmlns:ns4="http://www.smartturn.com/services/sales-order-types">EXTERNAL</ns4:type>
        <ns5:date xmlns:ns5="http://www.smartturn.com/services/sales-order-types">2007-03-20T00:04:34.479Z</ns5:date>
        <ns6:dateDue xmlns:ns6="http://www.smartturn.com/services/sales-order-types">2007-03-20T00:04:34.479Z</ns6:dateDue>
        <ns7:customerName xmlns:ns7="http://www.smartturn.com/services/sales-order-types">customer name</ns7:customerName>
        <ns8:customerContact
          xmlns:ns8="http://www.smartturn.com/services/sales-order-types">customer contact</ns8:customerContact>
        <ns9:customerContactPhone
          xmlns:ns9="http://www.smartturn.com/services/sales-order-types">customer phone</ns9:customerContactPhone>
        <ns10:customerAddress
          xmlns:ns10="http://www.smartturn.com/services/sales-order-types">
          <ns11:addressLine1
            xmlns:ns11="http://www.smartturn.com/services/occamtypes">1000</ns11:addressLine1>
          <ns12:addressLine2
            xmlns:ns12="http://www.smartturn.com/services/occamtypes">broadway</ns12:addressLine2>
          <ns13:city
        </ns10:customerAddress>
      </inSalesOrders>
    </saveSalesOrder>
  </soapenv:Body>
</soapenv:Envelope>
```
<ns14:state
xmlns:ns14="http://www.smartturn.com/services/occamtypes">California</ns14:state>

<ns15:country
xmlns:ns15="http://www.smartturn.com/services/occamtypes">USA</ns15:country>

<ns16:postalCode

</ns10:customerAddress>

<ns17:salesRep
xmlns:ns17="http://www.smartturn.com/services/sales-order-types">sales rep</ns17:salesRep>

<ns18:useShipToAsBillAddress
xmlns:ns18="http://www.smartturn.com/services/sales-order-types">false</ns18:useShipToAsBillAddress>

<ns19:shipToName
xmlns:ns19="http://www.smartturn.com/services/sales-order-types">ship to name</ns19:shipToName>

<ns20:shipToContact
xmlns:ns20="http://www.smartturn.com/services/sales-order-types">ship to contact</ns20:shipToContact>

<ns21:shipToContactPhone
xmlns:ns21="http://www.smartturn.com/services/sales-order-types">ship to contact phone</ns21:shipToContactPhone>

<ns22:status
xmlns:ns22="http://www.smartturn.com/services/sales-order-types">NEW</ns22:status>

<ns23:item
xmlns:ns23="http://www.smartturn.com/services/sales-order-types">
  <ns23:itemMasterId>BB-10</ns23:itemMasterId>
  <ns23:description>this is a desc</ns23:description>
  <ns23:details>these are details</ns23:details>
  <ns23:orderedQuantity>
    <ns26:value
xmlns:ns26="http://www.smartturn.com/services/occamtypes">10.0</ns26:value>
    <ns27:unitAbbreviation
xmlns:ns27="http://www.smartturn.com/services/occamtypes">ea</ns27:unitAbbreviation>
  </ns23:orderedQuantity>
  <ns23:customerRequestDate>2007-03-20T00:04:34.479Z</ns23:customerRequestDate>
  <ns23:price>
    <ns28:value
xmlns:ns28="http://www.smartturn.com/services/occamtypes">100.0</ns28:value>
    <ns29:type
xmlns:ns29="http://www.smartturn.com/services/occamtypes">$</ns29:type>
  </ns23:price>
  <ns23:vendorItemId>vendor item id</ns23:vendorItemId>
  <ns23:manufacturerItemId>manf item id</ns23:manufacturerItemId>
</ns23:item>
3.6.1.2 Sample saveSalesOrder response

<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <saveSalesOrderResponse
      xmlns="http://www.smartturn.com/services/OccamService/sales-order">
      <uploadResponse>
        <ns1:externalRefId
          xmlns:ns1="http://www.smartturn.com/services/occamtypes">Ext-id-00502</ns1:externalRefId>
        <ns2:status
          xmlns:ns2="http://www.smartturn.com/services/occamtypes">SUCCESS</ns2:status>
        <ns3:systemId
          xmlns:ns3="http://www.smartturn.com/services/occamtypes">SO-000001</ns3:systemId>
      </uploadResponse>
    </saveSalesOrderResponse>
  </soapenv:Body>
</soapenv:Envelope>

3.6.2 lookUpSalesOrders

Overview: This web service operation retrieves sales order documents from SmartTurn based on
the criteria used. This operation requires a username/password and optionally a criteria for
selecting sales order documents. This operation does NOT guarantee that the orders retrieved
are unchanged. That is, the orders could still be altered within SmartTurn after they are retrieved
through this operation.

This operation requires the permission “View Detail” under Sales Orders.

Input message: lookUpSalesOrder

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential:</td>
<td>• string: UserId (required)</td>
</tr>
</tbody>
</table>
On-Demand WMS (SmartTurn) Web Services - Web Services API

inCredential

| choice: string: Password (required) |

Criteria: inCriteria

- If no Criteria is passed, the first 100 sales orders available in the system are returned, sorted by sales order number. For this operation, the Criteria can include any SalesOrder elements except allowPartialShipment, broker, carrier, comments, customer, customerAddress, item, owner, shipTo, warehouse (but warehouseName can be included), and soCustomFields.
- See Section 1.5, Web Service Criteria, for information on criteria.

Output message: lookUpSalesOrderResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSsalesOrderLookupResult: result</td>
<td></td>
</tr>
<tr>
<td>• resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.4 Sales order lookup types, for more information on lookup results.</td>
<td></td>
</tr>
<tr>
<td>• error – The error message(s) that resulted when processing the lookup criteria.</td>
<td></td>
</tr>
<tr>
<td>• salesOrders – The collection of SalesOrder elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.</td>
<td></td>
</tr>
</tbody>
</table>

3.6.3 cancelSalesOrder

**Overview:** This web service operation cancels the selected sales order document from SmartTurn. This operation requires a username/password and either the SmartTurn sales order number (SO #) or the external reference number (External #), or both. This operation puts the selected sales order into a Cancelled status. This operation fails if the selected sales order is not already in Saved status.

This operation requires the permission “Cancel” under Sales Orders.

**Input message:** cancelSalesOrder

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential: inCredential</td>
<td>• string: UserId (required)</td>
</tr>
<tr>
<td>• choice: string: Password (required)</td>
<td></td>
</tr>
<tr>
<td>IdType: externalNumber</td>
<td>• The External # for the sales order in SmartTurn. Either externalNumber or number must be specified in the input message. If both are specified, they must refer to the same sales order, or else there is an error and the operation will not complete.</td>
</tr>
<tr>
<td>IdType: number</td>
<td>• The SO # for the sales order in SmartTurn. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same sales order, or else there is an error and the operation will not complete.</td>
</tr>
</tbody>
</table>
Output message: `cancelSalesOrderResponse`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| UploadResponse: uploadResponse | • string: `externalRefId` — This identifier is optionally part of the input message: it is the element `externalNumber`. It is returned as part of the output message.  
