



Reference Guide

Integration Guide

Change Log:

Here is the change log for this guide:

<i>Date of Change:</i>	<i>Notes:</i>
1/1/2019	<i>First version as a help guide on Zendesk.</i>
3/17/2020	<i>Second version as a word document. Added request and response objects.</i>
3/18/2020	<i>Added code for GetRefunds.</i>
3/19/2020	<i>Added code for CreateOrders.</i>
3/23/2020	<i>Update CreateOrders response object.</i>
3/24/2020	<i>Added OAuth support.</i>
3/25/2020	<i>Turned document into shareable PDF</i>
3/25/2020	<i>Added solution website parameter for OAuth.</i>
6/2/2020	<i>Added custom field documentation.</i>

How do you send information:

Our tool sends data using web hooks. Our software will ping your website to request data.

How do I get started:

Here are the steps:

1. Go to Connex and make an account.
2. If you use OAuth, implement the OAuth endpoint. You will be asked to sign in or register. If you use basic authentication, select custom store. Enter your credentials. In the solution website field, enter your base URL. This is the location of our end points on your server. If your website was `https://www.example.com`, enter `https://www.example.com/connex`. The end point is customizable.
3. Once you reach our install wizard, stop.
4. On the left, click orders and click order previewer.
5. Enter some parameters for the sync. If you see no data, ensure your parameters are correct. If there are errors, click log on the top right of Connex. Click action and export to excel for more information.

Here is how the fields match:

Location:	Field Name:	Notes:
Home	Order Date	First version as a help guide on Zendesk.
Home	Number	Second version as a word document. Added request and response objects.
Home	Payment	Added code for GetRefunds.
Home	Store	Added code for CreateOrders.
Home	Status	Update CreateOrders response object.
Details	Sku	Added OAuth support.
Details	Name	Turned document into shareable PDF
Details	Quantity	
Details	Price Excluding Tax	
Details	Price Including Tax	
Shipping	Shipping Excluding Tax	Added solution website parameter for OAuth.
Shipping	Shipping Including Tax	
Shipping	Method	
Shipping	Cost	
Shipping	Date	
Shipping	TrackingNumber	
Addresses	First row is billing	
Addresses	Second row is billing	
QB	Amount Paid	
QB	Balance Due	

Authentication Methods :

What authentication methods do you support?

Connex supports basic authentication and OAuth. We recommend basic authentication if you have a custom website. To use basic authentication, follow these steps:

1. Login to Connex.
2. Add a new connection.
3. Choose custom store.
4. Enter a user name and password. Your plugin will validate these credentials in the authenticate method.

If you have a commercial solution, we recommend OAuth. We recommend sending an OAuth token that never expires. Here is a list of parameters:

Field Name	Type	Notes:
<i>Solution</i>	<i>String</i>	<i>Name of your syncing solution.</i>
<i>AccessToken</i>	<i>String</i>	<i>Access token used for OAuth.</i>
<i>SolutionWebsite</i>	<i>String</i>	<i>Base URL for the web hooks. If you wanted to call GetOrders, an example URL would be https://www.example.com/connex/get-orders</i>

Here are the steps:

From your software, send this URL: <https://www.connexforquickbooks.com/integrations/custom?solutionName=X&accessToken=Y>. If you use Connex for QuickBooks Online, replace connexforquickbooks.com with connexonlinesync.com. Connex will redirect the user to sign in or register for an account.

Connex will send the user to the installation wizard.

Endpoints:

Authenticate:

This method is called, before our tool syncs information. The method's purpose is to see if the user name and password is correct. The method returns OK, if successful:

If you use basic authentication, here is a sample call:

Sample Call:

```
GET https://x.testingdevelopmentprocess.com/api/v1/authenticatetoweb-service
HTTP/1.1
Accept: application/json
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.36
Authorization: Basic X
Host: x.testingdevelopmentprocess.com
```

If you use OAuth, here is a sample call:

Sample Call:

```
GET https://x.testingdevelopmentprocess.com/api/v1/authenticatetoweb-service
HTTP/1.1
Accept: application/json
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.36
Authorization: Bearer X
```

Sample Response:

```
HTTP/1.1 200 OK
Server: nginx/1.15.6
Date: Tue, 17 Mar 2020 14:26:34 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
OK
```

