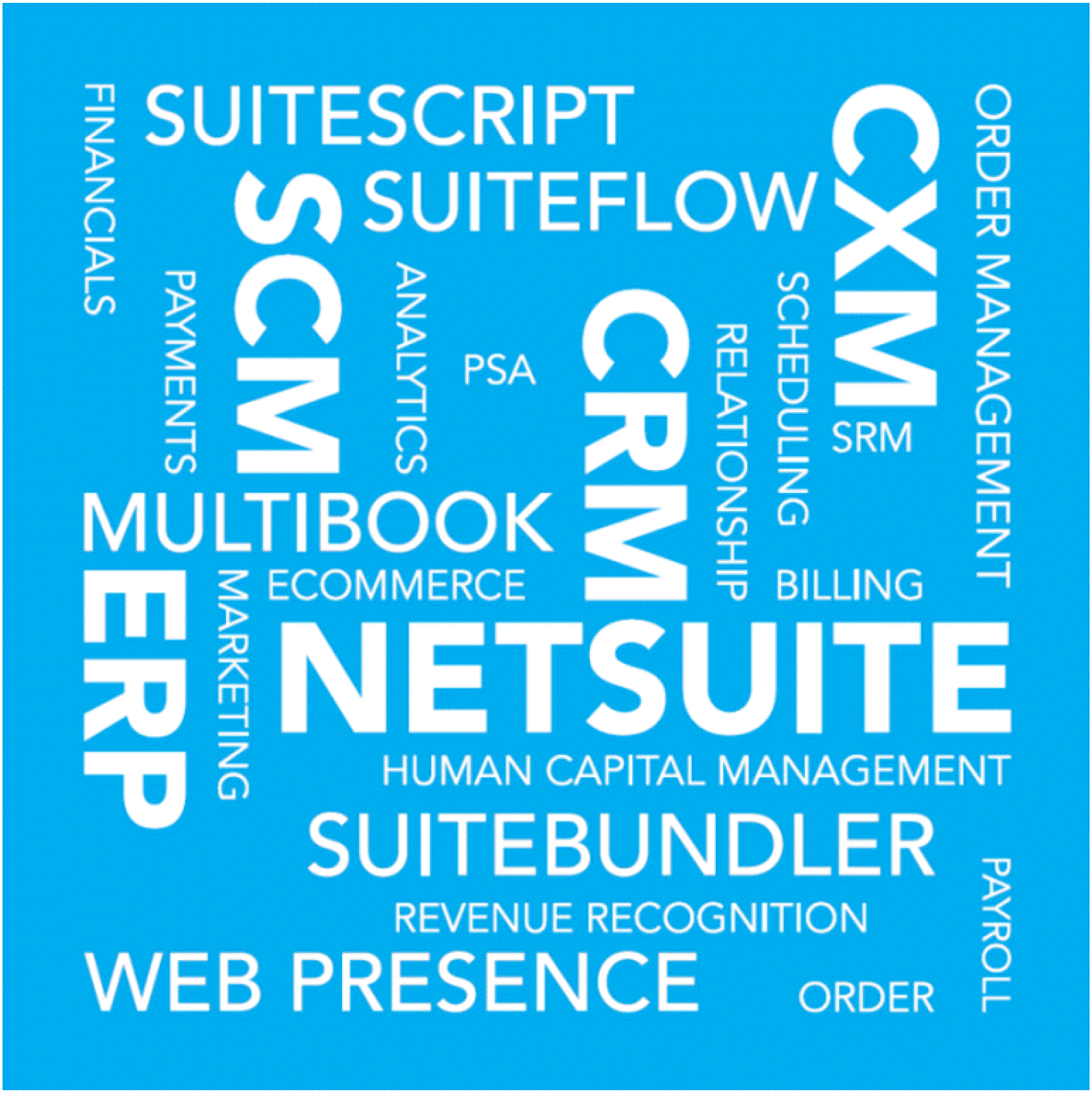
SuiteCommerce API Documentation



April 11, 2018

2018.1

and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Sample Code

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided "as is” and “as available”, for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

No Excessive Use of the Service

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer’s use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle’s other customers, Oracle may slow down or throttle Customer’s excess use until such time that Customer’s use stays within reasonable limits. If Customer’s particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer’s actual usage pattern.

Beta Features

Oracle may make available to Customer certain features that are labeled “beta” that are not yet generally available. To use such features, Customer acknowledges and agrees that such beta features are subject to the terms and conditions accepted by Customer upon activation of the feature, or in the absence of such terms, subject to the limitations for the feature described in the User Guide and as follows: The beta feature is a prototype or beta version only and is not error or bug free and Customer agrees that it will use the beta feature carefully and will not use it in any way which might result in any loss, corruption or unauthorized access of or to its or any third party’s property or information. Customer must promptly report to Oracle any defects, errors or other problems in beta features to support@netsuite.com or other designated contact for the specific beta feature. Oracle cannot guarantee the continued availability of such beta features and may substantially modify or cease providing such beta features without entitling Customer to any refund, credit, or other compensation. Oracle makes no representations or warranties regarding functionality or use of beta features and Oracle shall have no liability for any lost data, incomplete data, re-run time, inaccurate input, work delay, lost profits or adverse effect on the performance of the Service resulting from the use of beta features. Oracle’s standard service levels, warranties and related commitments regarding the Service shall not apply to beta features and they may not be fully supported by Oracle’s customer support. These limitations and exclusions shall apply until the date that Oracle at its sole option makes a beta feature generally available to its customers and partners as part of the Service without a “beta” label.

Table of Contents

Item Search API

ⓘ **Applies to:** SuiteCommerce Web Stores

Web and mobile app developers can use the Item Search API to access attributes of items in the webstore. The Item Search API is a REST API with JSON data objects. It supports:

- Facet filtering
- Keyword searches
- Sorting, pagination, and more

ⓘ **Note:** To use the Item Search API, you must have a SuiteCommerce Advanced web site, and you must have configured item search settings for your site. For more information, see the help topic Search Settings.

⚠ **Important:** The **Mont Blanc** release of SuiteCommerce Advanced requires a patch to ensure that the Item Search API response is cached in the CDN. See the help topic Item Search API Response Data not Cached for details.

See the following topics for more information about the Item Search API:

- Items Available to the Item Search API – This topic describes the factors that affect whether an item can be retrieved by the Item Search API.
- The Base URL – The base URL is the URL sent in the HTTP request to the Item Search API in order to retrieve a list of items. This topic describes the structure of the base URL.
- Item Search API Input Parameters – This topic describes all the input parameters that can be used in query strings to retrieve item record attributes.
- Sample Item Search Query and Results – This topic provides sample Item Search API queries and descriptions of response data.

Items Available to the Item Search API

ⓘ **Applies to:** SuiteCommerce Web Stores

The Item Search API returns data from all item records that have been indexed. For more information about the search index, see the help topic Search Index.

The Item Search API returns the items depending on the item availability in a given subsidiary at the time of indexing. Therefore, items are retuned only if the following conditions are met:

- If the Subsidiaries feature is enabled for your account, the item should be associated with a subsidiary that is online in the website.
- The item should be associated with an income account.
- The Display in Web Site box should be selected on the item record.
- The Inactive box should not be selected on the item record.
- The item should be assigned to a Site Builder category, or the Show Uncategorized Items box should be selected on the Web Site Setup page.
- The item should be in stock, or the selected Out of Stock Behavior option should display the item even when out of stock.
- The item type should not be Other Charge Item or Discount Item.
- The item price should be a numeric value. For example, some item types such as Discounts can have prices in percentage (10% off instead of £10 off).

Although all items configured for search are indexed, a maximum of 10K results are returned in a single search API call. However, any indexed item can be returned by a specific API call to that item using search parameters such as the item `id` (see Item Search API Input Parameters).

> ⓘ **Note:** All item records flagged to display in the web site are accessible to the API. The Item Search API does not provide authentication. The data retrieved by this service is exposed to public traffic on the internet.

The Base URL

ⓘ **Applies to:** SuiteCommerce Web Stores

The Item Search API, available with SuiteCommerce Advanced, is typically consumed by an AJAX client. You can use a client-side JavaScript framework like JQuery to send an HTTP request to the API and get the JSON response back. The URL is accessible by any client that supports HTTP and JSON.

You can retrieve results for a particular item search by sending an HTTP GET request. The Item Search API has the following format:

```
http://www.mywebstore.com/api/items
```

The base URL is described below:

- **Custom Domain** – A unique URL is required. This must be a custom domain, for example, **www.mywebstore.com**.
- **Items** – This API is designed to query item records in your NetSuite account. Note that **/items** in the URL is part of the path. It is required. You cannot assign a value to it.
- **Parameters** – These are typically name/value pairs, such as **fieldset=details**. You can use parameters defined by NetSuite along with values from your account. You can use multiple parameters in an Item Search API request by separating them with an ampersand (&). For more information, see Item Search API Input Parameters. See also, Sample Item Search API Query.

JSONP

JSONP (or JSON with padding) wraps the JSON response from API requests in a JavaScript variable that can be included in a script tag to enable cross-domain AJAX requests. Web developers can use `callback` as the input parameter, and then create the variable name. For more information about the jQuery API, visit api.jquery.com/jQuery.getJSON/. To learn more about input parameters supported by the Item Search API, see Item Search API Input Parameters.

Item Search API Input Parameters

ⓘ **Applies to:** SuiteCommerce Web Stores

Use the input parameters listed below to construct query strings that return item record attributes in a JSON response. You can use multiple parameters by separating them with an ampersand (&). The Item Search API is supported on both the shopping and checkout domains.

Use the input parameters listed below to create a query string for the Item Search API:

```
http://www.mywebstore.com/api/items?{parameters}
```

Parameter	Example
c (optional)	/api/items?c=1234567
callback (optional)	/api/items?callback=mycallback
commercecategoryid (optional)	/api/items?commercecategoryid=12345
commercecategoryurl (optional)	/api/items?commercecategoryurl=%2Ffoo%2Fbar%2Fbaz
commercecategoryname (optional)	/api/items?commercecategoryname=Summer%20Sale
correlateditems_fieldset (optional)	/api/items?fieldset=search&correlateditems_fieldset=mini
currency (optional)	/api/items?currency=EUR
debug (optional)	/api/items?debug=true&fieldset
facet.exclude (optional)	/api/items?include=facets&fieldset=search&facet.exclude=color,custitem_categories
facet ranges (optional)	/api/items?pricelevel5.ranges=1-9.99,10-19.99,20-*
facet range filters (optional)	/api/items?pricelevel5.from=10&pricelevel5.to=20
facets (optional)	/api/items?color=red,blue&gender=women
fields (optional)	/api/items?fields=internalid,name
fieldset (optional)	/api/items?fieldset=item-detail-view
id (optional)	/api/items?id=123
include (optional)	/api/items?include=facets
language (optional)	/api/items?country=ES&language=es
limit (optional)	/api/items?limit=10
matrixchilditems_fieldset (optional)	/api/items?fieldset=search&matrixchilditems_fieldset=mini
n (optional)	/api/items?n=3
offset (optional)	/api/items?limit=10&offset=10
pricelevel (required)	/api/items?pricelevel=5
q (optional)	/api/items?q=jeans

Parameter	Example
relateditems_fieldset (optional)	/api/items?fieldset=search&relateditems_fieldset=mini
region (optional)	/api/items?region=4
sort (optional)	/api/items?sort=storedisplayname:asc
url (optional)	/api/items?url=Vok-500-Cellular-Phone

The information below includes a description of each query parameter you can use with the Item Search API.

c (optional)

Specifies the NetSuite account.

```
/api/items?c=3925062
```

callback (optional)

Defines the JavaScript callback function to wrap the response in. Used for JSONP responses. Use callback as the parameter, and you can create the variable name. Note the example below:

```
/api/items?callback=mycallback
```

commercecategoryid (optional)

Specifies items by a given Commerce Category ID. See Sample Search Query for Returning Categories for an example.

commercecategoryurl (optional)

Specifies items by a given Commerce Category URL. See Sample Search Query for Returning Categories for an example.

commercecategoryname (optional)

Specifies items by a given Commerce Category Name. This parameter is dependent on the Commerce Category facet field being defined in the Web Site Setup record. See Sample Search Query for Returning Categories for an example.

country (optional)

Specifies the country associated with the language in an API request. For use with the language parameter to affect search results displayed in an account with multiple languages. For more information, see language (optional).

correlateditems_fieldset (optional)

Specifies the field set that overrides the reserved `correlateditems` field set. The `correlateditems` field set is used by the `correlateditems_detail` synthetic field by default to retrieve the item fields of all the correlated items.