• string: `systemId` — This identifier is optionally part of the input message: it is the element `number`. It is returned as part of the output message.  
• `ActionStatusType: status` — The result of the web service request: `SUCCESS` or `FAILURE`.  
• string: `error` — The error message. Only present if status is not `SUCCESS`.  
• string: `warning` — Warning messages, if any. The web service request will be successfully processed even if a warning is thrown. |

### 3.6.4 `overwriteSalesOrder`

**Overview:** This web service operation overwrites all information in the selected sales order documents from SmartTurn. A single operation can overwrite all information in up to 100 sales orders. This operation requires a username/password and either the SmartTurn sales order number (`SO #`) or the external reference number (`External #`), or both, for each sales order. This operation fails if the selected sales orders are not already in `Saved` status.

A SmartTurn sales order contains: (1) sales order header information; (2) a collection of one or more sales order items. Each sales order can contain up to 1000 sales order items.

This operation requires the permission “Edit” under *Sales Orders*.

**Input message:** `overwriteSalesOrder`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: `UserId` *(required)*  
• choice: string: `Password` *(required)* |
| SalesOrder: inSalesOrders | • `IdType: number` *(SO #)*  
• `IdType: externalNumber` *(External #)*  
• `SalesOrderTypeEnum: type` *(Source; one of INTERNAL or EXTERNAL)*  
• `PositiveInteger: changedOrderNumber` *(unused)*  
• `dateTime: date` *(SO date)*  
• `dateTime: dateDue` *(Due date)*  
• `Warehouse: warehouse` *(Warehouse; required only if the warehouse is different from the default warehouse for this login)*  
• `NameType: shipFromName` *(deprecated)*  
• `ContactType: shipFromContact` *(deprecated)*  
• `PhoneType: shipFromContactPhone` *(deprecated)* |
- **Address**: `shipFrom` (deprecated)
- **NameType**: `customerName` *(Bill To > Bill to customer)*
- **ContactType**: `customerContact` *(Bill To > Bill to contact)*
- **PhoneType**: `customerContactPhone` *(Bill To > Bill to phone)*
- **Address**: `customerAddress` *(Bill To > Address line 1, Address line 2, City, State, Country, Postal Code)*
- **NameType**: `salesRep` (deprecated)
- **boolean**: `useShipToAsBillAddress` *(Bill To > Same as Ship To; if true, customer name and customer contact info, if provided, will be overwritten by the ship to info)*
- **NameType**: `shipToName` *(Ship To > Ship to customer)*
- **ContactType**: `shipToContact` *(Ship To > Ship to contact)*
- **PhoneType**: `shipToContactPhone` *(Ship To > Ship to phone)*
- **Address**: `ShipTo` *(Ship To > Address line 1, Address line 2, City, State, Country, Postal Code)*
- **Affiliate**: `customer` *(Ship To > Customer)*
- **SalesOrderStatusEnum**: `status` *(Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)*
- **SalesOrderItem**: `item` *(from the Items table; required, 1 or more)*
- **CommentType**: `comments` *(Comments)*
- **boolean**: `allowPartialShipment` *(Partial Shipment Allowed)*
- **Affiliate**: `owner` (deprecated)
- **Affiliate**: `carrier` (deprecated)
- **Affiliate**: `broker` (deprecated)
- **NameType**: `orderedBy` *(Ordered by)*
- **string**: `priority` *(Priority)*
- **NameType**: `customerCode` *(Code from the Customer’s business affiliate record)*
- **string**: `ownerName` *(Owner)*
- **NameType**: `ownerCode` *(Code from the Owner’s business affiliate record)*
- **string**: `carrierName` *(Carrier)*
- **NameType**: `carrierCode` *(Code from the Carrier’s business affiliate record)*
- **string**: `brokerName` *(Broker)*
- **NameType**: `brokerCode` *(Code from the Broker’s business affiliate record)*
- **dateTime**: `createdDate` *(Date created; only populated on lookup)*
- **dateTime**: `lastModifiedDate` *(not displayed in SmartTurn user interface; only populated on lookup)*
- **WSSectionStatusEnum**: `billingStatus` *(Billing > Billing Status; only populated on lookup; one of NEW (deprecated), SAVED, APPROVED, IN_PROGRESS, CLOSED, COMMITTED, CONSUMED, CANCELLED)*
Output message: `overwriteSalesOrderResponse`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| UploadResponse: uploadResponse | • `string`: `externalRefId` — This identifier is optionally part of the input message: it is the element `externalNumber`. It is returned as part of the output message.  
• `string`: `systemId` — This identifier is optionally part of the input message: it is the element `number`. It is returned as part of the output message.  
• `ActionStatusType`: `status` — The result of the web service request: `SUCCESS` or `FAILURE`.  
• `string`: `error` — The error message. Only present if `status` is not `SUCCESS`.  
• `string`: `warning` — Warning messages, if any. The web service request will be successfully processed even if a warning is thrown. |

### 3.7 Shipment Operations

#### 3.7.1 `getShipments`

**Overview:** This web service operation retrieves shipments from SmartTurn. It is deprecated. Please use `lookupShipments` instead.

#### 3.7.2 `lookupShipments`

**Overview:** This web service operation retrieves shipments from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting shipments.

This operation requires the permission “View Detail”, under `Fulfillment>All Shipments`.

**Input message:** `lookupShipments`

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • `string`: `UserId` *(required)*  
• `choice`: `string`: `Password` *(required)* |
Criteria: inCriteria

- If no Criteria is passed, the first 1000 shipments available in the system are returned. For this operation, the Criteria can include any Shipment elements except for item, billToAddress, billToCustomerContact, comments, createRecord, customer, parcelAddress, parcelTrackingURL, shipFrom, shipTo, shipmentCustomFields, warehouse (but warehouseName can be included), shippingContactEmail, tax, hasVoidedItem, totalCube, totalDimWeight.
- See Section 1.5, Web Service Criteria, for information on criteria.

Output message: lookupShipmentsResponse

Elements (type: name) | Components
--- | ---
ShipmentLookupResult: result | - resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.5 Shipment lookup types, for more information on lookup results.
- error – The error message(s) that resulted when processing the lookup criteria.
- shipments – The collection of Shipment elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.

3.7.3 updateParcelPostInfo

Overview: This web service operation updates an existing SmartTurn shipment with parcel post information. On success, it sets the shipment’s Parcel Post Status field to Posted. The SmartTurn shipment must have a status of Closed in order for this operation to succeed. The operation requires a username/password and one or more WSParcelPostInfo elements.

This operation requires the permission “Parcel Post Shipment”, under Fulfillment>Parcel Post.

Input message: updateParcelPostInfo

Elements (name: type) | Components
--- | ---
OccamPasswordCredential: inCredential | - string: UserId (required)
- choice: string: Password (required)

WSParcelPostInfo: inParcelPostInfo | - IdType: shipmentId – This element must match a valid SmartTurn shipment ID (Shipment #) of a shipment that is shipped (status Closed).