CreateOrders

Syncs orders from QuickBooks to your website. For field descriptions, view the *GetOrders* method. Here is the response object:

Location:	Field Name:	Type:	Notes:
JMAGetOrdersResponse	SyncedSuccessfully	Int	Enter 1 if synced successfully
JMAGetOrdersResponse	SyncedUnSuccessfully	Int	Enter 1 if synced unsuccessfully
JMAGetOrdersResponse	Message	String	Add an error message, if sync was unsuccessful.
JMAGetOrdersResponse	SeverityMessage	Enum	Enter Error for failure and Info for success.
JMAGetOrdersResponse	OrderId	String	Order number from your selling channel.

Sample Call:

Syncs orders from QuickBooks to your website. For field descriptions, view the `GetOrders` method. Here is the response object:

```
POST https://x.testingdevelopmentprocess.com/api/v1/createorders HTTP/1.1
Accept: application/json
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.36
Authorization: Basic x
Host: x.testingdevelopmentprocess.com
```

```
{
  "Orders": [
    {
      "Id": 0,
      "OrderNumber": "web000077",
      "CreditCardName": "MasterCard",
      "Total": "3.92",
      "TotalTax": 0,
      "CreationDate": "03/16/2020 11:19:45 AM",
      "ModifiedDate": "03/16/2020 11:19:45 AM",
      "OrderDetails": [
        {
          "Id": 3,
          "OverrideUnitOfMeasure": "",
          "Name": "Dang Gui Shao Yao San",
          "PriceExclTax": 0.28,
          "PriceInclTax": 0.28,
          "Quantity": "14",
          "Sku": "Dang Gui Shao Yao San",
          "UnitOfMeasure": "gram",
          "Product": {
            "Id": 102,
            "Color": null,
            "Sku": "Dang Gui Shao Yao San",
            "Amount": 3.92,
            "Name": "Dang Gui Shao Yao San",
            "Description": null,
            "Size": null,
            "IsInventory": true
          }
        }
      ],
      "AmountPaid": 0,
      "BalanceDue": 0,
      "Class": null,
      "CurrencyCode": "USD",
      "DiscountExclTax": -1,
      "DiscountInclTax": -1,
      "DueDate": "2/1/2020",
    }
  ]
}
```

```

    "IsTaxInclusive": false,
    "Note": "These notes appear in QuickBooks",
    "OrderStatus": "Shipped",
    "ShippingDate": "1/1/2020 04:00:00 PM",
    "ShippingExclTax": 4.25,
    "ShippingMethod": "Shipping Charge:Shipping Charge (under 275g)",
    "ShippingInclTax": 4.25,
    "StoreName": "Golden Cabinet",
    "TaxRate": 0,
    "TotalExclTax": 100,
    "TotalInclTax": 106.25,
    "TotalShippingCosts": 0,
    "TrackingNumber": "",
    "TotalMerchantFees": 0,
    "MerchantFee": 0,
    "BillingAddress": {
      "FirstName": "Stephen",
      "LastName": "Bond (admin)",
      "FullName": "Stephen Bond (admin)",
      "Company": "Stephen TestPatient20200206"
    },
    "ShippingAddress": {
      "Address1": "4102 SE Hawthorne Blvd",
      "Address2": "",
      "City": "Portland",
      "Email": "stephen+admin@goldencabinetherbs.com",
      "FirstName": "Stephen",
      "LastName": "Bond (admin)",
      "FullName": "Stephen Bond (admin)",
      "PhoneNumber": "6103298019",
      "PostalCode": "97215",
      "RegionName": "OR",
      "TwoLetterIsoCode": "US",
      "Formula Name": "FormulaTest-2",
      "License Number": null,
      "CustomFields": [
        {
          "Name": "Formula Name",
          "Value": "FormulaTest-2",
        },
        {
          "Name": "Doses Per Day",
          "Value": "Four times per day",
        }
      ]
    }
  }
}
]
}