In the following example, the default `correlateditems` field set is overridden with a field set named `mini` using the Item Search API parameter `correlateditems_fieldset`.

```
/api/items?fieldset=search&correlateditems_fieldset=mini
```

currency (optional)

Specifies the currency in an API request. In an account that uses the Multi-Currency feature, pass the ISO Country Code to specify the currency format in which you want to display prices.

> **Note:** If you are using the Multiple Currencies feature, ensure that you select the correct default locale on the currency record to avoid any errors related to currency mismatch. For example, select New Zealand (English) as the default locale for New Zealand Dollar. For more information, see the help topic Creating Currency Records.

debug (optional)

Prints the API request parameters in the JSON response data.

```
/api/items?debug=true&fieldset
```

facet.exclude (optional)

Excludes fields from faceting in Item Search API calls. This request parameter accepts comma delimited field names. Excluding facet fields that are not essential can help to speed up API calls. In the example below, the API call specifies a facet value (custitem_categories=Kitchen-Utensils) for filtering, while excluding the field, custitem_categories from faceting.

```
/api/items?include=facets&fieldset=search&language=en&country=US&currency=USD&pricelevel=5&c=12
34567&n=2&custitem_categories=Kitchen-Utensils&sort=pricelevel5&asc&limit=24&offset=0&c=1234567
&facet.exclude=custitem_categories
```

facet ranges (optional)

Defines range buckets for numeric facets. You can add the ranges suffix on the field ID:

```
/api/items?pricelevel5.ranges=1-9.99,10-19.99,20-*
```

The example above returns an array of range buckets with the number of items in each bucket. In this case, 1 to 9.99, 10 to 19.99, and 20 or more. Note that facet ranges can only be used for numeric fields such as price. Facet ranges do not filter, but instead show additional data in the facet section of the JSON response pertaining to how many items are in each range.

facet range filters (optional)

Specifies a range of items to be returned in query results, and can only be used with numeric facet fields such as price. Use this parameter to specify a range of values to search for on item records. Use the suffix `from` and `to` on the field ID to perform the search:

```
/api/items?pricelevel5.from=10&pricelevel5.to=20
```

The example above returns all items with a price between 10 and 20 inclusive. Both `from` and `to` are required, and excluding one will result in an error.

> **Note:** You can only use the field ID of the facet in facet range filters, and not the URL component of the facet.

facets (optional)

Specifies the facets and facet values that should be returned. This parameter returns the item IDs associated with the facets specified. You can add a facet filter to the API request, followed by the facet values you want to use as the filter. For example:

```
/api/items?color=red,blue&gender=women
```

If special characters are included in the facet value you want to specify, you can call the search API requesting all facets (for example, `/api/items?pricelevel=5&include=facets`), and then use the appropriate value that is contained in the response to construct your API query.

> **Note:** If you create a custom list intended for use as a facet, do not include a comma in any of the list values. In an API request, the comma is interpreted as a separator between multiple options.

fields (optional)

Specifies the item record fields returned in response data. Your request can include a valid field ID, or multiple field IDs separated by commas to return multiple fields.

```
/api/items?fields=custitem1,upccode
```

fieldset (optional)

Specifies the field set in an API request. Returns all of the fields associated with a field set. When the Commerce Category field is defined in the Web Site Setup record field sets, the Commerce Category details for a given item are returned.

```
/api/items?fieldset=details
```

id (optional)

Specifies the item record. To get information for a specific item, your request can contain the item record ID. To return multiple items in response data, list multiple item IDs separated by commas. If the item record ID is not valid, then an error is returned.

include (optional)

Specifies an item attribute in the API request. Use this parameter to incorporate additional data in the JSON response. For example, the sample query below includes facet fields in the JSON response.

```
/api/items?include=facets
```

language (optional)

Specifies the translated text you set up in NetSuite. A shopper on your web site can use his or her local language to search and browse products on your site. Note that the translated text that shoppers see in product views is based on the translated text you setup on item records. JSON response data includes item attributes and field set fields in the languages you have set up in your NetSuite account. Note that you must add the corresponding fields to a field set and also check that the field set is included in the Reference ShopFlow SuiteApp. Specify the country parameter as well as the language parameter. All ISO country and language codes are valid, if the language is set up for use on your web site. To set up your web site to support multiple languages, see the help topic Setting Up a Site for Multiple Languages.

To affect the language displayed in a product view, the site template must be localized. For more information, see the help topic Localization.

This example displays search results in Spanish:

```
/api/items?country=ES&language=es
```

limit (optional)

Specifies the number of items displayed in search results. Use in combination with offset (optional).

Note: Use a value only between 0 to 100.

matrixchilditems_fieldset (optional)

Specifies the field set that overrides the reserved `matrixchilditems` field set. The `matrixchilditems` field set is used by the `matrixchilditems_detail` synthetic field by default to retrieve the item fields of all the matrix child items.

In the following example, the default `matrixchilditems` field set is overridden with a field set named `mini` using the Item Search API parameter `matrixchilditems_fieldset`.

```
/api/items?fieldset=search&matrixchilditems_fieldset=mini
```

n (optional)

Specifies the SuiteCommerce Advanced site ID.

```
/api/items?n=3
```

offset (optional)

Defines a segmented list of items when used in combination with `limit`. You can generate data for paginated lists by specifying the number of items to display on each page. In a site template, you can create the UI element shoppers will use to page through product search results.

```
/api/items?limit=10&offset=10
```

pricelevel (required)

Specifies items with a value for online price. The price level ID can be different depending on how price levels are configured in your NetSuite account. Typically, the default online price in NetSuite is `pricelevel=5`. In Reference Implementation SuiteApps the online price is derived from site settings on the Web Site Setup page.

When a customer with a custom price level is logged in on the web store, the price level assigned to that customer is retrieved instead of the default online price level.

> **Note:** If you have set up multiple price levels for items, ensure that you add all the price level fields as sort fields. For more information, see the help topic Sort Fields for Search Results.
>
> Also, if Alternate Price is not defined for an item but is set to be displayed in a field set, the Item Search API returns the value of Online Price by default. If Online Price is also not defined for the item, then the value of Alternate Price defaults to the Base Price.

For example, the default online price is `pricelevel=5`, but the price level assigned to a certain customer may be `pricelevel=4`. When this customer is browsing in the web store, the customer's assigned price, `pricelevel=4` is sent in the search API call. Products on your site that are displayed to this customer show his or her custom price level.

```
/api/items?pricelevel=4&include=facets
```

> **Note:** The pricelevel parameter is only usable when a user with permissions to see the requested pricelevel is logged in. In all other cases, the pricelevel displayed is in accordance with the default setting in the Web Site Set Up record.

q (optional)

Defines a keyword for the Item Search API request. The keyword must exist in the search fields you specified on the Web Site Setup page in NetSuite. To select search fields, go to Setup > SuiteCommerce Advanced > Set Up Web Site > Search Index > Search Fields.

For example, `q=jeans` searches for that keyword in the fields listed on the Search Fields subtab.

```
/api/items?q=jeans
```

You can also specify multiple keywords in the URL by enclosing them in straight quotes.

For example, `q="denim jeans"` searches for this exact phrase in the fields listed on the Search Fields subtab.

```
/api/items?q="denim jeans"
```

> **Note:** When you use the **q** parameter to query the Item Search API, relevance is the default sort option and will override your settings on the Sort Fields subtab in Web Site Setup.

relateditems_fieldset (optional)

Specifies the field set that overrides the reserved `relateditems` field set. The `relateditems` field set is used by the `relateditems_detail` synthetic field by default to retrieve the item fields of all the related items.

In the following example, the default `relateditems` field set is overridden with a field set named `mini` using the Item Search API parameter `relateditems_fieldset`.

```
/api/items?fieldset=search&relateditems_fieldset=mini
```

region (optional)

Specifies a Subsidiary in a NetSuite OneWorld account. Use `region` with the Subsidiary ID.

```
/api/items?fieldset=fieldset&region=4
```

sort (optional)

Defines the sort order of search results on your web site. You can add sorting to the API request by using the field ID you want to sort by. Returns item record data sorted by the field ID included in the API request. You can find the field ID on the Sort Fields subtab on the Web Site Setup page. An Item Search API request overrides your settings on the Search Fields tab. Supported directions are ascending (asc) and descending (desc).

```
/api/items?sort=storedisplayname:asc
```

When the value of `commercecategory` and either `commercecategoryid` or `commercecategoryurl` are specified, items are sorted by category. This is dependent on the Commerce Category sort field being defined in the Web Site Setup record.

For more information about setting up web site preferences for sorting in NetSuite, see the help topic Search Fields.

url (optional)

Specifies the value in the URL Component field on the item record in an API request. The `url` parameter only returns data for the item record that matches the url component value in the request. Returns the Item ID, if nothing else is specified in the request.

```
/api/items?url=Vok-500-Cellular-Phone
```

You can request associated images by using the `fields` parameter, for example:

```
/api/items?url=Vok-500-Cellular-Phone&fields=itemimages_detail
```

You can also use a field set that contains image information:

```
/api/items?url=Vok-500-Cellular-Phone&fieldset=search
```

Sample Item Search Query and Results

Applies to: SuiteCommerce Web Stores

This section provides sample Item Search API queries and descriptions of response data.

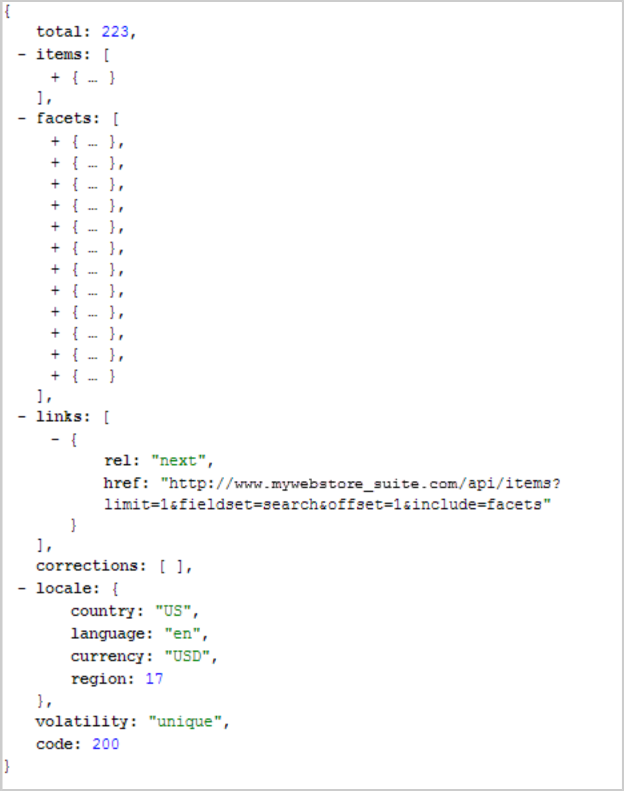
- Sample Item Search API Query
- Item Search API Output Response
- Sample Search Query for Returning Item Quantity
- Sample Search Query for Returning Categories
- Sample Search Query for Returning Items Available for Store Pickup
- Sample Search Query for Improving the Performance of Search Result Pages

Sample Item Search API Query

The query string below limits the results to one item, requests the search field set, and includes facets in the response data:

```
http://www.mywebstore_suite.com/api/items?pricelevel=5&limit=1&fieldset=search&offset=1&include
=facets
```

After the request is processed, the server returns response data and a status code. In the screenshot below, the data returned for items and facets has been collapsed. More information is included in the description of the output response, Item Search API Output Response.



```
{
   total: 223,
 - items: [
     + { … }
   ],
 - facets: [
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … }
   ],
 - links: [
     - {
           rel: "next",
           href: "http://www.mywebstore_suite.com/api/items?
           limit=1&fieldset=search&offset=1&include=facets"
       }
   ],
   corrections: [ ],
 - locale: {
       country: "US",
       language: "en",
       currency: "USD",
       region: 17
   },
   volatility: "unique",
   code: 200
}
```

Item Search API Output Response

Important: The **Mont Blanc** release of SuiteCommerce Advanced requires a patch to ensure that the Item Search API response is cached in the CDN. See the help topic Item Search API Response Data not Cached for details.