- DescriptionType: parcelCompanyName (Parcel Post>Parcel Company)
- DescriptionType: parcelContact (Parcel Post>Parcel Contact)
- DescriptionType: parcelContactPhone (Parcel Post>Parcel Phone)
- Address: parcelAddress (Parcel Post>Address line 1, Address)
On-Demand WMS (SmartTurn) Web Services - Web Services API

OUTPUT MESSAGE: updateParcelPostInfoResponse

RETURN VALUE ELEMENTS:

<table>
<thead>
<tr>
<th>uploadResponse: uploadResponse</th>
<th>string: externalRefId – The alphanumeric identifier used by the client (web service invoker) to identify the shipment being updated.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>status – The result of the web service request: SUCCESS or FAILURE.</td>
</tr>
<tr>
<td></td>
<td>error – The error message. Only present if status is not SUCCESS.</td>
</tr>
<tr>
<td></td>
<td>warning – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown.</td>
</tr>
</tbody>
</table>

3.8 Wave Fulfillment Operations

3.8.1 lookupPickTickets

Overview: This web service operation retrieves pick tickets from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting pick tickets from across the system.

This operation requires the permission “View Pick Ticket Detail”, under Wave Fulfillment>Pick Tickets.

Input message: lookupPickTickets
Elements (name: type) | Notes
---|---
OccamPasswordCredential: inCredential | • string: UserId (required)
• choice: string: Password (required)
Criteria: inCriteria | • If no Criteria is passed, then information for the first 1000 pick tickets available in the system are returned. For this operation, the Criteria can include any WSPickTicket elements except comments, pickTicketItem, and pickTicketAnomalyItem.
• See Section 1.5, Web Service Criteria, for information on criteria.

Output message: lookupPickTicketsResponse

Elements (name: type) | Components
---|---
WSPickTicketLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.6 Wave fulfillment lookup types, for more information on lookup results.
• error – The error message(s) that resulted when processing the lookup criteria.
• items – The collection of WSPickTicket elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.

3.8.2 lookUpBOLs

Overview: This web service operation retrieves bills of lading from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting bills of lading from across the system.

This operation requires the permission “View Bill of Lading Detail”, under Wave Fulfillment>Bills of Lading.

Input message: lookupBOLs

Elements (name: type) | Notes
---|---
OccamPasswordCredential: inCredential | • string: UserId (required)
• choice: string: Password (required)
Criteria: inCriteria | • If no Criteria is passed, then information for the first 1000 bills of lading available in the system are returned. For this operation, the Criteria can include any WSBOL elements except comments, shipToAddress, shipFromAddress, remitToAddress, remitToName, and bolPackage.
• See Section 1.5, Web Service Criteria, for information on criteria.

Output message: lookupBOLsResponse
Elements (name: type) | Components
---|---
WSBOLLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.6 Wave fulfillment lookup types, for more information on lookup results.
• error – The error message(s) that resulted when processing the lookup criteria.
• items – The collection of WSBOL elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.

3.8.3 closeDocuments

**Overview:** This web service operation closes the selected document from SmartTurn. This operation requires a username/password and identifying information for the document: the Document Type (BOL); the Document # (BOL #) or External #, or both; and optionally, the warehouse of the document. This operation puts the selected document into a Closed status. Any side effects of the status change are also executed.

As of Release 1.6, only bills of lading (Document #: BOL-nnnn) documents can be closed through this web service.

For bills of lading, calling this operation means that all Ready packages are automatically Closed, and then all inventory in Closed packages is decremented from the quantity on hand and leaves the warehouse. This operation fails if the selected document is not already in In Progress status, or if there are any packages that are not yet Ready or Closed.

**Input message:** closeDocuments

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| WSDocumentIdentificationWithType: documentIdentification | • WSDocumentType: documentType - The kind of document being closed. For bills of lading, use string: 'BOL'.  
• IdType: number - The document number for the document being closed. For bills of lading, this is the BOL # in SmartTurn. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete.  
• IdType: externalNumber - The External # for the document being closed. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete. |
### 3.8.4 shipClosedBOLPackages

**Overview:** The closeDocuments web service operation closes the document in a “happy path” scenario. For bills of lading (Document #: BOL-nnnn), this means closing all packages with a Status of Ready or Closed, and performing associated side effects, such as decrementing the inventory on hand.

Sometimes, business rules will require a little more specificity. For bills of lading, you can use the shipClosedBOLPackages operation. Calling this operation only ships the packages that are already Closed. This means:

- Any packages with a Status of Ready become Cancelled. The inventory from the Cancelled packages is “dropped” at the Destination Location (packing station) for this bill of lading.
- Any packages with a Status of Closed are shipped out: all inventory in Closed packages is decremented from the quantity on hand and leaves the warehouse.

This operation fails if the selected document is not already in In Progress status, or if there are any packages that are not yet Ready or Closed.

**Input message:** shipClosedBOLPackages
### Elements (name: type)

<table>
<thead>
<tr>
<th>OccamPasswordCredential: inCredential</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>• string: UserId (required)</td>
<td></td>
</tr>
<tr>
<td>• choice: string: Password (required)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WSDocumentIdentification: bolIdentification</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>• IdType: number: The document number for the document being closed. For bills of lading, this is the BOL # in SmartTurn. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete.</td>
<td></td>
</tr>
<tr>
<td>• IdType: externalNumber: The External # for the document being closed. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete.</td>
<td></td>
</tr>
<tr>
<td>• string: warehouseName: The Warehouse for the document in SmartTurn. The warehouseName is optional, but is required in two business situations:</td>
<td></td>
</tr>
<tr>
<td>• If the UserId in the inCredential for this request has ‘All Warehouses’ as the default warehouse context, then the warehouseName is required.</td>
<td></td>
</tr>
<tr>
<td>• If the UserId in the inCredential for this request as a particular warehouse as the default warehouse context, but has access to a second warehouse and needs to cancel a document from the second warehouse, then the warehouseName (name of the second warehouse) is required.</td>
<td></td>
</tr>
</tbody>
</table>

### Output message: shipClosedBOLPackagesResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>UploadResponse: uploadResponse</td>
<td>• string: externalRefId – This identifier is optionally part of the input message: it is the element externalNumber. It is returned as part of the output message.</td>
</tr>
<tr>
<td></td>
<td>• string: systemId – This identifier is optionally part of the input message: it is the element number. It is returned as part of the output message.</td>
</tr>
<tr>
<td></td>
<td>• ActionStatusType: status – The result of the web service request: SUCCESS or FAILURE.</td>
</tr>
<tr>
<td></td>
<td>• string: error – The error message. Only present if status is not SUCCESS.</td>
</tr>
<tr>
<td></td>
<td>• string: warning – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown.</td>
</tr>
</tbody>
</table>

### 3.8.5 commitDocuments

**Overview:** This web service operation commits the selected document from SmartTurn. This operation requires a username/password, the type of document being committed, and an
identifier for the document: the Document # (BOL #) or External #, or both, and optionally, the warehouse of the document. This operation puts the selected document into a Committed status. Any side effects of the status change are also executed. For bills of lading, this means that no additional edits can be made to header fields or packages fields. This operation fails if the selected document is not already in Closed status.

