

Connection Guide

ShipStation for

NetSuite

Version 0.9.1
Summer '23



Table of Contents

[Preinstallation](#)

[Introduction](#)

[Workflow Options](#)

[Field Mappings](#)

[NetSuite Instance Preparation](#)

[Connecting ShipStation to NetSuite](#)

[FAQ](#)

[Basic Troubleshooting](#)

Preinstallation

You will need the following to successfully complete this connection:

- NetSuite Account
- NetSuite Consumer Key and Secret
- NetSuite Access Token and Secret
- NetSuite Account ID
- ShipStation Account

Introduction

The ShipStation for NetSuite connector is intended to be a self-service, out-of-the-box integration that enables ShipStation merchants to quickly and efficiently connect their existing NetSuite instance to ShipStation for Sales Order -or- Item Fulfillment Record shipment processing.

This document serves as a connection guide for the ShipStation for NetSuite connector. The first sections cover [workflow options](#) and [standardized field mappings](#) that exist for the 3 unique workflows available with this integration. The following sections will walk you through NetSuite token creation and general [NetSuite instance preparation](#) needed *before* [connecting ShipStation to NetSuite](#). Additionally, there are [frequently asked questions](#) and [basic troubleshooting](#) sections at the end of this document.

Please read through this entire document.

Workflow Options

About the Unique Workflows

There are three unique workflow options that are available for merchants connecting NetSuite to ShipStation.

Sales Order

Pick Pack Ship - Picked

Pick Pack Ship - Packed

The **Sales Order** workflow will enable ShipStation to import NetSuite Sales Orders in full and allow the ShipStation merchant to fulfill (*or partially fulfill*) these Sales Orders. Upon label creation in ShipStation, an Item Fulfillment Record will be *created* in NetSuite. This Item Fulfillment Record will be linked to the NetSuite Sales Order and created in the **Shipped** status. Once all line items have been fulfilled, the NetSuite Sales Order will move from the **Pending Fulfillment** status to the **Pending Billing, Billed, or Completed** status. *The Sales Order workflow is optimal if splitting Sales Orders into multiple shipments in ShipStation is preferred to creating Item Fulfillment Records in NetSuite.*

The **Pick Pack Ship - Picked** workflow will enable ShipStation to import NetSuite Item Fulfillment Records created from an underlying Sales Order in the **Picked** status and allow the ShipStation merchant to fulfill the Item Fulfillment Record. Upon label creation in ShipStation, the existing NetSuite Item Fulfillment Record will be updated from the **Picked** status to the **Shipped** status. *The Pick Pack Ship workflows are optimal if a WMS is being utilized for the Pick/Pack process, scan actions create Item Fulfillment Records inside of NetSuite, etc.*

The **Pick Pack Ship - Packed** workflow will enable ShipStation to import NetSuite Item Fulfillment Records created from an underlying Sales Order in the **Packed** status and allow the ShipStation merchant to fulfill the Item Fulfillment Record. Upon label creation in ShipStation, the existing NetSuite Item Fulfillment Record will be updated from the **Packed** status to the **Shipped** status. *The Pick Pack Ship workflows are optimal if a WMS is being utilized for the Pick/Pack process, scan actions create Item Fulfillment Records inside of NetSuite, etc.*

Field Mappings

About the Standardized Field Mappings

A Note About Field Mappings

The ShipStation for NetSuite connector utilizes NetSuite **native fields** to import order data into ShipStation and to post shipment information from ShipStation into NetSuite.

Please Note: Custom fields are **NOT** supported with this connector.

Sales Order Workflow Import Field Mappings

NetSuite -> ShipStation

NetSuite Transaction API	API Field Name	ShipStation SalesOrder Export API	API Field Name	Notes
Transaction(type='SalesOrd')	id	Sales_Orders	order_id	
	trandid		order_number	
			status	Pending Fulfillment, Partially Fulfilled, Partially Fulfilled/Pending Billing -> 'AwaitingShipment' Pending Billing, Billed, Closed -> Completed Cancelled -> Cancelled NOTE: Pending Approval is NOT pulled.
	lastmodifieddate		Modified_date_time	NetSuite datetime converted to UTC
	createddate		Created_Date_time	NetSuite datetime converted to UTC
	shipdate		ship_by_date	Under Shipping_preferences element. NetSuite datetime converted to UTC
	totaltax			
Transactionline	order_id-id-linensequence number		line_item_id	
	Rateamount			
	quantity		quantity	
customer	customerid	Buyer	Buyer_id	Under Sales_Orders->buyer element
	customername		Buyername	
	phone		buyerphone	
transaction	email		customer_email	
transactionbillingaddress	addressee	bill_to	name	Under requested_fulfillments-> Bill_to element