The response data includes the following properties:

- **total** – This is the number of items that match the search query, but without taking into account the stock level variance since the last search index update. Consequently, the actual number of items displayed on the page might be less than the total displayed.

  For example, if your search query matches only nine items out of the 1000 items in your catalogue, the value of **total** displayed is nine. If three out of these nine matching items go out of stock since the last search index update, only six items are displayed even though the value of **total** displayed is nine.

- **items** – This is an array of items that includes the fields specified using the `fields` or `fieldset` parameter. Note that the following output for the items property shows fields that are included in the search field set, which was specified in the sample query:

```
items: [
{
         ispurchasable: false,
         custitem_ns_pr_attributes_rating: "",
         showoutofstockmessage: true,
         custitem38: false,
         custitem33: "&nbsp;",
         itemid: "OL5299",
         onspecial: false,
         onlinecustomerprice: 19.99,
         pricelevel5: 19.99,
         outofstockbehavior: "Disallow back orders but display out-of-stock message",
         storedescription2: "Features<BR>-Cool.Q  ZERO provides an immediate and ongoing coolin
g sensation<BR>-Mesh
         sides for ventilation<BR>-Lightweight foam brim folds up and packs down small for shov
ing in a pack<BR>-
         Drawcord attachment under the chin",
         storedisplayname2: "Chiller Wide Brim Hat",
         custitem_onsale: false,
         internalid: 6548
}
         ],
```

- **facets** – This shows the facets associated with items returned in the response data. Use the `include` parameter to show all of facets you defined in on the Web Site Setup page. Note the following sample from the output response shows IDs for two facets, custitem24 and custitem28. Each facet has two facet values.

```
facets: [
      {
         id: "custitem24",
         values: [
               {
                  url: "Men",
                  label: "Men"
               },
               {
                  url: "Women",
                  label: "Women"
               }
            ]
         },
         {
{
         id: "custitem28",
         values: [
               {
                  url: "Spring",
                  label: "Spring"
               },
               {
                  url: "Fall",
                  label: "Fall"
               }
            ]
         },
         {
```

For more information on defining facets in NetSuite, see the help topic Facet Fields. For more information about customizing faceted navigation in SuiteCommerce Advanced, see the help topic Faceted Navigation.

- **links** – This is data for creating hyperlinks that are automatically returned by the API. For example, note the `limit` and `offset` parameters are included in the href that is automatically returned in response data.

```
links:    [
      {
            rel: "prev",
            href: "http://www.mywebstore_suite.com/api/items?pricelevel=5&limit=1&fieldset=sear
ch&offset=-9&include=facets"
      },
   ],
```

- **corrections** – This shows the past and current values for facet URL components in the Item Search API request. If you did not enter any aliases in NetSuite for facets or facet values you used in the past, then the corrections property returns no data.

  For example, in the past you may have used `custitem_category` as a facet. At that time, the facet value was **Hockey Player**. Today, you are using **Equipment Category** as the facet, and the current facet value is **Player**. When you modify the facet URL values, you can add the original values to the Alias list in NetSuite. If you query the Item Search API using the original facet names and values, then corrections are returned in the response data. For more information about entering URL component aliases, see the help topic URL Components for Facets.

  Note that the output response to the following query string includes corrections, because the API request includes the original facet and facet value:

```
http://www.mywebstore_suite.com/api/items?pricelevel=5&fieldset=search&include=facets&custitem_
category=HockeyPlayer
```

  Note that both past and current facet URL components are included in the response data:

```
corrections:
         [
                  {
                        type:"facet",
                        id:"custitem_category",
                        usedAlias:"custitem_category",
                        url:"EquipmentCategory"
                   },
                   {
                         type:"facet-value",
                         facet:"custitem_category",
                         usedAlias:"HockeyPlayer",
                         url:"Player"
                   }
           ],
```

- **locale** – This is an object that represents the locale with respect to internationalization. For more information, see the help topic Localization.

```
locale: {
      country: "US",
      language: "en",
      currency: "USD",
      region: 17
},
```

- **volatility** – This refers to the CDN cache settings for your SuiteCommerce Advanced implementation. Changing this setting in your application can have a significant performance impact. For more information, see the help topics CDN Caching and setCDNCacheable(type).
- **code** – This is the HTTP status code. All responses should have a status code or a message that allows the client to continue processing the results or handle the error if necessary. If the request is successful, a 200 status code indicates that the request was received, understood, and has been processed.

Sample Search Query for Returning Item Quantity

If you use multiple locations and operate a SuiteCommerce Advanced web site, you can access item inventory per location. Create a field set that includes item quantity detail fields, and then use that field set in a query to the Item Search API. The following example, shows a query string that returns data from the Quantity Backordered field on the item record:

```
http://www.mywebstore.com/api/items?pricelevel=5&fieldset=myfieldsetwithqtybackordered
```

The JSON response data includes location (internalid) and quantity back ordered for each item:

```
items: [
{
      qtybackordered_detail:{
         qtybackordered:10
         locations:[
                     {
                        internalid:1,
                        qtybackordered:6
                     },
                     {
                        internalid:2,
                        qtybackordered:4
                     }
                     ]
                                 }
}
]
```

Accessing Item Quantity Per Location with the Item Search API

First make sure the data is entered properly on item records. Next create a field set using fields specifically designed to return item location details in JSON response data. Then you can return inventory data using the Item Search API. Note that you must create the field set using the fields listed as Item Quantity Detail Fields for SuiteCommerce Advanced.

To create a field set that includes item quantity information:

1. Go to Setup > SuiteCommerce Advanced > Setup Web Site.
2. Click the **Field Sets** subtab.
   a. Enter a **Name** for the field set, for example: **Qty Backordered**.
   b. Enter a **Field Set ID**, for example: **qtybackordered**.
   c. Select **Item** for **Record Type**.
   d. (Optional) Enter a description for the field set.
   e. In the Field Set popup window, select from the item quantity fields identified in the table below.
3. Click **Save**.
4. Use the field set in a query to the Item Search API.

Item quantity information is written to the JSON response. For an example, see the Sample Search Query for Returning Item Quantity.

Item Quantity Detail Fields for SuiteCommerce Advanced

The table below shows the item record fields that correspond with fields exposed to field sets.

Field Label on the Locations Subtab of the Item Record	Field Set Field Name	Field ID
Quantity Available	Available (Detail)	quantityavailable_detail
Quantity Backordered	Back Ordered (Detail)	quantitybackordered_detail
Quantity Committed	Quantity Committed (Detail)	quantitycommitted_detail
Quantity On Hand	On Hand (Detail)	quantityonhand_detail

Field Label on the Locations Subtab of the Item Record	Field Set Field Name	Field ID
Quantity On Order	On Order (Detail)	quantityonorder_detail

Sample Search Query for Returning Categories

The following query returns items where the comcat fieldset is configured with a commerce category ID 3. Facet information is also returned in the response. For each item returned, category details are returned including:

- The URL fragments defined for that category.
- The commerce category name used in the facet.

> **Note:** To successfully run this query, the Commerce Category facet field and field set must be configured in the Web Site Set Up record. For more details on working with Categories, see the help topic Commerce Categories.

```
http://www.mywebstore_suite.com/api/items?fieldset=comcat&include=facets&commercecategoryid=3
```

```
{
  "total": 4,
  "items": [
    {
      "internalid": 388,
      "commercecategory": {
        "primarypath": [
          {
            "urls": [
              "/apparel/partywear/shoes"
            ],
            "name": "Shoes",
            "id": 3
          },
          {
            "urls": [
              "/apparel/partywear"
            ],
            "name": "Partywear",
            "id": 2
          },
          {
            "urls": [
              "/apparel"
            ],
            "name": "Apparel",
            "id": 1
          }
        ],
        "categories": [
          {
            "urls": [
              "/apparel/partywear/shoes"
            ],
            "name": "Shoes",
            "id": 3
          }
        ]
      },
      "storedisplayname2": "Shoes - Reebok 9K"
    },
    ...
  ],
  "facets": [
    {
      "id": "commercecategoryname",
      "values": [
        {
          "url": "Shoes"
        }
      ]
    },
    ...
 ],
  "links": [],
  "corrections": [],
  "locale": {
    "country": "US",
    "language": "en",
    "currency": "USD",
    "region": 1
  },
  "volatility": "unique",
  "code": 200,
  "warnings": {}
}
```

Sample Search Query for Returning Items Available for Store Pickup

When you use the `isstorepickupallowed` field with the Item Search API and set it to true, the Item Search API returns the item IDs of all the matching items for which store pickup is enabled. When this field is set to false, the Item Search API returns the item IDs of all the matching items for which store pickup is not enabled.

If you enable the Multi-Location Inventory feature and host a SuiteCommerce Advanced web site, you can access the item quantity per location. If you use the `quantityavailableforstorepickup_detail` field with the Item Search API, you can also access the item quantity available for store pickup per location.

The following example shows a query string that returns data from the Quantity Available and Quantity Available For Store Pickup fields on the item record. Alternatively, you can create a field set that includes Quantity Available and Quantity Available For Store Pickup fields, and then use that field set in a query to the Item Search API.

```
http://www.mywebstore.com/api/items?isstorepickupallowed=true&fields=quantityavailable_detail,q
```

```
uantityavailableforstorepickup_detail
```

> **Note:** To successfully run this query, the Store Pickup Allowed facet field must be configured in the Web Site Setup record. You must also define the Available For Store Pickup (Detail) field in the Web Site Setup record field sets. For more information, see Accessing Item Quantity Available for Store Pickup Per Location.

For each item returned in the JSON response, the data includes the internalid of the item, total item quantity available, item quantity available by location, and item quantity available for store pickup by location:

```
"items":[
      {
            "internalid":388,
            "quantityavailable_detail":{
                  "quantityavailable":90.0,
                  "locations":[
                        {
                              "internalid":2,
                              "quantityavailable":39.0
                        },
                        {
                              "internalid":3,
                              "quantityavailable":51.0
                        }
                  ]
               },
               "quantityavailableforstorepickup_detail":{
                     "locations":[
                           {
                                 "internalid":2,
                                 "qtyavailableforstorepickup":39.0
                           },
                           {
                                 "internalid":3,
                                 "qtyavailableforstorepickup":51.0
                           }
                     ]
               }
      },
      {
            "internalid":387,
            "quantityavailable_detail":{
                  "quantityavailable":150.0,
                  "locations":[
                        {
                              "internalid":2,
                              "quantityavailable":150.0
                        }
                  ]
            },
            "quantityavailableforstorepickup_detail":{
                  "locations":[
                        {
                              "internalid":2,
                    "qtyavailableforstorepickup":150.0
                }
            ]
        }
    },
    {
        "internalid":107,
        "quantityavailable_detail":{
            "quantityavailable":4.0,
            "locations":[
                {
                    "internalid":2,
                    "quantityavailable":3.0
                },
                {
                    "internalid":3,
                    "quantityavailable":1.0
                }
            ]
        },
        "quantityavailableforstorepickup_detail":{
            "locations":[
                {
                    "internalid":2,
                    "qtyavailableforstorepickup":3.0
                }
            ]
        }
    }
],
```

In the above JSON response, we can observe the following:

- The first item with internal ID 388 is available at two locations, and both locations allow store pickup for the quantity available at the respective locations.
- The second item with internal ID 387 is only available at one location, and that location allows store pickup for the quantity available.
- The third item with internal ID 107 is available at two locations, but only one location allows store pickup for the quantity available at that location.

Accessing Item Quantity Available for Store Pickup Per Location

To access the quantity available for Store Pickup per location using the Item Search API, you need to create a field set that includes fields listed under Item Fields Related to Store Pickup.

To create a field set that includes information about item quantity available for store pickup:

1. Go to Setup > SuiteCommerce Advanced > Set Up Web Site.
2. Click the **Field Sets** subtab.
    1. Enter a **Name** for the field set. For example, enter **Store Pickup**.
    2. Enter a **Field Set ID** for the field set. For example, enter **storepickup_details**.