**Input message:** commitDocuments

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| WSDocumentTypeEnum: documentType | • string: 'BOL' (required) |
| WSDocumentIdentificationWithType: documentIdentification | • IdType: number: The document number for the document being committed. For bills of lading, this is the BOL # in SmartTurn. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete.  
• IdType: externalNumber: The External # for the document being committed. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete.  
• string: warehouseName: The Warehouse for the document in SmartTurn. The warehouseName is optional, but is required in two business situations:  
  • If the UserId in the inCredential for this request has ‘All Warehouses’ as the default warehouse context, then the warehouseName is required.  
  • If the UserId in the inCredential for this request as a particular warehouse as the default warehouse context, but has access to a second warehouse and needs to cancel a document from the second warehouse, then the warehouseName (name of the second warehouse) is required. |

**Output message:** commitDocumentsResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
UploadResponse:
uploadResponse

- string: externalRefId – This identifier is optionally part of the input message: it is the element externalNumber. It is returned as part of the output message.
- string: systemId – This identifier is optionally part of the input message: it is the element number. It is returned as part of the output message.
- ActionStatusType: status – The result of the web service request: SUCCESS or FAILURE.
- string: error – The error message. Only present if status is not SUCCESS.
- string: warning – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown.

3.8.6 consumeDocuments

Overview: This web service operation consumes the selected document from SmartTurn. This operation requires a username/password, the type of document being closed, and an identifier of for the document: the Document # (BOL #) or External #, or both, and optionally, the warehouse of the document. This operation puts the selected document into a Consumed status. This operation fails if the selected document is not already in Comitted status.

Input message: consumeDocuments

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential: inCredential</td>
<td>• string: UserId (required)</td>
</tr>
<tr>
<td></td>
<td>• choice: string: Password (required)</td>
</tr>
<tr>
<td>WSDocumentTypeEnum: documentType</td>
<td>• string: 'BOL' (required)</td>
</tr>
<tr>
<td>WSDocumentIdentificationWithType: documentIdentification</td>
<td>• IdType: number: The document number for the document being consumed. For bills of lading, this is the BOL # in SmartTurn. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete.</td>
</tr>
<tr>
<td></td>
<td>• IdType: externalNumber: The External # for the document being consumed. Either number or externalNumber must be specified in the input message. If both are specified, they must refer to the same document, or else there is an error and the operation will not complete.</td>
</tr>
<tr>
<td></td>
<td>• string: warehouseName: The Warehouse for the document in SmartTurn. The warehouseName is optional, but is required in two business situations:</td>
</tr>
<tr>
<td></td>
<td>• If the UserId in the inCredential for this request has ‘All Warehouses’ as the default warehouse context, then the warehouseName is required.</td>
</tr>
<tr>
<td></td>
<td>• If the UserId in the inCredential for this request as a particular warehouse as the default</td>
</tr>
</tbody>
</table>
warehouse context, but has access to a second warehouse and needs to cancel a document from the second warehouse, then the warehouseName (name of the second warehouse) is required.

Output message: `consumeDocumentsResponse`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| UploadResponse: uploadResponse | • `string`: `externalRefId` – This identifier is optionally part of the input message: it is the element `externalNumber`. It is returned as part of the output message.  
• `string`: `systemId` – This identifier is optionally part of the input message: it is the element `number`. It is returned as part of the output message.  
• `ActionStatusType`: `status` – The result of the web service request: SUCCESS or FAILURE.  
• `string`: `error` – The error message. Only present if `status` is not SUCCESS.  
• `string`: `warning` – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown. |

### 3.8.7 `updateBOLPkgAttributes`

**Overview:** This web service operation updates the fields of the packages for the specified bill of lading in SmartTurn. It is most useful for example if a third party system is generating shipping charges or tracking numbers and this data needs to be input into SmartTurn.

This operation requires a username/password: the BOL # and optionally, the warehouse of the bill of lading; and information about the packages to update. This operation requires all packages and their fields to be passed in as part of the request. If one of the packages is missing, the operation fails. If one of the fields is missing, that field is updated to a null or 0 value.

This operation fails if the selected document is in `Committed` or `Consumed` status.

This operation requires the permission “Edit Package” under `Wave Fulfillment>Bills of Lading`.

**Input message:** `updateBOLPkgAttributes`

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • `string`: `UserId` (required)  
• `choice`: `string`: `Password` (required) |
| WSBOLForPackageUpdate: bol | • `IdType`: `bolNumber`: The BOL # for the bill of lading whose packages are being updated.  
• `string`: `warehouseName`: The Warehouse for the bill of lading whose packages are being updated. The `warehouseName` is optional, but is required in two business situations:  
  • If the `UserId` in the `inCredential` for this request has ‘All Warehouses’ as the default warehouse context, then the `warehouseName` is required.
  • If the `UserId` in the `inCredential` for this request as a
particular warehouse as the default warehouse context, but has access to a second warehouse and needs to cancel a document from the second warehouse, then the warehouseName (name of the second warehouse) is required.

- WSBOLPackageForUpdate: bolPackage: Information about the package being updated. This includes:
  - integer: packageNumber (Package #, required)
  - string: trackingNumber (Tracking #)
  - string: trackingUrl (Tracking URL)
  - Currency: totalEstimatedCharges (Total Est Charges)
  - Quantity: cube (Cube)
  - Quantity: dimWeight (Dim Wt)
  - integer: pallets (Pallets)
  - HazardousFlagEnum: hazardous (Hazardous)
  - string: comments (Comments)

Output message: updateBOLPkgAttributesResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>UploadResponse: uploadResponse</td>
<td>• string: externalRefId – This identifier is optionally part of the input message: it is the element externalNumber. It is returned as part of the output message.</td>
</tr>
<tr>
<td></td>
<td>• string: systemId – This identifier is optionally part of the input message: it is the element number. It is returned as part of the output message.</td>
</tr>
<tr>
<td></td>
<td>• ActionStatusType: status – The result of the web service request: SUCCESS or FAILURE.</td>
</tr>
<tr>
<td></td>
<td>• string: error – The error message. Only present if status is not SUCCESS.</td>
</tr>
<tr>
<td></td>
<td>• string: warning – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown.</td>
</tr>
</tbody>
</table>

### 3.8.8 setBOLCustomFields

**Overview:** This web service operation updates the custom fields for custom field Type of Inventory and Outbound for the specified bill of lading in SmartTurn.

This operation requires a username/password; the BOL # and optionally, the warehouse of the bill of lading; and the custom fields and values.

If any of the Outbound (header level) custom fields are provided, then all of the Outbound custom fields must be provided, and they are updated to the provided values. Note that it is possible to include all of the Outbound (header level) custom fields and no line item information: then only the Outbound custom fields are updated, and no change is made to the line items. For this option, the operation fails if the selected document is in Committed or Consumed status.