	addr1		address_line_1	
	city		city	
	state		state_province	
	Country		Country	
	zip		postal_code	
transactionshipping address	addressee	Ship_to	name	Under requested_fulfillments-> ship_to element
	addr1		address_line_1	
	city		city	
	state		state_province	
	Country		country	
	zip		postal_code	
item	id	Product	product_id	
	displayaname		name	
	itemid	identifiers	sku	Under Product->identifiers element
	unitprice		unitprice	
	Weight	weight	unit	Under Product->Weight element
	Weightunit		value	
Transactionline(ite mtype='Shipltem')	rateamount	shipping_charges	Amount	Under payment-> shipping_Charges element
Location(name)	location	shipping_preferences	requested_warehouse	under requested_fulfillment->Shippi ng_Preferences
	shipcarrier		shipping_service	under requested_fulfillment->Shippi ng_Preferences
transaction	Otherrefnum	Extensions	Custom_field_1	Refers to PO# from NetSuite
	memo		Custom_field_2	Refers to Memo field from NetSuite
	source		Custom_field_3	Refers to Source field from NetSuite

Sales Order Workflow Export Field Mappings

ShipStation -> NetSuite

ShipStation Shipment Notification API	API Field Name	NetSuite Tranform Order API	API Field Name	Notes
ShipmentNotification	Notification_id	Item Fulfillment		This id is used in the response.
	Order_id			Used in the response
	ship_date		shippedDate	
	line_item_id		orderLine	Get the line number from the line_item_id field (order_id-product_id-line_item_id)
	Ext_location_id (Ship From Location Name)		location	
	quantity		quantity	
			Itemreceive	false
			Shipoverride	true
	ship_to.address_line_1		Addr1	
	ship_to.name		addressee	
			addrText	Complete address entered as strin
	ship_to.city		city	
	ship_to.state_province		state	
	ship_to.country_code		id	Two letter country code. For ex: US
	ship_to.postal_code		zip	
	tracking_number	Package	packagetrackingnumber	
			packageWeight	Total package weight is calculated using the weight of each item in the order times the quantity of that item. Weight = (item 1 * quantity 1) + (item 2 * quantity 2)....
	fulfillment_cost		shippingcost	The actual shipping paid during label creation in ShipStation

PickPackShip Workflow Import Field Mappings

NetSuite -> ShipStation

Note: The field mappings for the **Picked** and **Packed** versions of this workflow are the same. The only difference is which status corresponds to "Awaiting Shipment" in ShipStation.

NetSuite Transaction API	API Field Name	ShipStation SalesOrder Export API	API Field Name	Notes
Transaction(type='ItemShipment')	id	Sales_Orders	order_id	
	tranid		order_number	
	status		status	Picked, Packed or Shipped
	lastmodifieddate		Modified_date_time	NetSuite datetime converted to UTC
	createddate		Created_Date_time	NetSuite datetime converted to UTC
	shipdate		ship_by_date	Under Shipping_preferences element. NetSuite datetime converted to UTC
	totaltax			
Transactionline	itemfulfillmentid-id-lines equencenumber		line_item_id	
	Rateamount			
	quantity		quantity	
customer	customerid	Buyer	Buyer_id	Under Sales_Orders->buyer element
	customername		Buyername	
transaction	email		customer_email	
transactionbillingaddress	addressee	bill_to	name	Under requested_fulfillments->Bill_to element
	addr1		address_line_1	
	city		city	
	state		state_province	
	Country		Country	
	zip		postal_code	
transactionshippingaddress	addressee	Ship_to	name	Under requested_fulfillments->ship_to element
	addr1		address_line_1	
	city		city	
	state		state_province	
	Country		country	

	zip		postal_code	
item	id	Product	product_id	
	displayname		name	
	itemid	identifiers	sku	Under Product->identifiers element
	unitprice		unitprice	
	Weight	weight	unit	Under Product->Weight element
	Weightunit		value	
Transactionline(itemtype='Shipltem')	rateamount	shipping_charges	Amount	Under payment->shipping_Charges element
Location(name)	location	shipping_preferences	requested_warehouse	under requested_fulfillment->Shipping_Preferences
	shipcarrier		shipping_service	under requested_fulfillment->Shipping_Preferences
transaction	Otherrefnum	Extensions	Custom_field_1	Refers to PO# from NetSuite
	memo		Custom_field_2	Refers to Memo field from NetSuite
	source		Custom_field_3	Refers to Source field from NetSuite
Transaction	Lastmodifieddate	item	modified_date_time	transaction_type=itemfulfillmentrecord