   3. Select **Item** for **Record Type**.
   4. (Optional) Enter a description for the field set.
   5. In the Field Set popup window, select the store pickup related field names identified in the table below.
   6. Click **Add**.
3. Click **Save**.
4. Use the field set in a query to the Item Search API. For example,

```
http://www.mywebstore.com/api/items?fieldset=storepickup_details
```

The store pickup related information is returned in the JSON response.

Item Fields Related to Store Pickup

The table below shows the item record fields that correspond with fields exposed to field sets.

Field Label on the Locations Subtab of the Item Record	Field Set Field Name	Field ID
Allow Store Pickup	Store Pickup Allowed	isstorepickupallowed
Quantity Available For Store Pickup	Available For Store Pickup (Detail)	quantityavailableforstorepickup_detail

Sample Search Query for Improving the Performance of Search Result Pages

The `matrixchilditems_detail` field retrieves matrix child item fields from the `matrixchilditems` reserved field set. The `matrixchilditems` field set has the following fields configured as part of Import Field Sets:

`onlinecustomerprice_detail, internalid, quantityavailable, outofstockbehavior, outofstockmessage, stockdescription, isinstock, isbackorderable, ispurchasable, showoutofstockmessage, itemid, minimumquantity, itemtype, isfulfillable, isstorepickupallowed, quantityavailableforstorepickup_detail`

In the following example, you can observe that for each matrix child item about 15 item fields are retrieved. Consequently, the performance of pages that retrieve lot of matrix child items degrades.

```
http://www.mywebstore_suite.com/api/items?fieldset=details
```

```
{
      "total": 10,
      "items": [
         {
               "isinstock": true,
               "metataghtml": "",
               "custitem_ns_pr_item_attributes": "&nbsp;",
               "matrixchilditems_detail": [
                  {
                        "isinstock": true,
                        "itemid": "H2201100-110",
                        "ispurchasable": true,
                        "stockdescription": "",
                        "isbackorderable": true,
                        "onlinecustomerprice_detail": {
                              "onlinecustomerprice_formatted": "$15.99",
                              "onlinecustomerprice": 15.99
                        },
                        "internalid": 7541,
                        "showoutofstockmessage": false,
                        "outofstockbehavior": "- Default -",
                        "itemtype": "InvtPart",
                        "quantityavailable": 100,
                        "outofstockmessage": "",
                        "quantityavailableforstorepickup_detail": {
                              "locations": []
                        },
                        "isstorepickupallowed": false
                  },
                  {
                        "isinstock": false,
                        "itemid": "H2201100-274",
                        "ispurchasable": true,
                        "stockdescription": "",
                        "isbackorderable": true,
                        "onlinecustomerprice_detail": {
                              "onlinecustomerprice_formatted": "$15.99",
                              "onlinecustomerprice": 15.99
                        },
                        "internalid": 7542,
                        "showoutofstockmessage": false,
                        "outofstockbehavior": "- Default -",
                        "itemtype": "InvtPart",
                        "quantityavailable": 0,
                        "outofstockmessage": "",
                        "quantityavailableforstorepickup_detail": {
                              "locations": []
                        },
                        "isstorepickupallowed": false
                  }
         ],
.
.
.
.
```

To improve the performance of search result pages, you can now overwrite the default field set with any other field set using the Item Search API parameter `matrixchilditems_fieldset`. Consider that we have created a new field set `mini`, which has only four item fields. In the following example, fewer matrix child item fields are returned as we have overwritten the default field set with a field set `mini` (using the Item Search API parameter `matrixchilditems_fieldset`):

```
http://www.mywebstore_suite.com/api/items?fieldset=details&matrixchilditems_fieldset=mini
```

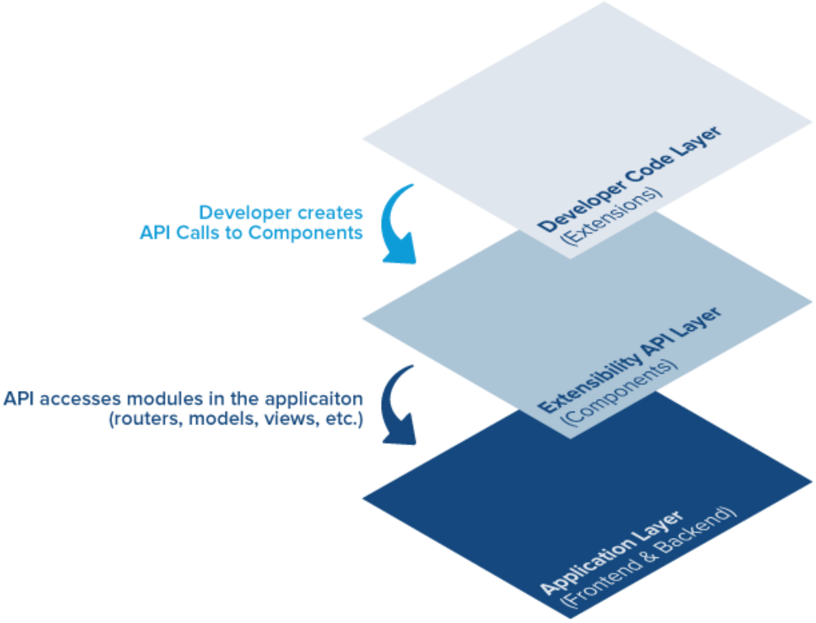
```
 {
      "total": 10,
      "items": [
         {
               "isinstock": true,
               "metataghtml": "",
               "custitem_ns_pr_item_attributes": "&nbsp;",
               "matrixchilditems_detail": [
                  {
                        "itemid": "H2201100-110",
                        "onlinecustomerprice_detail": {
                              "onlinecustomerprice_formatted": "$15.99",
                              "onlinecustomerprice": 15.99
                        },
                        "internalid": 7541,
                        "quantityavailable": 100,
                  },
                  {
                        "itemid": "H2201100-274",
                        "onlinecustomerprice_detail": {
                              "onlinecustomerprice_formatted": "$15.99",
                              "onlinecustomerprice": 15.99
                        },
                        "internalid": 7542,
                        "quantityavailable": 0,
                  }
         ],
.
.
.
.
```

If your search results page includes Related or Correlated items, you can further improve the performance by using the Item Search API parameters `relateditems_fieldset` and `correlateditems_fieldset` to overwrite the default field sets of the `Related Items (Detail)` and `Correlated Items (Detail)` fields respectively.

Overview

ⓘ **Applies to:** SuiteCommerce Web Stores | Aconcagua

SuiteCommerce includes a conceptual layer on top of existing SuiteCommerce architecture. Conceptually, this layer provides an environment for extension developers to customize SuiteCommerce Standard or SuiteCommerce Advanced without the need to extend or customize the underlying application code.



Developers create extensions in their local environment using the extension developer tools. This environment/workspace is represented by the Developer Code Layer in the image above. Each extension contains any number of JavaScript, SuiteScript, or configuration files to perform some purpose. Within this code, developers include API calls to components in the Extensibility API. This is represented by the Extensibility API Layer.

The API then accesses frontend and backend modules (routers, models, views, etc.) in the Application Layer. In this way, extensions are separated programmatically from the application and interact with the API only, not with the application directly. This makes customization more seamless and intuitive for extension developers and ensures that future updates to the application do not compromise developer customizations.

See the following topics for more information about the Extensibility API:

- Extensibility Component Classes – This topic provides an overview of the Component Classes available to you when creating extensions.
- Extensibility API Tutorials – This topic explains how to use the components plus their methods and events when building your extensions. This includes everything from accessing components to building CCTs, Views, etc.

Extensibility Component Classes

Applies to: SuiteCommerce Web Stores | Aconcagua

A component is piece of functionality in a SuiteCommerce application that connects with specific aspects of the application to accomplish a task. The Extensibility API is organized around these components. Each component includes an associated class and various exposed methods for obtaining or modifying data, views, or template context.

The following components classes are currently available using the Extensibility API:

- ProductDetailsPageComponent
- CartComponent
- ProductListPageComponent
- CheckoutComponent
- EnvironmentComponent
- CMSComponent
- LayoutComponent

For more information on how to use these components when building extensions, see the help topic Extensibility API Tutorials.

ProductDetailsPageComponent

ID: `PDP`

This component interacts entirely with the Product Details page. This accesses objects inside the Product Details View only. This is undefined if invoked from any other part of the application.

CartComponent

ID: `Cart`

This component interacts entirely with the LiveOrder, allowing you to add, update, and remove lines from the order, apply promotions to the order, select a shipping method to the order, associate the shipping and billing address to the order, add payment methods, etc. This component can access objects available in all three ssp applications: shopping, myaccount and checkout.

This component is available in the Frontend and the Backend.

ProductListPageComponent

ID: `PLP`

This component interacts entirely with the Search Page, allows navigating to the next or previous page, set the current page, change the rendering, search for a product, get the last search term,etc. This component accesses objects inside the Facets Browse View in the shopping application only. This is undefined if invoked from any other part of the application.

CheckoutComponent

ID: `Checkout`

This component interacts entirely with the Checkout page and the checkout steps flow. This component lets you modify the steps visually, such as adding, removing, and reordering steps, etc. This component can access objects available in the checkout application only.

EnvironmentComponent

ID: `Environment`

This component is general to the entire SuiteCommerce Application and provides the `utils` method and access to Configuration values. This component does not provide interaction with any particular view. This component can access any object available in all three ssp applications: shopping, myaccount and checkout.

CMSComponent

ID: `CMS`

This component is created only for the purpose of registering new Custom Content Types. If you want to add views inside of your CCT you can interact with all the rest of the components available. This component can access objects available in all three ssp applications: shopping, myaccount, and checkout.

LayoutComponent

ID: `Layout`

This component gives you an option for modifying an aspect of SuiteCommerce , but do not have a specific component available. For example, no components currently exist that let you add or modify views on the Home Page, the Header, Footer, or MyAccount. Using `Layout`, you can access all the view methods for each view (addChildViews, addChildView, removeChildView, etc.). As more components become available in future SuiteCommerce updates, you can simply replace the use of `Layout` for the appropriate (new) component, keeping the same methods.

Commerce API

ⓘ **Applies to:** SuiteCommerce Web Stores

The Commerce API exposes a set of shopping objects and functions, that you can use to:

- Develop web store customizations
- Create SSP application scripts, along with the full range of existing server-side SuiteScript. For details about the SuiteScript API, see the help topic SuiteScript Functions.

See the following topics for more information about the Commerce API:

- Shopping Objects – The Commerce API provides several shopping objects whose associated methods enable you to control all aspects of web store behavior. These topics describe all the shopping objects and their associated methods.
- JSON Object Fields – These topics describe the structure of all JSON objects that are available for use with the Commerce API shopping methods.
- SSP Application Governance – The governance model for SSP applications limits the number of usage units that can be executed by an SSP application script. This topic lists the number of usage units consumed by Commerce API shopping methods so that you can use them to create SSP applications that meet the requirements of the governance model.

Shopping Objects

Applies to: SuiteCommerce Web Stores

The Commerce API includes only one object that is globally accessible in the SuiteScript context, the webContainer object.

The following table lists the objects that are accessible after you have gotten a webContainer. The table also lists the calls you can use to get a reference to each object. You need a reference to a shopping object to call its methods.

Object	Description	Call to get Object	For available methods, see:
shoppingSession	Tracks session context.	nlapiGetWebContainer().getShoppingSession()	ShoppingSession Methods
customer	Holds data for the logged in customer.	nlapiGetWebContainer().getShoppingSessi	Customer Methods
order	Holds shopping cart data and methods for placing the order.	nlapiGetWebContainer().getShoppingSession().getOrder()	Order Methods
pageGenerator	Used to generate web pages.	nlapiGetWebContainer().getPageGenerato	PageGenerator Methods
standardTagLibrary	Implements web site tags.	nlapiGetWebContainer().getStandardTagLibrary()	StandardTagLibrary Methods

Note the following:

- These customer and order objects are different from NetSuite customer and order records, because they hold information within the given shopping session only.
- To access a customer or an order, in addition to getting a webContainer, you also need to get a shoppingSession.
- You can only use Commerce API shopping methods against Commerce API objects. You cannot use shopping methods against NetSuite records.
- Shopping object methods support passing of primitive strings and JSON objects. See Input Parameters/Return Values for Shopping Object Methods.
- The Commerce API includes JSON objects to support shopping object methods. For details about supported fields for JSON objects, see JSON Object Fields.
- For information about governance for shopping object methods, see SSP Application Governance.