```

Sample Response:

```
HTTP/1.1 200 OK
Server: nginx/1.15.6
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Cache-Control: no-cache, private
Date: Tue, 17 Mar 2020 14:29:32 GMT
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 58
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
{
  "SyncedSuccessfully": 1,
  "SyncedUnsuccessfully": 0,
  "Messages": [
    {
      "HelpLink": "",
      "ErrorWriteToLogged": "2020-03-23T12:13:47.1734908Z",
      "ID": 0,
      "UserName": "",
      "Website": "",
      "Severity": 0,
      "Message": "Order 1 from QuickBooks has synced successfully!",
      "OrderId": "1",
      "Id": 0,
      "SeverityMessage": "Info"
    },
    {
```

```

    "HelpLink": "",
    "ErrorWriteToLogged": "2020-03-23T12:13:48.8555299Z",
    "ID": 0,
    "UserName": "",
    "Website": "",
    "Severity": 1,
    "Message": "Error syncing order 2. Contact JMA: Invalid ShipTo Address",
    "OrderId": "",
    "Id": 0,
    "SeverityMessage": "Error"
  }
]
}

```

GetOrders

Returns a list of sales from your website. Here is a list of fields:

Location:	Field Name:	Type:	Notes:
JMAOrder	OrderNumber	String	Order number on sale
JMAOrder	CreationDate	Date-Time	Date sale was placed.
JMAOrder	ShippingDate	Date-Time	Date sale was shipped.
JMAOrder	StoreName	String	If your solution collects sales from various channels, such as Amazon, enter the store's name.
JMAOrder	Total	Decimal	Grand total of sale.
JMAOrder	TotalTax	Decimal	Total tax on sale.
JMAOrder	ShippingExclTax	Decimal	Total shipping minus tax.

JMAOrder	ShippingInclTax	decimal	Total shipping plus tax.
JMAOrder	TrackingNumber	string	Shipping tracking number
JMAOrder	ShippingMethod	string	Shipping method on the sale.
JMAOrder	TotalShippingCosts	decimal	Total cost that the seller must pay for shipping.
JMAOrder	TotalMerchantFees	decimal	Total merchant fees on the sale.
JMAOrder	CurrencyCode	string	Two letter currency code on sale.
JMAOrder	CustomFields	CustomField	CustomField
JMAOrder	AmountPaid	decimal	If order is fully paid, should equal total.
JMAOrder	BalanceDue	decimal	If order is partially paid, should equal the total minus amount paid.
Address	FirstName	string	
Address	LastName	string	
Address	Company	string	
Address	City	string	
Address	Region	string	Two letter state code.
Address	PostalCode	string	
Address	Phone	string	
Address	Email	string	
Address	Address1	string	
Address	Address2	string	
Address	Address3	string	
JMAOrderDetail	PriceExclTax	decimal	string
JMAOrderDetail	PriceInclTax	decimal	string
JMAOrderDetail	Quantity	decimal	
JMAOrderDetail	Sku	string	
JMAOrderDetail	Name	string	Name of product
JMAOrderDetail	UnitOfMeasure	string	Unit of measure, like Bx, if none supplied

Here is a list of parameters that we can send:

Field Name:	Notes:	Format:
<i>date_modified_min</i>	<i>If querying sales by the date modified in your store, this is the starting date.</i>	<i>mm/dd/yyyy hh:mm</i>
<i>date_modified_max</i>	<i>If querying sales by the date modified in your store, this is the end date.</i>	<i>mm/dd/yyyy hh:mm</i>
<i>date_created_min</i>	<i>If querying sales by the date they were created, this is the start date.</i>	<i>mm/dd/yyyy hh:mm</i>
<i>date_created_max</i>	<i>If querying sales by the date they were created, this is the end date.</i>	<i>mm/dd/yyyy hh:mm</i>
<i>storeName</i>	<i>If your website has multiple selling channels, you can filter orders by the channel name.</i>	<i>Amazon,eBay</i>
<i>/orders/orderNumber</i>	<i>If you want to filter orders by a status, this is the order status.</i>	<i>shipped,fulfilled</i>

Sample Call:

```
GET https://x.testingdevelopmentprocess.com/api/v1/orders?date_created_min=3%2f10%2f2020+12%3a00%3a00+AM&date_created_max=3%2f17%2f2020+11%3a59%3a00+PM HTTP/1.1
```

```
Accept: application/json
```

```
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36
```

```
Authorization: Basic x
```

```
Host: x.testingdevelopmentprocess.com
```