PickPackShip Workflow Export Field Mappings

ShipStation -> NetSuite

ShipStation Shipment Notification API	API Field Name	NetSuite Tranform Order API	API Field Name	Notes
ShipmentNotification	Notification_id	Item Fulfillment		This id is used in the response.
	Order_id			Used in the response
	ship_date		shippedDate	
	line_item_id		orderLine	Get the line number from the line_item_id field (item_fulfillment_id-product_id-line_item_id)
	Ext_location_id (Ship From Location Name)		location	
	quantity		quantity	
			Itemreceive	false
			Shipoverride	true
	ship_to.address_line_1		Addr1	
	ship_to.name		addressee	
			addrText	Complete address entered as strin
	ship_to.city		city	
	ship_to.state_province		state	
	ship_to.country_code		id	Two letter country code. For ex: US
	ship_to.postal_code		zip	
	tracking_number	Package	packagetrackingnumber	
			packageWeight	Total package weight is calculated using the weight of each item in the order times the quantity of that item. Weight = (item 1 * quantity 1) + (item 2 * quantity 2)....
	fulfillment_cost		shipmentcost	The actual shipping paid during label creation in ShipStation

NetSuite Instance Preparation

About Token Generation and Settings of Interest

THESE STEPS MUST BE FOLLOWED TO ENSURE A PROPER CONNECTION BETWEEN SHIPSTATION AND NETSUITE

Token-based authentication, SuiteTalk SOAP and REST Web Services must be enabled

- Navigate to **Setup** -> **Company** -> **Enable Features**
- From the **SuiteCloud** tab, check the following options
 - SOAP Web Services
 - REST Web Services
 - Token-Based Authentication
- From the **Analytics** tab, check the following option
 - SuiteAnalytics Workbook
- From the **Transactions** tab, please note the checkbox status of the following option, as it will be required to complete the setup process in ShipStation
 - Multiple Shipping Routes

User Role and Integration Role must be created for the ShipStation for NetSuite Connector

- From NetSuite's top menu bar, navigate to **Customization** > **SuiteBundler** > **Search & Install Bundles** (*Do not click on List*)
 - On the **Search & Install Bundles** page, enter "471864" or "ShipStation for NetSuite" in the **Keywords** search box and click **Search**
 - From the search results, select the **ShipStation for NetSuite** bundle
 - On the **Bundle Details** page, click **Install**. Then from the **Preview Bundle Install** page, click **Install Bundle**
 - On the **Installed Bundles** page, the **Status** will show "Pending..." Installation may take a few minutes. You can click **Refresh** to see the progress.
 - When installation is successful, the **Status** will show a green checkmark
 - This will install the pre-configured **ShipStation Admin Role** and **ShipStation TBA** Integration Role

User Role Assignment for Existing Employee -or- New Employee

(Using an existing employee is recommended, as it will not require an additional NetSuite license)

- To assign role to an **EXISTING EMPLOYEE**
 - From NetSuite's top menu bar, navigate to **Lists > Employees > Employees**
 - On the **Employees** page, on the row for the user wish to add to the Role, click **Edit**
 - On the **Employee** page, in the **Access** section under **Roles**, use the **Role** dropdown to select the **ShipStation Admin Role**. Then click **Add**
 - Click **Save**
- To create a **NEW EMPLOYEE** and assigning role
 - From NetSuite's top menu bar, navigate to **Lists > Employees > Employees > New**
 - On the **Employee** screen, fill out the following fields at minimum:
 - Under **Primary Information**, enter a **Name**, such as "ShipStation" for the first name and "TBA" for the last name
 - Under **Email | Phone | Address**, provide an **Email**
 - In the **Access** section, select **Give Access** and set the **Password** and **Confirm Password**
 - In the **Access** section under **Roles**, use the **Role** dropdown to select the **ShipStation Admin Role**. Then click **Add**
 - Click **Save**