Input Parameters/Return Values for Shopping Object Methods

Set Functions

Either a primitive string or a JSON object can be passed in to **set** methods, depending on their implementation. A primitive string would be used primarily to hold an object key, while a JSON object would provide a flat representation of an object's name/value pairs.

Get Functions

An optional array can be passed in as a filter to **get** methods. If field names are passed in to a **get** method, only the name/value pairs for the passed field names are returned. If no field names are passed, all name/value pairs for the expected object are returned. If no value is available for a field, null is returned for that field.

ShoppingSession Methods

A shoppingSession object tracks context.

The following methods are available:

- constructDomainBridgingUrl(link)
- doChangePassword(Params, newPassword)
- getAbsoluteUrl(domaintype,path) (Deprecated)
- getAbsoluteUrl2(domaintype,path)
- getCampaignID()
- getCorrelatedItems(item)
- getCountries()
- getCountryTaxPreferences()
- getCustomer()
- getDeviceFingerPrintingHtml()
- getEffectiveShoppingDomain()
- getInformationItemFieldValues(infoItems)
- getItemFieldValues(items)
- getPartnerID()
- getPasswordHint(customerEmail) (Deprecated)
- getPaymentMethod(paymentMethodId, fields)
- getPaymentMethods(fields)
- getRedirectURL()
- getRelatedItems(item)
- getRelatedCorrelatedItems()
- getShipToCountries()
- getShopperCurrency()
- getShopperLanguageLocale()
- getShopperPreferences()
- getShopperSubsidiary()
- getSiteCategoryContents(siteCategory, recursive) (Deprecated)
- getSiteCategoryFieldValues(siteCategories)
- getSiteCategoryTree(siteCategory)
- getSiteSettings(fields)
- getSSPApplication()

- getStates(countries)
- isChangePasswordRequest()
- isLoggedIn()
- isRecognized()
- logIn(customer)
- logout()
- proceedToCheckout(checkoutSettings)
- registerCustomer(customer)
- registerGuest(guest)
- sendPasswordRetrievalEmail(customeremail) (Deprecated)
- sendPasswordRetrievalEmail2(customeremail)
- removeItemOptions(itemlineid)
- setShopperCurrency(currencyid)
- setShopperLanguageLocale(localeid)
- setShopperPreferences(prefs)
- setShopperSubsidiary(subsidiary)
- verifyEmailChange(key)

constructDomainBridgingUrl(link)

Converts cross-domain link to domain bridging URL containing encrypted parameters. This method provides a secure way to transfer shopper data between different domains.

Note: Do not change or extract data from the domain bridging URL as the format of the URL may change. Only use the URL in the browser to retrieve and apply the shopper session data from the original domain.

Parameters

- `link` [required] {String} – Absolute or relative URL of link targeting a different shopping or checkout domain from the one the shopper is currently on

Returns

- `link` {String} – Domain bridging URL; the original URL is returned if bridging is either not possible or not necessary
- `null` – Returned if the input parameter is invalid

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

doChangePassword(Params, newPassword)

Changes the password for the customer identified by email. Returns true if the password is changed correctly.

Throws the following errors:

- ERR_WS_INVALID_LINK – if parameters contain a token that is expired or incomplete. Note that it is not possible to determine if the link is expired or incomplete. The same error is thrown in either case.
- ERR_WS_INVALID_PASSWORD – if the password is empty or exceeds the maximum allowed length of 255 characters.
- ERR_WS_WEAK_PASSWORD – if the password at least 6 characters.

Parameters

- `Params` – pass parameters from original request
- `newPassword` – {String} value for new password to be set

Returns

Boolean

Supported Domains

Checkout

Login Required?

no

Back to ShoppingSession Methods | **Back to** Shopping Objects

getAbsoluteUrl(domaintype,path)

⚠ **Important:** In certain scenarios, the `getAbsoluteUrl(domaintype,path)` method returns a relative URL instead of an absolute URL. For example, the `getAbsoluteUrl(domaintype,path)` method returns a relative URL when the target domain is same as the current domain.

Therefore, the `getAbsoluteUrl(domaintype,path)` method has been deprecated as of NetSuite Release 2017.1. Use the `getAbsoluteUrl2(domaintype,path)` method instead.

Gets absolute URL for given path and domain type.

Parameters

- `domaintype` [optional] {String} - value should be either “shopping” or “checkout”
- `path` [required] {String}

Returns

String

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getAbsoluteUrl2(domaintype,path)

Gets absolute URL for given path and domain type.

> ⓘ **Note:** This method should be used instead of the deprecated `getAbsoluteUrl(domaintype,path)` method.

Parameters

- `domaintype` [optional] {String} - value should be either “shopping” or “checkout”
- `path` [required] {String}

Returns

String

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getCampaignID()

Gets the campaign ID of the leadsource parameter from the current session.

Parameters

No parameters to set.

Returns

`campaignid` {String}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getCorrelatedItems(item)

Gets array of correlated items for given item.

> **Note:** Item details will be returned only if the correlated item is available in the store from which the API was called from.

Parameters

- `item` [required]{object with values for fields}
  - `internalid` [required]
  - `itemtype` [required]
  - `itemfields` [optional] {array of item field names to query}

> **Note:** For better performance limit the number of queried fields by specifying only fields you need in the itemfields array.

Returns

Array of objects of type item

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getCountries()

Gets all countries

Parameters

No parameters to set.

Returns

Array of objects with fields:

- `name` {string}
- `code` {string}
- `iszziprequired` {boolean}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getCustomer()

Gets the Customer details from the current session.

Parameters

No parameters to set.

Returns

Returns the customer shopping object. For more information, see Customer Methods.

> **Note:** The customer object holds information within the given shopping session only and is different from the NetSuite customer record.

Supported Domains

Checkout, Shopping

Login Required?

Yes

Back to ShoppingSession Methods | **Back to** Shopping Objects

getCountryTaxPreferences()

Returns a JSON object with tax preferences related a shopper's country. The values returned are based on the default shopper subsidiary derived from the site or the country selected by the shopper.

Parameters

No parameters to set.

Returns

- `vatregistration` {boolean} — When set to T, the VAT registration field can be displayed during checkout and the `vatregistration` can be set on the customer object.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getDeviceFingerPrintingHtml()

Sets device finger printing for CyberSource for the device a shopper is logged in from when accessing a web store. This can be used with the Cybersource payment gateway for added site security. The API call must be made in the last step of checkout. For more details on device fingerprinting in SuiteCommerce Advanced sites, see the help topic Device Fingerprinting.

Parameters

No parameters to set.

Returns

Returns an HTML string that needs to be rendered on the last page of a checkout that includes the Submit Order button.

Supported Domains

Checkout

Login Required?

Yes

Back to ShoppingSession Methods | **Back to** Shopping Objects

getEffectiveShoppingDomain()

Returns name of shopping domain associated with current session.

Parameters

No parameters to set.

Returns

`domainname` {String} – The shopping domain related to the current session. For example: `shopping.corp.netsuite.com`.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getInformationItemFieldValues(infoItems)

Gets attribute values for given information items.

Parameters

- `infoItems` [required] {Array}

Each object in array has values for field:

- `internalid` [required]

Returns

Array of objects of type item.

> **Note:** Depending on item fields configured, the array of objects returned might differ.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getItemFieldValues(items)

Gets attribute values for given items.

Parameters

- `items` [required] {Array} Each object in array has values for fields:
  - `internalid` [required]
  - `itemtype` [required]

Returns

Array of objects of type item.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getPartnerID()

Gets the Partner ID of the leadsource parameter from the current session.

Parameters

No parameters to set.

Returns

`partnerid` {String}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getPasswordHint(customerEmail)

Retrieves password hint for customer with given email address.

> ⚠ **Important:** Password hints are no longer necessary. The password retrieval email message includes a URL with a password reset code. Therefore, the getPasswordHint() method has been deprecated as of NetSuite Release 2017.2. As of 2017.2, the method will return an empty string, but generate no errors.

Parameters

- `customerEmail` [required] {String}

Returns

An empty string

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getPaymentMethod(paymentMethodId, fields)

Gets payment method for given ID.

Parameters

- `paymentMethodId` [required] {String}
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned

Returns

Object of type paymentmethod.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getPaymentMethods(fields)

Gets payment methods accepted by the web store.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned

Returns

Array of objects of type paymentmethod.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getRedirectURL()

Gets the URL which is the result of a redirect given the input. Returns null if there is no redirect. Wildcards are also supported (*).

Parameters

- `url` [required] {String}

Returns

String

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getRelatedItems(item)

Gets array of related items for given item.

Parameters

- `item` [required]{object with values for fields}
  - `internalid` [required]
  - `itemtype` [required]
  - `itemfields` [optional] {array of item field names to query}

> **Note:** For better performance limit the number of queried fields by specifying only fields you need in the itemfields array.

Returns

Array of objects of type item.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getRelatedCorrelatedItems()

Gets a list of related and/or correlated items for a given item id.

Parameters

- `type`: [required]{string} Available options are RELATED, CORRELATED, or BOTH.
- `fieldsetid` : set of fields returned.
- `items`: list of item IDs. When not specified, items from the cart are returned.
- `limit` : Maximum count of returned items. When not specified, the default 10 is used. -1 sets no limit.

> **Important:** Use caution when setting the limit to -1 as this can have an impact on performance.

Example

```
function service(request, response)
{
  var session = nlapiGetWebContainer().getShoppingSession();
  items = [’101’];
  response.writeLine("Related Items: " + JSON.stringify(session.getRelatedCorrelatedItem’RELATE
D’, ’relateditems’, items, 10)));
  response.writeLine("Correlated Items: " + JSON.stringify(session.getRelatedCorrelatedItems(’C
ORRELATED’, ’relateditems’, items, 10)));
  response.writeLine("Related & Correlated Items: " + JSON.stringify(session.getRelatedCorrelat
edItems(’BOTH’, ’relateditems’, items, 10)));
}
```

Returns

Array of correlated or related items.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getShipToCountries()

Gets countries to which web store can ship.

Parameters

No parameters to set.

Returns

Array of objects with fields:

- `name{string}`
- `code{string}`
- `iszip required` {boolean}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getShopperCurrency()

Gets currency for current shopping session.

Parameters

No parameters to set.

Returns

Object with values for fields:

- `internalid` {String}
- `name` {String}

- `symbol` {String}
- `precision` {Number}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getShopperLanguageLocale()

Gets language and locale for current shopping session.

Parameters

No parameters to set.

Returns

`localeid` {String}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getShopperPreferences()

Gets language, subsidiary, and currency preferences for the current shopping session.

Parameters

No parameters to set.

Returns

Object with values for fields:

- `language` {String}
- `subsidiary{String}`
- `currency{String}`

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getShopperSubsidiary()

Gets subsidiary for current shopping session.

Parameters

No parameters to set.

Returns

`subsidiaryid` {String}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getSiteCategoryContents(siteCategory, recursive)

⚠ **Important:** The `getSiteCategoryContents()` API has been deprecated as of NetSuite Release 2014.2. Use the `getSiteCategoryTree()` method instead. See getSiteCategoryTree(siteCategory).

Gets contents of given site category.

If not provided, `siteCategory` will be defaulted to the root category, and recursive will be defaulted to false. When recursive is set to `false`, the API returns the category contents that include subcategories and items; when recursive is set to `true`, the API returns the complete tree of subcategories but not items.

ⓘ **Note:** To get the complete category tree from the root, use `getSiteCategoryContents(true)`.

Parameters

- `siteCategory` [optional] {Array} Each object in array has values for field: `internalid` [required]
- `recursive` [optional]

Returns

Array of objects of type sitecategory or item.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getSiteCategoryFieldValues(siteCategories)

Gets attribute values of given site categories.

Parameters

- `siteCategories` [required] {Array} Each object in array has values for field:
  - `internalid` [required]

Returns

Array of objects of type sitecategory.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getSiteCategoryTree(siteCategory)

When the `internalid` attribute is present in the `siteCategory` object, then the method returns the category sub-tree from this category. When the `internalid` attribute is not specified, then the whole category tree of the specified site is returned.

When calling this API, you must specify which fields to return. When no fields are specified, then no fields are returned except the `internalids` of categories. Specify fields using the `itemfields` attribute on the `siteCategory` object.