Sample Response:

```
HTTP/1.1 200 OK
Server: nginx/1.15.6
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Cache-Control: no-cache, private
Date: Tue, 17 Mar 2020 14:29:32 GMT
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 58
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
{
  "Orders": [
    {
      "Id": 77,
      "OrderNumber": "web000077",
      "CreditCardName": "chargeUserCard",
      "Total": "3.92",
      "TotalTax": 0,
      "CreationDate": "03/16/2020 11:19:45 AM",
      "ModifiedDate": "03/16/2020 11:19:45 AM",
      "OrderDetails": [
        {
          "Id": 3,
          "OverrideUnitOfMeasure": "",
          "Name": "Dang Gui Shao Yao San",
          "PriceExclTax": 0.28,
          "PriceInclTax": 0.28,
          "Quantity": "14",
          "Sku": "Dang Gui Shao Yao San",
          "UnitOfMeasure": "gram",
          "Product": {
            "Id": 102,
            "Color": null,
            "Sku": "Dang Gui Shao Yao San",
            "Amount": 3.92,
```

```
    "Name": "Dang Gui Shao Yao San",
    "Description": null,
    "Size": null,
    "IsInventory": true
  }
}
],
"AmountPaid": 0,
"BalanceDue": 0,
"Class": null,
"CurrencyCode": "USD",
"DiscountExclTax": -1,
"DiscountInclTax": -1,
"DueDate": "12/31/1969 04:00:00 PM",
"IsTaxInclusive": false,
"Note": " / ",
"OrderStatus": "Shipped",
"ShippingDate": "12/31/1969 04:00:00 PM",
"ShippingExclTax": 4.25,
"ShippingMethod": "Shipping Charge:Shipping Charge (under 275g)",
"ShippingInclTax": 4.25,
"StoreName": "Golden Cabinet",
"TaxRate": 0,
"TotalExclTax": 100,
"TotalInclTax": 106.25,
```

```
“TotalShippingCosts”: 0,  
“TrackingNumber”: “”,  
“TotalMerchantFees”: 0,  
“MerchantFee”: 0,  
“BillingAddress”: {  
  “FirstName”: “Stephen”,  
  “LastName”: “Bond (admin)”,  
  “FullName”: “Stephen Bond (admin)”,  
  “Company”: “Stephen TestPatient20200206”  
},  
“ShippingAddress”: {  
  “Address1”: “4102 SE Hawthorne Blvd”,  
  “Address2”: “”,  
  “City”: “Portland”,  
  “Email”: “stephen+admin@goldencabinetherbs.com”,  
  “FirstName”: “Stephen”,  
  “LastName”: “Bond (admin)”,  
  “FullName”: “Stephen Bond (admin)”,  
  “PhoneNumber”: “6103298019”,  
  “PostalCode”: “97215”,  
  “RegionName”: “OR”,  
  “TwoLetterIsoCode”: “US”,  
  “VATNumber”: null  
  “Formula Name”: “FormulaTest-2”,  
  “License Number”: null,
```

```
    "Patient Name": "Stephen TestPatient20200206",
    "Patient Of": "Stephen Bond (admin)",
    "Scoops per Dose": "1 heaping scoop (2.5g)",
    "Doses Per Day": "Four times per day",
    "Refills": "6",
    "Shipping Charge": "Shipping Charge(under 275g)"
  }
}
],
"Payments": [
  {
    "PaymentMethodId": 77,
    "CreditCardName": "Visa",
    "Id": 77,
    "OrderId": 77,
    "Order": {
      "OrderNumber": 77
    },
    "PaymentAmount": 0,
    "PaymentDate": "12/31/1969 04:00:00 PM",
    "ReferenceNumber": "12345",
    "SkipPayment": false,
    "OrderNumber": 77,
    "ModifiedDate": "12/31/1969 04:00:00 PM",
    "Note": "Payment for order 123"
  }
]
}
```

GetRefunds

Returns a list of refunds. Please look at the *GetOrders* for a list of field names. Partial and full refunds are supported. The order number must match the sale order number.