Consumer/Client Key and Secret

- During the ShipStation for NetSuite Bundle install, the **ShipStation TBA** Integration Role was created.
- The Consumer Key and Secret associated with this Integration Role are:
 - Consumer Key:
c0c9681d8c73dd71635cb07afa73139e9d922227b25facb56e9781040e6c075c
 - Consumer Secret:
aa76230da4262495d6d18c24ac8b576f6727aa93a004b30ede2f8b3a63ac2e99
- *These credentials are required to complete the setup process in ShipStation*

Generate NetSuite Token Key and Secret

- Navigate to **Setup > Users/Roles > Access Tokens > New**
 - Select the **Application Name/Integration Record** *ShipStation TBA*
 - Select the **New** or **Existing User** that was assigned the ShipStation Admin Role
 - Select **ShipStation Admin Role**
 - Click **Save**
- The **NetSuite Access Token Key** and **Secret** are generated after saving the record. *These values are shown one time only, so please make sure these values are saved in a secure location. They are required to complete the setup process in ShipStation*

Locate NetSuite Account ID

- From NetSuite's top menu bar, navigate to **Setup > Company > Company Information**
- Alternatively, the ID can be located in the URL bar of your internet browser
 - Ex: <https://YourNetSuiteID.app.netsuite.com/>
- *Please copy and paste this ID into a safe document, it is required to complete the setup process in ShipStation*

Connecting ShipStation to NetSuite

About the ShipStation for NetSuite Connection Process

Locate NetSuite Connection Modal

- In ShipStation, click **Account Settings (Gear Icon) > Selling Channels > Store Setup > Connect a Store or Marketplace**
 - Search **NetSuite**
 - Click **NetSuite** tile

Configuring Your Connection

- To accept the the NetSuite for ShipStation user agreement, press **Continue**
- Reference and input the following information gathered from the [NetSuite Instance Preparation](#) section of this document in the corresponding fields
 - NetSuite Account ID
 - **For sandbox use, please include “-sb1”, “-sb2”, etc**
 - NetSuite Consumer Key
 - NetSuite Consumer Secret
 - NetSuite Access Token Key
 - NetSuite Access Token Secret
- From the **Import Fulfillment As...**dropdown menu, select the desired workflow option described in the [Workflow Options](#) section of this document
- Reference the status of the **Multiple Shipping Routes** checkbox from the [NetSuite Instance Preparation](#) section of this document

Optional Filter Configuration

Using the optional **Class**, **Department**, and **Location** filters, ShipStation can be configured to only import select Sales Orders or Item Fulfillment Records from NetSuite. If multiple values for a filter are required, please comma separate the values. *These values must be an EXACT match to the values as they appear in NetSuite.* **Note: Class, Department and Location values will be read from line item details**

Completing the Connection

- Once you have verified the information entered is correct, click **Connect**

Additional Required Configuration

- In ShipStation, click **Account Settings (Gear Icon) > Shipping > Ship From Location**
 - Create a unique **Ship From Location** for each NetSuite **Location** that will be used during the fulfillment process by clicking **Add New Location** and entering the necessary information.
 - The **Ship From Location** name in ShipStation must match the **Location** name in NetSuite
 - *This is required for shipment details to properly post back into NetSuite*
- The ShipStation for NetSuite connector *requires* that item weights are present in the **Item** record in NetSuite
- At this time, NetSuite Integrated Shipping options are not compatible with the ShipStation for NetSuite connector
- Any NetSuite Shipping Items configured as Integrated Shipping Label options are not be supported in this integration and will result in an error message during the shipment notification process back to NetSuite.

Frequently Asked Questions

General Questions

“Am I able to connect ShipStation to NetSuite by myself, or do I have to work with an implementation partner?”

The information contained in this document enables the ShipStation for NetSuite connector to be configured without the assistance of an implementation partner/team. That said, ShipStation will happily introduce you to one of our trusted implementation partners for help with the setup of this integration, if requested.

“Do I have to install the ShipStation for NetSuite bundle in NetSuite, or can I use an existing Integration Role and User Role?”

The ShipStation for NetSuite bundle was designed to expedite the connection process for this integration and is pre-configured to ensure the best fulfillment experience. Using pre-existing Integration and User Roles can lead to unexpected results. ShipStation is unable to provide support for connections not utilizing the ShipStation for NetSuite bundle.

“Can I display the shipping rates that I see in ShipStation on my website/shopping cart?”

The ShipStation for NetSuite integration does not support in-cart rating at this time

“Are Lot Numbers and Serial Numbers supported?”