Examples

- The following example returns the entire category tree for the current site. Each category record contains the fields `internalid`, `pagetitle2`, and `urlcomponent`.

  ```
  getSiteCategoryTree({ itemfields : ["pagetitle2","urlcomponent"] });
  ```

- The following example returns the category sub-tree of category 65. Each category record contains only the `internalid` field.

  ```
  getSiteCategoryTree({ id : 65 });
  ```

Note: The examples above return only the category tree, not the category contents.

Parameters

- `siteCategory` [optional] {Array} Each object in array has values for field: `internalid` [required]

Returns

Array of objects of type sitecategory

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getSiteSettings(fields)

Gets settings of current web store.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned

Returns

Object with fields listed for sitesettings JSON object.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getSSPApplication()

Gets URL root of SSP application powering the current touchpoint.

Parameters

No parameters to set.

Returns

`urlroot` field value for sitesettings JSON object.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

getStates(countries)

Gets states and/or provinces.

Parameters

- `countries` [optional] Array of strings for country codes

Returns

Returns an array of state/province names and state/province codes in the following format:

{"states":[{"name":"Alabama","code":"AL"},{"name":"Alaska","code":"AK"},...], "countrycode":"US"}

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

isChangePasswordRequest()

Returns true if all page parameters from the current request are valid for the password reset.

Throws the error "ERR_WS_INVALID_LINK" if all page parameters from the current request contain required parameters for password reset but the link has expired or is incomplete.

Parameters

No parameters to set.

Returns

Boolean

Supported Domains

Checkout

Login Required?

no

Back to ShoppingSession Methods | **Back to** Shopping Objects

isLoggedIn()

> **Important:** This method is no longer supported or recommended. For new implementations, use the isLoggedIn2() method. This method works correctly in the Checkout domain but may return false when used in the Shopping domain even when the user is logged in. The method is currently being use in pre_Denali versions of SuiteCommerce Advanced. If you encounter session issues in those bundles, contact support to migrate to using the `isLoggedIn2()` method.

Checks whether customer has logged in. This method can be used with the `isRecognized()` method. See Using isRecognized() with isLoggedIn2().

Parameters

No parameters to set.

Returns

Boolean

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

isLoggedIn2()

Checks whether customer has logged in. This method can be used with the `isRecognized()` method. See Using isRecognized() with isLoggedIn2().

> **Note:** This method should be used instead of the deprecated isLoggedIn() method.

Parameters

No parameters to set.

Returns

Boolean

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

isRecognized()

Determines whether a user is recognized. This method can be used with the `isLoggedIn2()` method. See Using isRecognized() with isLoggedIn2().

Parameters

No parameters to set.

Returns

Boolean

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

logIn(customer)

Logs in a customer.

Parameters

- `customer` [required]{object with values for fields}
  - `email` [required]
  - `password` [required]

Returns

Object with fields:

- `customerid` {String} - internal id of customer record for logged in customer
- `redirecturl{String}` - URL to which user is redirected after logging in

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

logout()

Logs out a customer.

> **Note:** If `logout()` is attempted from a shopping context, it does not succeed and an error is returned.

Parameters

No parameters to set.

Returns

Object with field:

- redirecturl

The field `redirecturl` has the complete home URL to redirect to after logout. If not using the `redirecturl` field returned from logout(), the custom redirecturl should have the `logoff=T` parameter.

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

proceedToCheckout(checkoutSettings)

Used for integration with a third party checkout provider such as PayPal Express. This API can only be called from a secure scheme (https).

Parameters

- `checkoutSettings` Object with values for fields:
  - `type;` either “paypalexpress” or “google”
  - `continueurl`
  - `cancelurl`

Returns

No value returned.

Supported Domains

Checkout

> **Note:** Must have at least one order.

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

registerCustomer(customer)

Registers the customer and logs them in.

Parameters

- `customer` [required]{object with values for fields}
  - `firstname` [required] {String}
  - `lastname` [required] {String}
  - `companyname` [optional] {String}
  - `email` [required] {String}
  - `password` [required] {String}
  - `password2` [required] {String}
  - `passwordhint` [optional] {String}
  - `emailsubscribe` [optional] {String}
  - `leadsource` [optional] {String}

Returns

Object with values for fields:

- `customerid` {String} - internal id of record for registered customer
- `redirecturl` {String} - URL to which user is redirected after registering

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

registerGuest(guest)

Registers a guest.

> **Note:** Throws exception if register guest with password.

Parameters

- `guest` [optional]{object with values for fields}
  - `firstname` [optional] {String}
  - `lastname` [required] {String}

Returns

Object with values for fields:

- `customerid` {String} - internal id of record for registered customer
- `redirecturl` {String} - URL to which user is redirected after registering

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

sendPasswordRetrievalEmail(customeremail)

Sends password retrieval email message to the given email address.

> ⚠ **Important:** The password reset link generated by the sendPasswordRetrievalEmail() includes the customer's email address. Therefore the method has been deprecated as of NetSuite Release 2017.2. Use the sendPasswordRetrievalEmail2(customeremail) method instead.

Parameters

- `customerEmail` [required] {String}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

sendPasswordRetrievalEmail2(customeremail)

Sends password retrieval email message to the given email address.

> ⓘ **Note:** This method does not include the customer's email address in the reset password link. Use this method instead of the deprecated sendPasswordRetrievalEmail(customeremail).

Parameters

- `customerEmail` [required] {String}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

setShopperCurrency(currencyid)

Sets currency for current shopping session

> ⓘ **Note:** Multiple Currencies feature must be enabled, specified currency must be supported in sitesettings, and logged in user must have permission to change currency.

Parameters

- `currencyid` [required] {string}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

setShopperLanguageLocale(localeid)

Sets language and locale for current shopping session

> ⓘ **Note:** Multi-Language feature must be enabled, and specified language must be supported in sitesettings.

Parameters

- `localeid` [required]{String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

setShopperPreferences(prefs)

Sets language, subsidiary and currency for current shopping session.

Parameters

- `prefs` [required]{object with values for fields}
  - `language` [optional] {String}
  - `subsidiary` [optional] {String}
  - `currency` [optional] {String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

setShopperSubsidiary(subsidiary)

Sets subsidiary for current shopping session (for NetSuite OneWorld users).

⚠ **Important:** The subsidiary set in a session is **not** currently supported out-of-the-box for One World in the SuiteCommerce Advanced SuiteApp. For the SuiteCommerce Advanced SuiteApp, further customization is necessary. Also, the subsidiary for an existing recognized customer can **not** be changed using this API.

Parameters

- `subsidiary` [required]{String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to ShoppingSession Methods | **Back to** Shopping Objects

verifyEmailChange(key)

Takes an email change key, verifies it, and then changes the email. If the request can't be processed, an exception is thrown.

Parameters

- `key` [required]{String} — value of the email change key.

Returns

Boolean

Returns `true` if the key value is valid and the email has been changed. Returns `false` in all other cases (for example, if the key is invalid or expired).

Supported Domains

Checkout

Login Required?

Yes

Back to ShoppingSession Methods | **Back to** Shopping Objects

Using isRecognized() with isLoggedIn2()

There are four possible cases for using the `isLoggedIn2()` method and `isRecognized()` method to correctly determine the logged in status of a user:

- new (anonymous) user: `isRecognized() == false` and `isLoggedIn2() == false`
- registered and logged in user: `isRecognized() == true` and `isLoggedIn2() == true`
- user who clicked logout: `isRecognized() == false` and `isLoggedIn2() == false`
- user who had logged in, then closed his browser and later returns to the shop: `isRecognized() == true` and `isLoggedIn2() == false`

> **Note:** This last scenario is the only case where the `isRecognized()` status differs from the `isLoggedIn2()` status. In this case the user didn’t click on the logout link and is therefore still recognized but `isLoggedIn2()` returns false.

Customer Methods

A customer object holds data for the logged in customer. This object is different from a NetSuite customer record, because it holds information for the given shopping session only.

The following Customer methods are available:

- addAddress(address)
- addCreditCard(creditcard)
- emailCustomer(subject, body)

- getAddress(addressid, fields)
- getAddressBook(fields)
- getCampaignSubscriptions(fields)
- getCampaignSubscriptions(subscriptionId,fields)
- getCreditCard(creditcardid, fields)
- getCreditCards(fields)
- getCustomFields()
- getCustomFieldValues()
- getFieldValues(fields)
- isGuest()
- removeAddress(addressid)
- removeCreditCard(creditcardid)
- setLoginCredentials(customer)
- updateAddress(address)
- updateCampaignSubscriptions(subscriptions)
- updateCreditCard(creditcard)
- updateProfile(customer)

addAddress(address)

Adds an address for current customer.

Parameters

- `address` [required] {Object with values for fields}
  - `addressee` [required]
  - `addr1` [required]
  - `addr2` [optional]
  - `addr3` [optional]
  - `city` [required]
  - `state` [required]
  - `country` [required]
  - `zip` [required]
  - `phone` [optional]
  - `isresidential` [optional]
  - `defaultshipping` [optional]
  - `defaultbilling` [optional]

Returns

String - key of added address.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

addCreditCard(creditcard)

Adds a credit card for current customer.

Parameters

- `creditcard` [required] {Object with values for fields}
    - `ccnumber` [required]
    - `ccname` [required]
    - `authcode` [optional]
    - `customercode` [optional]
    - `paymentmethod` [required] (numerical ID for credit card type in NetSuite)
    - `expmonth` [required]
    - `expyear` [required]
    - `validfrommon` [required for UK]
    - `validfromyear` [required for UK]
    - `debtcardissueno` [required for UK]
    - `ccdefault` [optional]

Returns

String - key of added credit card.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

emailCustomer(subject, body)

Sends email to current customer with given subject and body.

Parameters

- `subject` [required] {String}
- `body` [required] {String}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getAddress(addressid, fields)

Gets address with given ID for current customer.

Parameters

- `addressid` [required] {string}
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type address.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getAddressBook(fields)

Gets all addresses for current customer

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Array of objects of type address.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getCampaignSubscriptions(fields)

Gets subscription information for the current customer

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Array of objects of type subscriptions.

Supported Domains

Checkout, Shopping

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getCampaignSubscriptions(subscriptionId,fields)

Gets data for a single campaign subscription with passed id.

Parameters

- `subscriptionId` [required] {String}
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned

Returns

Array of objects of type subscriptions

Supported Domains

Checkout, Shopping

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getCreditCard(creditcardid, fields)

Gets credit card with given ID for current customer.

Parameters

- `creditcard` [required] {Object with values for fields}
    - `internalid` [required]

  - `ccnumber` [required]
  - `ccname` [required]
  - `authcode` [optional]
  - `customercode` [optional]
  - `paymentmethod` [required]
  - `expmonth` [required]
  - `expyear` [required]
  - `validfrommon` [required for UK]
  - `validfromyear` [required for UK]
  - `debtcardissueno` [required for UK]
  - `ccdefault` [optional]

Returns

Object of type creditcard.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getCreditCards(fields)

Gets all credit cards for current customer.

Parameters

- `customer` [required] {Object with values for fields}
  - `internalid` [required]
  - `email` [optional]
  - `emailsubscribe` [optional]
  - `firstname` [optional]
  - `lastname` [optional]
  - `middlename` [optional]
  - `phoneinfo` [optional]
  - `customfields` [optional] {Object with {customfield id : customfield value} }

Returns

Array of objects of type creditcard.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getCustomFields()

Gets custom fields on customer record for current customer.

Parameters

No parameters to set.

Returns

Object with custom field names.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Customer Methods | **Back to** Shopping Objects

getCustomFieldValues()

Gets custom field values on customer record for current customer.

Parameters

No parameters to set.

Returns

Object with custom field values.

Supported Domains

Checkout, Shopping

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

getFieldValues(fields)

Gets standard field values on customer record for current customer.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type customer.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

isGuest()

Checks whether the shopper is logged in as a guest. Returns False when the shopper is not logged in or is logged in as a customer.

Parameters

No parameters to set.

Returns

Boolean

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Customer Methods | **Back to** Shopping Objects

removeAddress(addressid)

Removes address with given ID from current customer's list of addresses.

Parameters

- `addressid` [required] {string}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

removeCreditCard(creditcardid)

Removes given credit card from current customer.