Field Name	Notes:	Format
<code>date_created_min</code>	Begin date range of when re-funds were created.	MM/DD/YYYY HH:MM
<code>date_created_max</code>	End date range when refunds were created.	MM/DD/YYYY HH:MM

Sample Call:

```
GET https://x.testingdevelopmentprocess.com/api/v1/refunds?date_created_min=3%2f10%2f2020+12%3a00%3a00+AM&date_created_max=3%2f17%2f2020+11%3a59%3a00+PM HTTP/1.1
```

```
Accept: application/json
```

```
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36
```

```
Authorization: Basic x
```

```
Host: x.testingdevelopmentprocess.com
```

Sample Response:

```
HTTP/1.1 200 OK
Server: nginx/1.15.6
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Cache-Control: no-cache, private
Date: Tue, 17 Mar 2020 14:29:32 GMT
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 58
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
```

```
{
  "CancelledOrders": [
    {
      "Id": 77,
      "OrderNumber": "web000077",
```

```
“CreditCardName”: “chargeUserCard”,
“Total”: “3.92”,
“TotalTax”: 0,
“CreationDate”: “03/16/2020 11:19:45 AM”,
“ModifiedDate”: “03/16/2020 11:19:45 AM”,
“OrderDetails”: [
  {
    “Id”: 3,
    “OverrideUnitOfMeasure”: “”,
    “Name”: “Dang Gui Shao Yao San”,
    “PriceExclTax”: 0.28,
    “PriceInclTax”: 0.28,
    “Quantity”: “14”,
    “Sku”: “Dang Gui Shao Yao San”,
    “UnitOfMeasure”: “gram”,
    “Product”: {
      “Id”: 102,
      “Color”: null,
      “Sku”: “Dang Gui Shao Yao San”,
      “Amount”: 3.92,
      “Name”: “Dang Gui Shao Yao San”,
      “Description”: null,
      “Size”: null,
      “IsInventory”: true
    }
  }
]
```

```
],  
  "AmountPaid": 0,  
  "BalanceDue": 0,  
  "Class": null,  
  "CurrencyCode": "USD",  
  "DiscountExclTax": -1,  
  "DiscountInclTax": -1,  
  "DueDate": "12/31/1969 04:00:00 PM",  
  "IsTaxInclusive": false,  
  "Note": " / ",  
  "OrderStatus": "Shipped",  
  "ShippingDate": "12/31/1969 04:00:00 PM",  
  "ShippingExclTax": 4.25,  
  "ShippingMethod": "Shipping Charge:Shipping Charge (under 275g)",  
  "ShippingInclTax": 4.25,  
  "StoreName": "Golden Cabinet",  
  "TaxRate": 0,  
  "TotalExclTax": 100,  
  "TotalInclTax": 106.25,  
  "TotalShippingCosts": 0,  
  "TrackingNumber": "",  
  "TotalMerchantFees": 0,  
  "MerchantFee": 0,  
  "BillingAddress": {  
    "FirstName": "Stephen",
```

```
    "LastName": "Bond (admin)",
    "FullName": "Stephen Bond (admin)",
    "Company": "Stephen TestPatient20200206"
  },
  "ShippingAddress": {
    "Address1": "4102 SE Hawthorne Blvd",
    "Address2": "",
    "City": "Portland",
    "Email": "stephen+admin@goldencabinetherbs.com",
    "FirstName": "Stephen",
    "LastName": "Bond (admin)",
    "FullName": "Stephen Bond (admin)",
    "PhoneNumber": "6103298019",
    "PostalCode": "97215",
    "RegionName": "OR",
    "TwoLetterIsoCode": "US",
    "VATNumber": null
  },
  "CustomFields": {
    "Formula Name": "FormulaTest-2",
    "License Number": null,
    "Patient Name": "Stephen TestPatient20200206",
    "Patient Of": "Stephen Bond (admin)",
    "Scoops per Dose": "1 heaping scoop (2.5g)",
    "Doses Per Day": "Four times per day",
```

```

    "Refills": "6",
    "Shipping Charge": "Shipping Charge(under 275g)"
  }
}
],
"Payments": []
}

```

Inventory

This call syncs orders from QuickBooks to your website.