Lot Numbers and Serial Numbers must originate inside of NetSuite to be supported with our PickPackShip workflow options at this time.

“Can I partially fulfill Sales Orders with the ShipStation for NetSuite connector?”

*Yes! If the Sales Order workflow option is being used, after the NetSuite Sales Order imports into ShipStation, you have the ability to split the Sales Order into multiple shipments. When ShipStation notifies NetSuite of a partial shipment, the Sales Order status will update to **Partially Fulfilled/Pending Billing**. Once the entire Sales Order has been fulfilled, the NetSuite Sales Order will update to **Pending Billing, Billed, or Completed***

“Can I combine multiple Sales Orders or Item Fulfillment Records for the same customer in ShipStation?”

Yes, Sales Orders and Item Fulfillment Records can be combined during the fulfillment process in ShipStation. Once they are shipped, NetSuite will receive updates to all linked Sales Orders or Item Fulfillment Records. Note: Actual Shipping Cost will post to all combined Item Fulfillment Records

“Can I partially fulfill an Item Fulfillment Record with the ShipStation for NetSuite connector?”

Item Fulfillments do not support partial shipment notifications.

“How are Assembly/Kit/Group Items handled with this connector?”

If Assembly, Kit, or Group Items are being utilized within NetSuite, using a PickPackShip workflow is recommended to ensure the proper communication between ShipStation and NetSuite.

“Can I use NetSuite Integrated Shipping Label options with this connector?”

Shipping Items that are configured as Integrated Shipping Label options in NetSuite are not supported as this time and will cause the shipment notification process to fail if they are in use. It is recommended that the Integrated Shipping Label functionality is disabled for these Shipping Items, or that new Shipping Items with out Integrated Shipping Label functionality are created for use with this connector.

Questions about Import and Shipment Notification

“What happens if I cancel a Sales Order in NetSuite after it has already imported into ShipStation?”

If the Sales Order workflow option is being utilized, ShipStation will receive the status update during the next request to NetSuite for Sales Order information. Once ShipStation receives this status update, this Sales Order will be removed from the Awaiting Shipment grid.

“How about for an Item Fulfillment Record?”

*The PickPackShip flows cannot receive Item Fulfillment Record cancellation status updates from NetSuite. These Item Fulfillment Records must be manually cancelled inside of ShipStation. This can be done by selecting the **Checkbox** next the Sales Order in ShipStation and then clicking **Other Actions > Cancel***

“What shipping information is sent back to NetSuite?”

Currently, the shipment notification sent from ShipStation to NetSuite includes Shipped status, Tracking Number, Tracking URL, Shipment Cost, Location, Line Item, Item Qty

“Can ShipStation send NetSuite the Shipping Paid amount with my upcharge added in?”

*ShipStation will only send NetSuite the actual amount that was paid/invoiced while creating a shipment. This posts into the **Shipping Cost** field on the Item Fulfillment Record that was either created or updated during the fulfillment process.*

“Does cancelling a Sales Order or Item Fulfillment Record in ShipStation also cancel it in NetSuite?”

No, cancelling a Sales Order or Item Fulfillment Record in ShipStation will not update the Sales Order or Item Fulfillment Record in NetSuite to a cancelled status. ShipStation will only update NetSuite with information during the shipment notification process.

Troubleshooting

Issue: “I’m not seeing any Sales Orders/Item Fulfillment Records in ShipStation”

- Use the store refresh menu in ShipStation to manually trigger a request for new/updated Sales Order/Item Fullfillment Record data from NetSuite
- Use the store refresh menu in ShipStation to check for import errors
 - Errors will appear in red and will display the full error message if clicked
- Verify there are Sales Orders or Item Fulfillment Records present in NetSuite with a valid import status. See workflow [field mappings](#) above for valid status values for each unique workflow.
- Verify that ShipStaiton is successfully communicating with NetSuite
 - If none of the steps above result in successful order import the connection can be verified in NetSuite by clicking **Setup > Integration > Manage Integrations**
 - Click **ShipStation TBA**
 - Click **Execution Log > REST Web Services**
 - If no logs are present, or there are no recent logs, the connection between ShipStation and NetSuite is not functioning correctly
 - Please reach out to ShipStation support for further assistance
- Verify that any optional filters that were added during the configuration are correct
 - Reconfigure the connection to verify correct filter values were entered
 - In ShipStation, click **Account Settings (Gear Icon) > Selling Channels > Store Setup > Action Menu** for NetSuite Store connection > **Edit Marketplace Connection**
 - Enter **NetSuite Account ID**
 - Enter **Consumer Key & Secret**
 - Enter **Access Token Key & Secret**
 - Select workflow from **Import Fulfillments As...** dropdown menu
 - Use checkbox to define **Multiple Shipping Routes** option found in NetSuite
 - Enter any desired filter values for **Class, Department, Location**
 - Click **Continue**