Parameters

- `creditcardid` [required] {string}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

setLoginCredentials(customer)

Sets login credentials for current guest customer.

Parameters

- `customer` [required] {Object with values for fields}
  - `internalid` [required]
  - `email` [required]
  - `password` [required]
  - `customfields` [Optional] {Object with {customfield id : customfield value} }

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

updateAddress(address)

Updates a specified address.

Parameters

- `address` [required] {Object with values for fields}
  - `internalid` [required]
  - `addressee` [required]
  - `addr1` [required]
  - `addr2` [optional]
  - `addr3` [optional]
  - `city` [required]
  - `state` [required]
  - `country` [required]
  - `zip` [required]
  - `phone` [optional]
  - `isresidential` [optional]
  - `defaultshipping` [optional]
  - `defaultbilling` [optional]

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

updateCampaignSubscriptions(subscriptions)

Updates a specified campaign subscription.

Parameters

- `subscriptions` [required] {Object with values for fields}
  - `internalid` [required]
  - `subscribed` [required]

    Customer can only opt-in or opt-out of an email campaign category by setting subscribed to ‘T' or ‘F'.

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

updateCreditCard(creditcard)

Updates given credit card for current customer.

Note: If given `ccnumber` does not match an existing record, `internalid` is ignored and a new credit card record is added for customer.

Parameters

- `creditcard` [required] {Object with values for fields}
  - `internalid` [required]
  - `ccnumber` [required]
  - `ccname` [required]
  - `authcode` [optional]
  - `customercode` [optional]
  - `paymentmethod` [required]
  - `expmonth` [required]
  - `expyear` [required]
  - `validfrommon` [required for UK]
  - `validfromyear` [required for UK]
  - `debtcardissueno` [required for UK]
  - `ccdefault` [optional]

Returns

String - key of added or updated credit card.

Supported Domains

Checkout

Login Required?

Yes

Back to Customer Methods | **Back to** Shopping Objects

updateProfile(customer)

Updates profile of current customer.

Parameters

- `customer` [required] {Object with values for fields}
  - `internalid` [required]
  - `email` [optional]
  - `emailsubscribe` [optional]
  - `firstname` [optional]
  - `lastname` [optional]
  - `middlename` [optional]
  - `phoneinfo` [optional]
  - `customfields` [optional]{Object with {customfield id : customfield value} }

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Order Methods

An order object holds shopping cart data and methods for placing the order. This object is different from a NetSuite order record, because it holds information for the given shopping session only.

The following Order methods are available:

- addFreeGiftItem(item)
- addItem(item)
- addItems(items)
- applyGiftCertificate(giftcertificate)
- applyPromotionCode(promocode)
- estimateShippingCost(address)
- setItemExcludedFromShipping(cartItemId,excluded)
- getAppliedGiftCertificate(giftcertificate, fields)
- getAppliedGiftCertificates(fields)
- getAppliedPromotionCode(promocode, fields)
- getAppliedPromotionCodes(fields)
- getAvailableShippingMethod(shipmethodid,fields)
- getAvailableShippingMethods(orderLineIds, shipAddressId)
- getAvailableShippingMethods(fields)
- getBillingAddress()

- getCustomFields()
- getCustomFieldValues()
- getEligibleFreeGiftItems()
- getFieldValues(fields)
- getItem(orderitemid, fields)
- getItemOption(orderitemid,fields)
- getItems(bConsolidated), getItems(fields, bConsolidated)
- getItems(fields)
- getItemShippingFieldValues(orderitemid)
- getOrderSummary(fields)
- getPayment(fields)
- getPurchaseNumber()
- getShippingAddress()
- getShippingMethod(fields)
- mergeItems(orderLineIds)
- rejectFreeGiftItem(item)
- removeAllItems()
- removeBillingAddress()
- removeItem(orderitemid)
- removePayment()
- removePurchaseNumber()
- removePromotionCode(promocode)
- removeAllPromotionCodes()
- removeShippingAddress()
- removeShippingMethod()
- setBillingAddress(addressid)
- setCustomFieldValues(customfields)
- setEnableItemLineShipping()
- setItemOptions(itemlineid,optionMap)
- setItemShippingAddress(orderItemId, shipAddressId)
- setItemShippingAddress(orderItemIds, shipAddressId)
- setItemShippingMethod(orderItemId, shipMethodId)
- setItemShippingMethod(orderItemIds, shipMethodId)
- setPayment(payment)
- setPurchaseNumber(purchaseNumber)
- setShippingAddress(addressid)
- setShippingMethod(shipmethod)
- setTermsAndConditions(readandagree)
- splititem(orderLine)
- submit(ordersettings)
- updateItemFulfillmentPreferences(item)

- updateItemsFulfillmentPreferences(items)
- updateItemQuantity(item)
- updateItemQuantities(items)

addFreeGiftItem(item)

Adds the eligible item to the cart and marks it as a free item.

Parameters

- `item` [required] {Object with values for fields}
  - `item_id` [required] {Number}
  - `promotion_id` [required] {Number}
  - `quantity` [required] {Number or String}
  - `options` [required for item with item options]

Note: *quantity* can be a negative number.

Returns

String - the `orderitemid` of added item.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

addItem(item)

Adds an item to order.

> **Note:** If your selected Out Of Stock Behavior option is **Disallow back orders but display out-of-stock message** or **Remove out-of-stock items from store**, you might still be able to add an out of stock item to your cart using the `addItem(item)` order method. However, when you try to place the order with such out of stock item in your shopping cart, an error message is displayed and you will not be able to place the order successfully.

Parameters

- `item` [required] {Object with values for fields}
  - `internalid` [required]
  - `quantity` [required] {Number or String}
  - `options` [required for item with item options]
  - `fulfillmentPreferences` [optional] {fulfillmentPreferences object}

Note: *quantity* can be a negative number.

Returns

String - the `orderitemid` of added item.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

addItems(items)

Adds multiple items to cart/order within a single request.

> **Note:** If your selected Out Of Stock Behavior option is **Disallow back orders but display out-of-stock message** or **Remove out-of-stock items from store**, you might still be able to add out of stock items to your cart using the `addItems(items)` order method. However, when you try to place the order with such out of stock items in your shopping cart, an error message is displayed and you will not be able to place the order successfully.

Parameters

- `items` [required]{Array of objects with values for fields}
  - `internalid` [required]
  - `quantity` [required] {Number or String}
  - `options` [required for item with item options]
  - `fulfillmentPreferences` [optional] {fulfillmentPreferences object}

Returns

String - the `orderitemid` of added items.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

applyGiftCertificate(giftcertificate)

Applies given gift certificate to the order.

Parameters

- `giftcertificate` [required] {String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

applyPromotionCode(promocode)

Applies given promotion code to the order.

Parameters

- `promocode` [required] {String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

estimateShippingCost(address)

Gets estimated shipping cost for given address.

> **Note:** Estimated shipping cost depends on order's shipping method, enabling of shipping estimator on site administration page, and whether order requires shipping.

Parameters

- `address` [optional] {Object with fields}
  - `zip` [required]
  - `country` [required]
  - `state` [optional]

Returns

Number

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

setItemExcludedFromShipping(cartItemId,excluded)

Identifies when an item on an order is excluded from shipping.

Parameters

- `cartItemId` [required] {string}
- `excluded` [required] {boolean}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getAppliedGiftCertificate(giftcertificate, fields)

Gets specified gift certificate.

Parameters

- `giftCertificate` [required] Certificate(giftcertificate, fields)
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type giftcertificate.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getAppliedGiftCertificates(fields)

Gets all applied gift certificates.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Array of objects of type giftcertificate

Supported Domains

Checkout

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getAppliedPromotionCode(promocode, fields)

Gets specified promotion code.

Parameters

- `promoCode` [required] {String}
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type promocode

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getAppliedPromotionCodes(fields)

Gets all applied promotion codes.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Array of objects of type promocode.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getAvailableShippingMethods(orderLineIds, shipAddressId)

Returns available shipping methods and addresses for the given item in the order.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `orderLineIds` [required] Array of `orderitemids` to merge
- `shipAddressId` [required] String

Returns

Array of objects of type shipmethod.

Supported Domains

Checkout

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getAvailableShippingMethod(shipmethodid,fields)

Gets specified shipping method.

Parameters

- `shipmethodid` [required] {String}
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type shipmethod.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getAvailableShippingMethods(fields)

Gets all available shipping methods.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Array of objects of type shipmethod.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getBillingAddress()

Gets billing address for order.

Parameters

No parameters to set.

Returns

Object of type address.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

getCustomFields()

Gets custom fields on order record.

Parameters

No parameters to set.

Returns

Object with custom field names.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getCustomFieldValues()

Gets custom field values for current order.

Parameters

No parameters to set.

Returns

Object with custom field values.

Supported Domains

Checkout, Shopping

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

getEligibleFreeGiftItems()

Gets a list of all Free Item promotions that are currently applied. It also gets the Item IDs of all the free items.

Parameters

No parameters to set.

Returns

Array of objects with the following fields:

- `promotion_id`
- `eligible_items`
- `eligible_quantity`
- `rejected_quantity`
- `added_quantity`

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getFieldValues(fields)

Gets standard field values for the current order.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type order.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getItem(orderitemid, fields)

Gets specified order item.

Parameters

- `orderitemid` [required] {String}
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type orderitem.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getItemOption(orderitemid,fields)

Gets item option for specified order item.

Parameters

- `orderitemid` [required] {String}
- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

The options field from the object of type orderitem.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getItems(bConsolidated), getItems(fields, bConsolidated)

Used to configure multiple shipping routes on web orders. Depending on the value for `bConsolidated`, order items are returned in either a split view or in a consolidated view.

For example, displaying an un-consolidated view of all the items in the cart allows shoppers to select the shipping address for each item. You can then use the consolidated view to display items to be shipped, grouped by shipping address.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `bConsolidated` Boolean
- `fields` Array of field names to be included in returned JSON object; if omitted, all supported fields are returned

Returns

Object of type orderitem.

Supported Domains

Checkout

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getItems(fields)

Gets order items.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Array of objects of type orderitem.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getItemShippingFieldValues(orderitemid)

Gets shipping field values for specified order item. This API is only available when the Multiple Shipping Routes feature is enabled.

Parameters

- `orderitemid` [required] {String}

Returns

Object with fields:

- `shipaddress`
- `shipmethod`

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

getOrderSummary(fields)

Gets the order summary.

> **Important:** For the most optimized results, use the `itemcount` argument with this method. When this method is called with no arguments, all information about the order is returned which results in poor performance.

Example

```
 var itemcount = nlapiGetWebContainer().getShoppingSession().getOrder().getOrderSummary([’itemc
ount’]);
```

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type ordersummary.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

getPayment(fields)

Gets payment details for order.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type payment.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

getPurchaseNumber()

Gets the Purchase order number associated with the order.

Parameters

No parameters to set.

Returns

String

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

getShippingAddress()

Gets shipping address for order.

Parameters

No parameters to set.

Returns

Object of type address

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

getShippingMethod(fields)

Gets shipping method for order.

Parameters

- `fields` [optional] Array of field names to be included in returned JSON object; if omitted, all supported fields are returned}

Returns

Object of type shipmethod.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

mergeItems(orderLineIds)

Merges the given line items into a single order line item. The given order lines can only differ in quantity and shipping address and method.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `orderLineIds`[required]: Array of `orderitemids` to merge

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Sample Code

```
nlapiGetWebContainer().getShoppingSession().getOrder().mergeItems(orderLineIds);
```

Back to Order Methods | **Back to** Shopping Objects

rejectFreeGiftItem(item)

Allows shoppers to reject the free item provided the item is not already in the cart.

Parameters

- `item` [required] {Object with values for fields}
  - `promotion_id` [required] {Number}
  - `quantity` [required] {Number or String}

Note: *quantity* can be a negative number.

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

removeAllGiftCertificates()

Removes all gift certificates from an order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Sample Code

```
nlapiGetWebContainer().getShoppingSession().getOrder().removeAllGiftCertificates()
```

Back to Order Methods | **Back to** Shopping Objects

removeAllItems()

Removes all items from order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

removeBillingAddress()

Removes the current billing address from the order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

yes

Back to Order Methods | **Back to** Shopping Objects

removeItem(orderitemid)

Removes specified item from order.