Here are a list of fields:

Field Name	Notes:	Format
Price	Price of product in QuickBooks.	decimal
StockQuantity	Stock level in QuickBooks	integer
SKU	SKU from website. Should match the QuickBooks item name field.	string
Warehouse	Warehouse in QuickBooks that holds inventory. Usually empty.	string

Sample Call:

```
PUT https://www.mysite.com/api/v1/inventory?sku=ABC123
```

```
Accept: application/json
```

```
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
```

```
Chrome/41.0.2228.0 Safari/537.36
```

```
Authorization: Basic x
```

```
Host: x.testingdevelopmentprocess.com
```

```
{"price": 100, "stock_quantity": 1}
```

```
[
  {
    "AssemblyItems": [],
    "Id": 0,

```

```
    "DoNotSync": false,
    "CustomFields": {},
    "JMACustomFields": [],
    "ProductGuid": "00000000-0000-0000-0000-000000000000",
    "Sku": "Test",
    "HandlingTime": 0,
    "Amount": 1,
    "MSRP": 0,
    "PurchaseCost": 0,
    "StockQuantity": 1,
    "IsDownload": false,
    "IsService": false,
    "Images": [],
    "Manufacturers": [],
    "Weight": 0,
    "IsTaxable": false,
    "IsEnabled": true,
    "Categories": [],
    "DateUpdated": "0001-01-01T00:00:00",
    "DateInserted": "0001-01-01T00:00:00",
    "ProductWarehouses": [],
    "IsInventory": true,
    "CustomPrice": 0
  },
  {
```

```
    "AssemblyItems": [],
    "Id": 0,
    "DoNotSync": false,
    "CustomFields": {},
    "JMACustomFields": [],
    "ProductGuid": "00000000-0000-0000-0000-000000000000",
    "Sku": "Test2",
    "HandlingTime": 0,
    "Amount": 1,
    "MSRP": 0,
    "PurchaseCost": 0,
    "StockQuantity": 1,
    "IsDownload": false,
    "IsService": false,
    "Images": [],
    "Manufacturers": [],
    "Weight": 0,
    "IsTaxable": false,
    "IsEnabled": true,
    "Categories": [],
    "DateUpdated": "0001-01-01T00:00:00",
    "DateInserted": "0001-01-01T00:00:00",
    "ProductWarehouses": [],
    "IsInventory": true,
    "CustomPrice": 0
  }
]
```

For the response, here are a list of fields

Field Name	Notes:	Format
<i>date_created_min</i>	<i>Begin date range of when re-funds were created.</i>	<i>MM/DD/YYYY HH:MM</i>
<i>date_created_max</i>	<i>End date range when refunds were created.</i>	<i>MM/DD/YYYY HH:MM</i>

Sample Response:

HTTP/1.1 200 OK

Server: nginx/1.15.6

Date: Tue, 17 Mar 2020 14:26:34 GMT

Content-Type: text/html; charset=UTF-8

Transfer-Encoding: chunked

Connection: keep-alive

Vary: Accept-Encoding

X-Frame-Options: SAMEORIGIN

X-XSS-Protection: 1; mode=block

X-Content-Type-Options: nosniff

```
{
  "ProductSkusErrors": {
    "Test2": "There is no SKU called test 2."
  },
  "ProductSyncSuccess": [
    {
      "AssemblyItems": [],
      "Id": 0,
      "DoNotSync": false,
      "CustomFields": {},
      "JMACustomFields": [],
      "ProductGuid": "00000000-0000-0000-0000-000000000000",
      "Sku": "Test",
      "HandlingTime": 0,
      "Amount": 1,
    }
  ]
}
```

```
    "MSRP": 0,  
    "PurchaseCost": 0,  
    "StockQuantity": 1,  
    "IsDownload": false,  
    "IsService": false,  
    "Images": [],  
    "Manufacturers": [],  
    "Weight": 0,  
    "IsTaxable": false,  
    "IsEnabled": true,  
    "Categories": [],  
    "DateUpdated": "0001-01-01T00:00:00",  
    "DateInserted": "0001-01-01T00:00:00",  
    "ProductWarehouses": [],  
    "IsInventory": true,  
    "CustomPrice": 0  
  }  
]  
}
```