- If this does not resolve the issue
 - Reconfigure the connection with no filter values present to test for all data
 - In ShipStation, click **Account Settings (Gear Icon) > Selling Channels > Store Setup > Action Menu** for NetSuite Store connection > **Edit Marketplace Connection**
 - Enter **NetSuite Account ID**
 - Enter **Consumer Key & Secret**
 - Enter **Access Token Key & Secret**
 - Select workflow from **Import Fulfillments As...** dropdown menu
 - Use checkbox to define **Multiple Shipping Routes** option found in NetSuite
 - Enter any desired filter values for **Class, Department, Location**
 - Click **Continue**
- If the issues persists after these troubleshooting steps, please reach out to ShipStation support

Issue: “I am receiving this error during Sales Order/Item Fulfillment Record Import”

Example Errors

- **An error occurred attempting to update orders: Input string '16.8' is not a valid integer. Path 'sales_orders[1].requested_fulfillments[0].items[1].quantity', line 1, position 1999.**
 - ShipStation cannot import Sales Orders/Item Fulfillment Records that contain items with “partial quantities”
 - Quantity values must be whole numbers
- **An error occurred attempting to update orders: We encountered an error communicating with Netsuite. Please attempt to refresh the store in a few minutes. (504)**
 - Refresh store after waiting a few minutes
 - If this error persists, please reach out to ShipStation support

- An error occurred attempting to update orders: 520 : {"message":"Failed to parse response body","transaction_id":"aabc13dc-42aa-46f4-bb6a-c2a4829a39bb"}
 - Confirm that the correct Consumer Key/Secret and Access Tokens were entered during the connection process
 - Confirm the Multiple Shipping Routes checkbox value in the ShipStation connection modal matches the value inside of NetSuite
 - If this error persists, please reach out to ShipStation support
- An error occurred attempting to update orders: Unauthorized : {"message":"Unauthorized","transaction_id":"84cf9fe6-947c-46da-955f-4e93d2f1b398"}
 - Confirm that the correct Consumer Key/Secret and Access Tokens were entered during the connection process
 - If this error persists, please reach out to ShipStation support

Issue: “My Sales Order/Item Fulfillment Record didn’t update after it was fulfilled in ShipStation”

If the Sales Order/Item Fulfillment Record did not update to the Shipped status after label creation in ShipStation

- Navigate to the **Shipments** page in ShipStation
- Locate the **Marketplace Notified** column
- Check the status of the Marketplace Notification
 - Shipments that have not attempted to notify will be greyed out
 - Successful Notifications will show as “Notified”
 - Unsuccessful Notifications will show as “Failed” in red
 - Click on the “Failed” icon to expand the Error Message

Example Errors

- Notification failed for Order ID: with Reason: Error while accessing a resource. Value 0.0 outside of valid min/max range for field packageweight
 - Commonly due to an item record in NetSuite not having a weight associated with the item. The ShipStation for NetSuite connector requires weights to be present in the NetSuite item record

- Notification failed for Order ID: with Reason: Error while accessing a resource. You must have 'Transactions -> Fulfill Sales Orders' edit permission to fulfill sales orders
 - Commonly due to an unsupported User Role configuration in NetSuite. Please install the ShipStation for NetSuite bundle containing the pre-configured User Role and Integration Role. Refer to the [NetSuite Instance Preparation](#) section of this document for instructions
- Error while accessing a resource. You have attempted an invalid sublist or line item operation. You are either trying to access a field on a non-existent line or you are trying to add or remove lines from a static sublist
 - Typically this message is populated when the NetSuite Shipping Item used for this order is configured as a NetSuite Integrated Shipping option. Additionally, this message could be populated when attempting to fulfill assembly/kit/group items while using the Sales Order workflow. We recommend using the PPS workflows if these item types are being utilized in NetSuite.

For additional troubleshooting/error resolution, please reach out to ShipStation support at support@shipstation.com