If any of the Inventory (line item level) custom fields are provided, then all of the line items and all of their Inventory custom fields must be provided, and the line items must be arranged in packages as they are on the bill of lading. Additionally, all of the Outbound (header level) custom fields must be provided. The Inventory Custom fields are updated to the provided values. If one of the packages or one of the line items is missing, or if one of the custom fields for the line item is missing, the operation fails. For this option, the operation fails if the selected document is in Closed, Committed, or Consumed status. (That is, header fields can be updated while the document is Closed, as long as it is not yet Committed. But line item fields cannot be updated once the document is Closed.)

This operation requires the permission “Edit Bill of Lading” under Wave Fulfillment>Bills of Lading.

**Input message:** updateBOLPkgAttributes

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| WSBOLCustomFields: bol |  
• IdType: bolNumber: The BOL # for the bill of lading whose packages are being updated.  
• string: warehouseName: The Warehouse for the bill of lading whose packages are being updated. The warehouseName is optional, but is required in two business situations:  
  • If the UserId in the inCredential for this request has ‘All Warehouses’ as the default warehouse context, then the warehouseName is required.  
  • If the UserId in the inCredential for this request as a particular warehouse as the default warehouse context, but has access to a second warehouse and needs to cancel a document from the second warehouse, then the warehouseName (name of the second warehouse) is required.  
• customFields: customFields: The header level (business process Outbound) custom fields.  
• WSBOLPackageCustomFields: bolPackage: Information about the package whose line items (custom fields of business process Inventory) are being updated. This includes:  
  • integer: packageNumber (Package #)  
  • WSPackItemCustomFields: packItem  
    • integer: lineNumber (Line # of the BOL pack item, in Full View)  
    • customFields: customFields (the Inventory custom fields) |

**Output message:** updateBOLPkgAttributesResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.9 Reporting Operations

3.9.1 lookUpInventory

Overview: This web service operation retrieves inventory items from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting inventory items.

This operation requires the permission “View Inventory Report Details”, under Inventory>Inventory Reports.

Input message: lookUpInventory

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criterion or LookUpManager is passed in inCriteria, the first 1000 inventory items available in the system are returned. For this operation, the Criteria can include any InventoryItem elements except for license, location, quantityOnHand, and valueEach. (If you need to filter by these attributes, create an inventory manager within the SmartTurn UI and then pass that as the LookUpManager.)  
• See Section 1.5, Web Service Criteria, for information on criteria. |

Output message: lookUpInventoryResponse

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
InventoryLookupResult: result

- resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.7 Reporting lookup types, for more information on lookup results.
- error – The error message(s) that resulted when processing the lookup criteria.
- inventoryItems – The collection of InventoryItem elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.

3.9.2 lookUpBillingItems

Overview: This web service operation retrieves billing lines from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting billing lines from across all billable documents (purchase orders, receipts, sales orders, shipments, storage documents).

This operation requires the permission “Billing Line Manager and Criteria Execute”, under Billing>Billing Line Manager.

Input message: lookUpBillingItems

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria or LookUpManager is passed in inCriteria, the first 500 billing lines available in the system are returned. For this operation, the Criteria can include any WSBillingLineItem element.  
• See Section 1.5, Web Service Criteria, for information on criteria. |

Output message: lookUpBillingItemsResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| WSBillingItemsLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.7 Reporting lookup types, for more information on lookup results.  
• error – The error message(s) that resulted when processing the lookup criteria.  
• billingItems – The collection of WSBillingLineItem elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |
3.9.3 lookUpItemMasterAtp

Overview: This web service operation retrieves available to promise information for item masters from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting item masters from across the system.

This operation requires the permission “Available to Promise”, under Inventory>Inventory Reports.

Input message: lookUpItemMasterAtp

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria is passed, then available to promise information for the first 1000 item masters available in the system are returned. For this operation, the Criteria can include any WSItemMasterAtp elements except untaggedQuantityOnHand, taggedQuantityOnHand, quantityOnHand, pendingKitCreateQuantity, pendingKitConsumptionQuantity, savedPurchases, savedSales, pendingPurchases, pendingSales, atp, overStock, reorderPoint, value, and preferredVendor.  
• See Section 1.5, Web Service Criteria, for information on criteria. |

Output message: lookUpItemMasterAtpResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| WSItemMasterAtpLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.7 Reporting lookup types, for more information on lookup results.  
• error – The error message(s) that resulted when processing the lookup criteria.  
• items – The collection of WSItemMasterAtp elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |
### 3.9.4 lookUpVoidedItems

**Overview:** This web service operation retrieves “anomaly” items – that is, line items that record deviations from normal business processes, such as voided shipment items or rejected picks – from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting anomaly items from across the system.

**Input message:** lookUpVoidedItems

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential: inCredential</td>
<td></td>
</tr>
<tr>
<td>• string: UserId (required)</td>
<td></td>
</tr>
<tr>
<td>• choice: string: Password (required)</td>
<td></td>
</tr>
<tr>
<td>Criteria: inCriteria</td>
<td></td>
</tr>
<tr>
<td>• If no Criteria is passed, then information for the first 1000 voided shipment items available in the system are returned. For this operation, the Criteria can include any WSVoidedItem elements except originalQuantity, customFields, voidedQuantity, and isInventoryAffected.</td>
<td></td>
</tr>
<tr>
<td>• See Section 1.5, Web Service Criteria, for information on criteria.</td>
<td></td>
</tr>
</tbody>
</table>

**Output message:** lookUpVoidedItemsResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSVoidedItemsLookupResult: result</td>
<td></td>
</tr>
<tr>
<td>• resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.7 Reporting lookup types, for more information on lookup results.</td>
<td></td>
</tr>
<tr>
<td>• error – The error message(s) that resulted when processing the lookup criteria.</td>
<td></td>
</tr>
<tr>
<td>• items – The collection of WSVoidedItem elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0.</td>
<td></td>
</tr>
</tbody>
</table>

### 3.9.5 lookUpActivityLines

**Overview:** This web service operation retrieves activity lines from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting activity items from across the system.

This operation requires the permission “Activity Line Manager and Criteria Execute”, under Administration>Documents.

**Input message:** lookUpActivityLines

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OccamPasswordCredential: inCredential</td>
<td></td>
</tr>
<tr>
<td>• string: UserId (required)</td>
<td></td>
</tr>
<tr>
<td>• choice: string: Password (required)</td>
<td></td>
</tr>
</tbody>
</table>
Criteria: inCriteria

- If no Criteria or LookUpManager is passed, then information for the first 1000 activity lines available in the system are returned. For this operation, the Criteria can include any WSActivityLine elements except quantity and customFields.
- See Section 1.5, Web Service Criteria, for information on criteria.

Output message: lookUpActivityLinesResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| WSActivityLinesLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.7 Reporting lookup types, for more information on lookup results.  
  • error – The error message(s) that resulted when processing the lookup criteria.  
  • items – The collection of WSActivityLine elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |

3.10 Administrative Operations

3.10.1 saveItemMaster

Overview: This web service operation saves (creates or updates) item masters in SmartTurn. The operation requires a username/password and a collection of item masters. A single operation can save (create or update) 1000 item masters. If the item master is not already configured in SmartTurn, this operation will create a new item master.

This operation requires the permission “Add/Edit”, under Administration>Item Masters.

Input message: saveItemMaster

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
  • choice: string: Password (required) |
| ItemMaster: inItemMasters | • IdType: id (Item #)  
  • DescriptionType: description (Description)  
  • ItemTypeEnum: type (Item Type)  
  • Group: group (Group)  
  • boolean: lotControlled (General>Lot Controlled)  
  • boolean: expirationDateControlled (General>Expiration Date Controlled)  
  • boolean: receiptDateControlled (not used in application) |
<table>
<thead>
<tr>
<th>NameType</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>manufacturer</td>
<td>(General&gt;Manufacturer)</td>
<td></td>
</tr>
<tr>
<td>ManufacturerId</td>
<td>(General&gt;Manufacturer ID)</td>
<td></td>
</tr>
<tr>
<td>ItemId</td>
<td>(General&gt;Manufacturer Item ID)</td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td>serialControlled (deprecated)</td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td>serialUnique (deprecated)</td>
<td></td>
</tr>
<tr>
<td>SerialControlType</td>
<td>serialControlType (one of NONE, UNIQUE, SIMPLE_LABEL, OPEN CONTAINER, CLOSED CONTAINER; General&gt;Serial Control Mode)</td>
<td></td>
</tr>
<tr>
<td>UpcCode</td>
<td>upc</td>
<td>(General&gt;UPC)</td>
</tr>
<tr>
<td>EanCode</td>
<td>ean</td>
<td>(General&gt;EAN)</td>
</tr>
<tr>
<td>GeneralLedgerCode</td>
<td>generalLedgerCode</td>
<td>(General&gt;General Ledger Code)</td>
</tr>
<tr>
<td>HazardousFlag</td>
<td>hazardous</td>
<td>(one of YES, NO; Material Handling&gt;Hazardous)</td>
</tr>
<tr>
<td>ItemClassification</td>
<td>classification (one of A, B, C; Material Handling&gt;Classification)</td>
<td></td>
</tr>
<tr>
<td>Currency</td>
<td>purchaseCost</td>
<td>(PO/Receiving&gt;Purchase Cost per Receiving Unit)</td>
</tr>
<tr>
<td>Currency</td>
<td>salesCost</td>
<td>(Sales/Fulfillment&gt;Sales Price per Shipping Unit)</td>
</tr>
<tr>
<td>boolean</td>
<td>sellable</td>
<td>(Sales/Fulfillment&gt;Sellable)</td>
</tr>
<tr>
<td>Unit</td>
<td>standardUnit</td>
<td>(Units of Measure&gt;Standard Unit)</td>
</tr>
<tr>
<td>Unit</td>
<td>measurementUnit</td>
<td>(Units of Measure&gt;Measurement Unit)</td>
</tr>
<tr>
<td>Unit</td>
<td>shippingUnit</td>
<td>(Units of Measure&gt;Shipping Unit)</td>
</tr>
<tr>
<td>Unit</td>
<td>receivingUnit</td>
<td>(Units of Measure&gt;Receiving Unit)</td>
</tr>
<tr>
<td>Unit</td>
<td>weightUnit</td>
<td>(Units of Measure&gt;Weight Unit)</td>
</tr>
<tr>
<td>Unit</td>
<td>billingUnit</td>
<td>(Units of Measure&gt;Billing Unit)</td>
</tr>
<tr>
<td>boolean</td>
<td>outBoundIdControlled</td>
<td>(General&gt;Outbound ID Controlled)</td>
</tr>
<tr>
<td>boolean</td>
<td>allowConsolidationOfPallets</td>
<td>(Material Handling&gt;Allow Consolidation of Pallets)</td>
</tr>
<tr>
<td>boolean</td>
<td>basedOnMeasurementUnit</td>
<td>(Units of Measure&gt;Based on Measurement Unit)</td>
</tr>
<tr>
<td>OutboundVerificationStatus</td>
<td>outBoundIdVerificationStatus (one of NO_CHECK [General&gt;Outbound ID Length Not Checked], CHECK_FROM_ENTRY [General&gt;Outbound ID Length Checked From Entry], FIXED_LENGTH [General&gt;Outbound ID Length Fixed])</td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td>outBoundIdControlled</td>
<td>(General&gt;Outbound ID Length)</td>
</tr>
<tr>
<td>ItemmasterWarehouse</td>
<td>itemMasterWarehouses</td>
<td>(an entry in the table on the Warehouses tab)</td>
</tr>
<tr>
<td>Affiliate</td>
<td>owner</td>
<td>(General&gt;Owner)</td>
</tr>
</tbody>
</table>
• WSMultiAttributeFulfillmentEnum:
pickOnItemAttributes (one of NONE, CORE, CORE_AND_FLEX; Sales/Fulfillment>Pick on item attributes)
• CustomFields: itemMasterCustomFields (fields in the Custom tab; see Section 2.4, Flex Field Types)
• string: preferredVendor (PO/Receiving>Preferred Vendor)

Output message: saveItemMasterResponse

<table>
<thead>
<tr>
<th>Return value</th>
<th>Elements</th>
</tr>
</thead>
</table>
| uploadResponse: result | • string: externalRefId – The alphanumeric identifier used by the client (web service invoker) to identify the item master. Because item masters cannot be generated externally, the externalRefId is always identical to the inItemMasters : id element, which is the unique identifier for the item master.  
• status – The result of the web service request: SUCCESS or FAILURE.  
• error – The error message. Only present if status is not SUCCESS.  
• warning – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown. |

Notes:
The inItemMasters : id element is a required unique identifier. It represents an item master as a natural identifier and is visible to end users throughout SmartTurn as Item #.

The inItemMasters : description element is a required brief prose description for the item, limited to 255 characters.

The inItemMasters : group element is a required unique name that represents an existing item master group in SmartTurn. (If your business does not use item master groups, enter "All", which is the SmartTurn default group.)

The inItemMasters : allowConsolidationOfPallets element set to true indicates that pallets are always consolidated as much as possible after every operation; set to false indicates that inventory items received on separate pallets (separate receipt lines) will remain as separate pallets and will not merge. If left blank, this element defaults to true.

The inItemMasters : basedOnMeasurementUnit element set to true indicates that all of the measurements for all of the units of measure for this item are proportional to those in the measurementUnit, based on the factor; set to false indicates that there are no restrictions or validations on any of the measurements for any of the units of measure. For example, if your each has a weight of 10 lbs and your case has a factor of 8, then your case must weigh 80 lbs if this element is true. If this element is false, your case can weigh any amount; for instance it might weigh 85 lbs if it includes extra packaging. If left blank, this element defaults to true.

The inItemMasters : pickOnItemAttributes element determines whether, for this item master, users can specify in the sales order line item some properties of the inventory item that should be used to fulfill this sales order line. A value of NONE means that only Item #, Ordered Qty, and Owner will determine fulfillment. A value of CORE means that additionally, Lot #, Serial #, License #, Expire
Date, and Product ID will determine fulfillment. A value of CORE_AND_FLEX means that additionally, custom fields of Business Process Type Inventory will determine fulfillment.

The inItemMasters : standardUnit, shippingUnit, receivingUnit, weightUnit, billingUnit, and measurementUnit: elements are all of type Unit. (Only the standardUnit is required.) The Unit type includes unitDetails and unit of measure abbreviation. If you are using an existing unit of measure, you only need to specify the abbreviation. For example, assuming an existing unit of measure “pound” with abbreviation “lb”:

```xml
<s0:inItemMasters>
   <s0:shippingUnit>
      <tns:abbreviation>lb</tns:abbreviation>
   </s0:shippingUnit>
</s0:inItemMasters>
```

If you are creating a new unit of measure, you also need to include the inItemMasters : Unit : unitDetails element in the saveItemMasters request. The unitDetails element requires a name and a factor, as well as optional measurements for this unit of measure: height, width, length, cube, weight, ti, hi, numInPallet. For the optional measurements, enter the value and the unitAbbreviation. The factor indicates how many eaches exist in this unit of measure. The measurements indicate the physical characteristics of the goods; if left blank, they default to an interpolation from the measurements specified in the measurementUnit and adjusted relative to the factor. In the following example, there are 20 eaches in a bolt, and the measurements are left as interpolations from the measurementUnit:

```xml
<s0:inItemMasters>
   <s0:shippingUnit>
      <tns:abbreviation>bolt</tns:abbreviation>
      <tns:unitDetails>
         <tns:name>bolt</tns:name>
         <tns:factor>20</tns:factor>
      </tns:unitDetails>
   </s0:shippingUnit>
</s0:inItemMasters>
```

The inItemMasters : itemMasterCustomFields element allows you to update custom field values for the item master. You need to specify the type of custom field, appropriate value, and name. For example:

```xml
<s0:inItemMasters>
   <s0:CustomField>
      <tns:name>Seasonal</tns:name>
      <tns:listDisplayValue>Summer</tns:listDisplayValue>
      <tns:key>sample_key</tns:key>
   </s0:CustomField>
   <s0:CustomField>
      <tns:name>Requires special packaging {describe}</tns:name>
      <tns:textValue>needs ‘right side up’ and ‘fragile’ stickers</tns:textValue>
   </s0:CustomField>
</s0:inItemMasters>
```
More information about the CustomFields type is in Section 2.4, Flex Field Types. Note that the custom fields must already have been set up within SmartTurn; this operation just provides the value of the custom field for this item master.

3.10.2 lookUpItemMaster

Overview: This web service operation retrieves item masters from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting item masters.

This operation requires the permission “View Detail”, under Administration>Item Masters.

Input message: lookUpItemMaster

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria is passed, the first 1000 item masters available in the system are returned. For this operation, the Criteria can include any ItemMaster elements except for itemMasterCustomFields, owner, group, serialControlled (deprecated), serialUnique (deprecated), standardUnit, measurementUnit, shippingUnit, receivingUnit, weightUnit, billingUnit, itemMasterWarehouses, allowConsolidationOfPallets, basedOnMeasurementUnit, expirationDateControlled, lotControlled, outBoundIdControlled, receiptDateControlled, and sellable.  
• See Section 1.5, Web Service Criteria, for information on criteria. |

Output message: lookUpItemMasterResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| ItemMasterLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.8 Administrative lookup types, for more information on lookup results.  
• error – The error message(s) that resulted when processing the lookup criteria.  
• itemmasters – The collection of ItemMaster elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |
### 3.10.3 lookUpBusinessAffiliates

**Overview:** This web service operation retrieves business affiliates from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting business affiliates from across the system.

This operation requires the permission “View Detail”, under Administration>Business Affiliates.

**Input message:** lookUpBusinessAffiliates

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria is passed, the first 100 business affiliates available in the system are returned. For this operation, the Criteria can include any of the following WSBusinessAffiliate elements: companyName, companyCode, ownerStatus, vendorStatus, carrierStatus, brokerStatus, customerStatus, ownerName, ownerCode.  
• See Section 1.5, Web Service Criteria, for information on criteria. |

**Output message:** lookUpBusinessAffiliatesResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| WSBusinessAffiliateLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.8 Administrative lookup types, for more information on lookup results.  
• error – The error message(s) that resulted when processing the lookup criteria.  
• businessAffiliates – The collection of WSBusinessAffiliate elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |

### 3.10.4 saveBusinessAffiliates

**Overview:** This web service operation saves (creates or updates) business affiliates in SmartTurn. The operation requires a username/password and a collection of business affiliates. A single operation can save (create or update) 100 business affiliates. If the business affiliate is not already configured in SmartTurn, this operation will create a new business affiliate.

This operation requires the permission “Add/Edit”, under Administration>Business Affiliates.

**Input message:** saveBusinessAffiliates

<table>
<thead>
<tr>
<th>Elements (type: name)</th>
<th>Components</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
On-Demand WMS (SmartTurn) Web Services - Web Services API

**WSBusinessAffiliate: inBusinessAffiliates**

- `string`: `companyName` (Name)
- `IdType`: `companyCode` (Code)
- `WSBAStatus`: `customerStatus` (Types>Customer Status)
- `WSBAStatus`: `vendorStatus` (Types>Vendor Status)
- `WSBAStatus`: `ownerStatus` (Types>Owner Status)
- `WSBAStatus`: `carrierStatus` (Types>Carrier Status)
- `WSBAStatus`: `brokerStatus` (Types>Broker Status)
- `string`: `ownerName` (the Name of the Business Affiliate Owner, if any, for this business affiliate)
- `string`: `ownerCode` (the Code of the Business Affiliate Owner, if any, for this business affiliate)
- `Address`: `mainAddress` (Main>Address line 1, Address line 2, City, State, Country, Postal code)
- `Address`: `shipToAddress` (Ship To>Address line 1, Address line 2, City, State, Country, Postal code)
- `Address`: `billToAddress` (Bill To>Address line 1, Address line 2, City, State, Country, Postal code)
- `Address`: `shipFromAddress` (Ship From>Address line 1, Address line 2, City, State, Country, Postal code)
- `WSContactList`: `contacts` (Contacts table)
- `WSBillingPolicyList`: `billingPolicies` (table in Billing Policies tab)

**Output message**: `saveBusinessAffiliateResponse`

**Return value** | **Elements**
---|---
`uploadResponse`: `result` |  
- `string`: `externalRefId`
- `status` – The result of the web service request: `SUCCESS` or `FAILURE`.
- `error` – The error message. Only present if `status` is not `SUCCESS`.
- `warning` – Warning messages, if any. The web service request will be successfully processed even if a warning is thrown.

**Notes:**

The `inBusinessAffiliates`: `companyName` and `companyCode` elements are visible to end users throughout SmartTurn. The `companyCode` must be unique across all business affiliates for this tenant.

The `inBusinessAffiliates`: `customerStatus`, `vendorStatus`, `ownerStatus`, `carrierStatus`, and `brokerStatus` can each be set to `ACTIVE`, `INACTIVE`, or `NEVER_ACTIVATED`. If you are creating a business affiliate, set these to `ACTIVE` or `NEVER_ACTIVATED`. If you are updating a business affiliate, set these to `ACTIVE` or `INACTIVE`. An `xxxStatus` of `ACTIVE` means the business affiliate is available for those roles in SmartTurn. For example, `brokerStatus` of `ACTIVE` means the business affiliate can be entered in the Broker field on SmartTurn documents or other places in the system where a valid business affiliate of type Broker is required.
The `inBusinessAffiliates` elements must all be set individually. (The SmartTurn user interface provides a shortcut in which by default all addresses are set to the `mainAddress`. This behavior is not supported through the web service operation.)

The `inBusinessAffiliates` elements are optional and indicate the Business Affiliate Owner of this business affiliate. If this business affiliate should have a Business Affiliate Owner, the `ownerName` and `ownerCode` must match exactly the `companyName` and `companyCode` of an existing active business affiliate of type Owner in SmartTurn (that is, an existing business affiliate with `ownerStatus` of ACTIVE). Note that these are automatically set to `companyName` and `companyCode`, respectively, if this business affiliate is of type Owner.

The `inBusinessAffiliates` is a list of contacts for this business affiliate. Each `WSContact` requires a `lastName` and `email`. (Note that for backwards compatibility, the `email` and `lastName` elements are not required in the schema; however, they are required by the current business logic in SmartTurn). Each `WSContact` optionally has other contact information as well. Additionally, each `WSContact` requires information about what kind of contact it is: `isPaymentContact`, `isShipmentContact`, `isPurchaseOrderContact`, `isSalesOrderContact`, `isMainContact`. For each business affiliate, there can only be one contact that is a named kind of contact. For example, for each business affiliate, there can only be one contact with `isPaymentContact` set to true. There can be an unlimited number of contacts for which all of the `isXXXContact` predicates are false.

```xml
<s0:inBusinessAffiliates>
  <s0:contacts>
    <s0:contact>
      <s0:isPaymentContact>false</s0:isPaymentContact>
      <s0:isShipmentContact>false</s0:isShipmentContact>
      <s0:isPurchaseOrderContact>false</s0:isPurchaseOrderContact>
      <s0:isSalesOrderContact>false</s0:isSalesOrderContact>
      <s0:isMainContact>true</s0:isMainContact>
      <s0:firstName>John</s0:firstName>
      <s0:lastName>Smith</s0:lastName>
      <s0:email>jsmith@smartturn.com</s0:email>
    </s0:contact>
    <s0:contact>
      <s0:isPaymentContact>true</s0:isPaymentContact>
      <s0:isShipmentContact>true</s0:isShipmentContact>
      <s0:isPurchaseOrderContact>false</s0:isPurchaseOrderContact>
      <s0:isSalesOrderContact>false</s0:isSalesOrderContact>
      <s0:isMainContact>false</s0:isMainContact>
      <s0:lastName>Rodriguez</s0:lastName>
      <s0:email>mrodriguez@smartturn.com</s0:email>
    </s0:contact>
  </s0:contacts>
</s0:inBusinessAffiliates>
```
If you are updating an existing business affiliate in the system, SmartTurn matches contacts based on the email and lastName in the WSContact.

- If the email and lastName match, the other contact information is overwritten with the contact information in the saveBusinessAffiliates request. This includes the isXXXContact predicates.
- Any contacts that you do NOT include in a update request remain unchanged. For example, if your existing business affiliate has three contacts and you want to update one of them, you only need to send information for that one contact. The information in that one contact is then overwritten with the contact information in the saveBusinessAffiliates request, and the other contacts remain unchanged.

The inBusinessAffiliates : billingPolicy is a list of billing policies for this business affiliate. (Billing policies are only relevant elsewhere in SmartTurn if the business affiliate has an ownerStatus of ACTIVE, but they can be stored for any business affiliate.) Each WSBillingPolicy requires a chargeTypeName, billingBasis, rate, and documentType. The chargeTypeName must match the name of an existing charge type in SmartTurn. The generalLedgerCode is optional; if not provided, the generalLedgerCode will be set to the general ledger code of the charge type. The billingBasis can be perUoM, perLine, or perDocument. If it is perUoM, then the uomName is required. The documentType can be PO (purchase order), RECEIPT, SO (sales order), SHIPMENT, or STORAGE. The combination of chargeTypeName, billingBasis, uomName, and documentType must be unique for this business affiliate. If the billingBasis is perUoM or perLine and the documentType is PRODUCTIONORDER (kit order) or PRODUCTIONSERVICE (kit), the WSBillingPolicy also requires the lineType to be set. The lineType indicates which lines of the kit order or kit (products, by-products, or components) are being billed for when the policy is applied.

If you are updating an existing billingPolicy in the system, SmartTurn matches billingPolicies based on the combination of chargeTypeName, billingBasis, uomName, and documentType.

- If the billing policy is a match based on those four elements, the other billingPolicy information is overwritten with the new billing policy information in the saveBusinessAffiliates request.
- Any billing policies that you do NOT include in an update request remain unchanged. For example, if your existing business affiliate has three billing policies and you want to update the rate on one of them, you only need to send information for that one policy. The information in that one policy is then overwritten with the new information in the saveBusinessAffiliates request, and the other billing policies remain unchanged.

```xml
<s0:billingPolicies>
  <s0:billingPolicy>
    <s0:chargeTypeName>receiving - pallets</s0:chargeTypeName>
    <s0:billingBasis>perUoM</s0:billingBasis>
    <s0:uomName>pallets</s0:uomName>
    <s0:rate>2.75</s0:rate>
    <s0:documentType>RECEIPT</s0:documentType>
  </s0:billingPolicy>
</s0:billingPolicies>
```
3.10.5 lookUpLocation

**Overview:** This web service operation retrieves locations from SmartTurn based on the criteria used. The operation requires a username/password and optionally a criteria for selecting locations from across the system.

This operation requires the permission “Add/Edit”, under Administration>Locations.

**Input message:** lookUpLocation

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Notes</th>
</tr>
</thead>
</table>
| OccamPasswordCredential: inCredential | • string: UserId (required)  
• choice: string: Password (required) |
| Criteria: inCriteria | • If no Criteria is passed, the first 1000 locations available in the system are returned. For this operation, the Criteria can include any of the WSLocation elements, with the exception of the capacityStatus, isAdjustable, isPutAwayLocation, isPickableLocation, isOwned, and isBonded fields.  
• See Section 1.5, Web Service Criteria, for information on criteria. |

**Output message:** lookUpLocationResponse

<table>
<thead>
<tr>
<th>Elements (name: type)</th>
<th>Components</th>
</tr>
</thead>
</table>
| WSLocationLookupResult: result | • resultMetadata – The metadata about the result collection, including totalRecordsCount, currentPageIndex, recordCountInThisPage, and totalPages. See Section 2.3.8 Administrative lookup types, for more information on lookup results.  
• error – The error message(s) that resulted when processing the lookup criteria.  
• locations – The collection of WSLocation elements that match the input criteria. If the lookup returns no results, this component is not returned, and the resultMetadata indicates a totalRecordsCount of 0. |