Parameters

- `orderitemid` [required] {String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

removeItemOptions(itemlineid)

Removes specified item options from a row in the cart.

Parameters

- `itemlineid` [required]

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

removePayment()

Removes the current payment method for the order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

removePurchaseNumber()

Removes the associated Purchase order number from order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

removePromotionCode(promocode)

Removes specified promotion code from order.

Parameters

- `promocode` [required] {String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

removeAllPromotionCodes()

Removes all promotion codes from the order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

removeShippingAddress()

Removes the current shipping address from the order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

removeShippingMethod()

Removes the current shipping method for the order.

Parameters

No parameters to set.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setBillingAddress(addressid)

Sets the billing address for the order.

Parameters

- `addressid` [required] {String}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setCustomFieldValues(customfields)

Sets custom field values for order.

Parameters

- `customfields` [required] {Object with name/value pair for each field}

> ⚠ **Important:** Custom fields of the type **Document** are NOT supported for non-Employee external centers.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setEnableItemLineShipping()

Toggles the item line shipping flag for the current order. This flag **must** be set prior to using other Multiple Ship To APIs.

> ⚠ **Important:** The Multiple Shipping Routes feature is required.

An exception is thrown when:

- The Multiple Shipping Routes feature is not enabled
- The site is not using the Standard Sales Order Form for scripting

Parameters

- `enable` [required]: Boolean

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Sample Code

```
nlapiGetWebContainer().getShoppingSession().getOrder().setEnableItemLineShipping(true);
```

Back to Order Methods | **Back to** Shopping Objects

setItemOptions(itemlineid,optionMap)

Updates the item options for a particular cart row.

Parameters

- `itemlineid` [required]
- `optionMap` [required] {Array of Item Option ID value pairs}

> **Note:** `itemlineid` is a combination of item id and set. This is included because of the possibility for multiple quantities of the same item but with different item options.

Returns

itemLineId of the new line with the new item options.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

setItemShippingAddress(orderItemId, shipAddressId)

Sets the shipping address for the given order line item.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `orderItemId`[required]: String
- `shipAddressId`[required]: String

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Sample Code

```
nlapiGetWebContainer().getShoppingSession().getOrder().setItemShippingAddress(orderItemId, ship
AddressId);
```

Back to Order Methods | **Back to** Shopping Objects

setItemShippingAddress(orderItemIds, shipAddressId)

Sets the shipping address for an array of given order line items.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `orderItemIds`[required]: Array of `orderItemId` fields.
- `shipAddressId`[required]: String

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setItemShippingMethod(orderItemId, shipMethodId)

Sets the shipping method for the given order line item.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `orderItemId`[required]: String.
- `shipMethodId`[required]: String

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Sample Code

```
nlapiGetWebContainer().getShoppingSession().getOrder().setItemShippingMethod(orderItemId, shipM
ethodId);
```

Back to Order Methods | **Back to** Shopping Objects

setItemShippingMethod(orderItemIds, shipMethodId)

Sets the shipping method for an array of given order line items.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `orderItemIds`[required]: Array of `orderItemId` fields.
- `shipMethodId`[required]: String

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setPayment(payment)

Sets payment for order; for credit card, creates a new record if `internalid` not provided.

Parameters

- `payment` [required] {Object with values for fields}
  - `internalid` [optional]

  - *ccname [optional]
  - *ccnumber [optional]
  - *expmonth [optional]
  - *expyear [optional]
  - paymentmethod[required]
  - paymentterms[optional]
- creditcard [required] (this is an object within the payment object.)

Note: * Fields are required only for credit card payment methods.

For information on creating non-credit card payment methods, see the help topic Alternative Credit Card Payment Methods.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setPurchaseNumber(purchaseNumber)

Sets the purchase number for the order.

Parameters

- purchaseNumber [required] {String}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setShippingAddress(addressid)

Sets the shipping address for the order.

Parameters

- `addressid` [required] {String}

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setShippingMethod(shipmethod)

Sets shipping method for order.

Parameters

- `shipmethod` [required] {Object with values for fields}
  - `shipmethod` [required]
  - `shipcarrier` [required]

> **Note:** Valid values that can be set for shipcarrier are `ups` or `noups`.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

setTermsAndConditions(readandagree)

If required by site, sets whether user has read and agreed to terms and conditions.

Parameters

- `readandagree` [required] {Boolean}, or {String}, where value must be “T” or “F”

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

splititem(orderLine)

Splits the given line item into multiple order line items. You can then set a shipping address for each order line.

> **Note:** The `setEnableItemLineShipping()` flag must be set before using this method. See setEnableItemLineShipping().

Parameters

- `orderLine` [required] {Object with values for fields}
  - `orderitemid` [required]
  - `quantities` [optional] An array of quantities to split into. If not provided, all of the line items from the split are given a quantity of 1.

Returns

No value returned.

Supported Domains

Checkout

Login Required?

Yes

Sample Code

```
nlapiGetWebContainer().getShoppingSession().getOrder().splitItem(orderitem);
```

Back to Order Methods | **Back to** Shopping Objects

submit(ordersettings)

Places the shopping order.

Parameters

- checkoutsettings

Returns

Object with fields:

- `status`
- `internalid`
- `confirmationnumber`

Supported Domains

Checkout

Login Required?

Yes

Back to Order Methods | **Back to** Shopping Objects

updateItemFulfillmentPreferences(item)

Updates specified item's fulfillment choices for the order.

Parameters

- `item` [required] {Object with values for fields}
  - `orderitemid` [required]
  - `fulfillmentPreferences` [required] {fulfillmentPreferences object}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

updateItemsFulfillmentPreferences(items)

Updates specified items' fulfillment choices for the order.

Parameters

- `items` [required] {Array of objects with values for fields}
  - `orderitemid` [required]
  - `fulfillmentPreferences` [required] {fulfillmentPreferences object}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

updateItemQuantity(item)

Updates specified item's quantity for order.

Parameters

- `item` [required]{Object with values for fields}
  - `orderitemid` [required]
  - `quantity` [required] {Number or String}

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Back to Order Methods | **Back to** Shopping Objects

updateItemQuantities(items)

Updates specified items' quantities for order.

Parameters

- `items` [required]{Array of objects with values for fields}
  - `orderitemid` [required]
  - `quantity` [optional] {Number or String} Defaults to 1 if omitted.

Returns

No value returned.

Supported Domains

Checkout, Shopping

Login Required?

No

Understanding Multiple Ship To

APIs for Multiple Ship To provide the ability to code a checkout experience that allows shoppers to ship items in their cart to multiple shipping addresses.

Before working with the Multiple Ship To APIs, ensure the **Multiple Shipping Routes** feature is enabled in your account. For detailed information on the Multiple Shipping Routes feature, see the help topic Multiple Shipping Routes.

The set of APIs related to Multiple Ship To include:

- updateItemFulfillmentPreferences(item)
- updateItemsFulfillmentPreferences(items)
- setEnableItemLineShipping()
- splititem(orderLine)
- mergeItems(orderLineIds)
- setItemShippingAddress(orderItemId, shipAddressId)
- setItemShippingAddress(orderItemIds, shipAddressId)
- setItemShippingMethod(orderItemId, shipMethodId)
- setItemShippingMethod(orderItemIds, shipMethodId)
- getItems(bConsolidated), getItems(fields, bConsolidated)
- getAvailableShippingMethods(orderLineIds, shipAddressId)
- addItem(item)
- addItems(items)

Known Limitations

- **Promotions**: Multiple Ship To does not work with promotions with shipping impact. The promo code is automatically removed from the order.
- **One shipping method per shipping address**: Multiple Ship To is limited to defining one shipping method per shipping address. Shoppers can not create multiple shipments to the same address. For example, a shopper can not have one set of items shipping to their Home address using a Standard shipping method and another set of items shipping to their Home address using a One Day shipping method.
- **Real time rates and performance**: There is a known performance impact on the Proceed to Checkout flow when a ship method with real time rates is used. This can be exacerbated for orders with Multiple Ship To. Ensure that you do perform performance testing before going live when using real time rates.
- **Login Required**: In order to use the above mentioned Order methods for Multiple Ship To, you must be logged in.

PageGenerator Methods

Methods on the pageGenerator object are used to generate a web store page.

The following PageGenerator methods are available:

- addBreadCrumb(label, url)
- addHeadHtml(html)

- addStylesheetHtml(html)
- addTab(id, label, url)
- addToPageInitScript(scriptLines)
- removeBreadCrumbs()
- removeTab(id)
- removeTabs()
- setDocTypeHtml(html)
- setMetaTagHtml(html)
- setPageTitle(title)
- setSelectedTab(id)
- showTabNavigation(show)

addBreadCrumb(label, url)

Adds a bread crumb to the end of the existing bread crumb trail.

Parameters

- `label` [required] {string} - text displayed for the bread crumb
- `url` [required] {string} - URL to which the bread crumb links

Back to PageGenerator Methods | **Back to** Shopping Objects

addHeadHtml(html)

Adds the specified HTML to the top of the page's <head> section, before the stylesheet HTML.

Parameters

- `html` [required] {string} - HTML content

Back to PageGenerator Methods | **Back to** Shopping Objects

addStylesheetHtml(html)

Adds the specified HTML to the top of the page's <head> section, after the stylesheet HTML

Parameters

- `html` [required] {string} - HTML content

Back to PageGenerator Methods | **Back to** Shopping Objects

addTab(id, label, url)

Adds a tab to the right of current tabs on the page.

Parameters

- `id` [required] {string} - unique string id for the tab

- `label` [required] {string} - label displayed within the tab
- `url` [required] {string} - URL to which the tab links

Back to PageGenerator Methods | **Back to** Shopping Objects

addToPageInitScript(scriptLines)

Adds lines of JavaScript to the page's initialization script.

Parameters

- `scriptLines` [required] {string}

Back to PageGenerator Methods | **Back to** Shopping Objects

removeBreadCrumbs()

Removes all bread crumbs from the page.

Parameters

No parameters to set.

Back to PageGenerator Methods | **Back to** Shopping Objects

removeTab(id)

Removes a specified tab from the page.

Parameters

- `id` [required] {string} - unique string id for the tab

Back to PageGenerator Methods | **Back to** Shopping Objects

removeTabs()

Removes all tabs from the page.

Parameters

No parameters to set.

Back to PageGenerator Methods | **Back to** Shopping Objects

setDocTypeHtml(html)

Establishes HTML for the page `<doctype>` tag.

Parameters

- `html` [required] {string} - HTML content

Back to PageGenerator Methods | **Back to** Shopping Objects

setMetaTagHtml(html)

Establishes HTML for the page's `<meta>` tags.

Parameters

- `html` [required] {string} - HTML content

Back to PageGenerator Methods | **Back to** Shopping Objects

setPageTitle(title)

Establishes a title for the page.

Parameters

- `title` [required] {string} - title text

Back to PageGenerator Methods | **Back to** Shopping Objects

setSelectedTab(id)

Causes the specified tab to be displayed as the currently active tab.

Parameters

- `id` [required] {string} - unique string id for the tab

Back to PageGenerator Methods | **Back to** Shopping Objects

showTabNavigation(show)

Indicates whether tab navigation should be shown or hidden.

Parameters

- `show` [required] {boolean} - true to display navigation, false to hide it

StandardTagLibrary Methods

Methods on a `standardTagLibrary` object implement NetSuite web store tags.

Each of the following methods returns the same result as the corresponding NetSuite page tag. All results are strings.

- getCartLinkHtml()
- getCartUrl()
- getCheckoutLinkHtml()
- getCheckoutUrl()
- getCompanyId()
- getCrumbTrail()

- getCurrencySelectHtml()
- getCustomerCenterLinkHtml()
- getCustomerCenterUrl()
- getGlobalSearchFormHtml()
- getGlobalSearchHtml()
- getJSessionId()
- getLanguageSelectHtml()
- getLoginLinkHtml()
- getLoginUrl()
- getLogoutLinkHtml()
- getLogoutUrl()
- getPageFullHead()
- getPageHead()
- getPageLinks()
- getPageTabs()
- getPageTop()
- getRegionSelectHtml()
- getRegistrationLinkHtml()
- getRegistrationUrl()
- getReloginLinkHtml()
- getReloginUrl()
- getSideBarWidth()
- getSiteLogoHtml()
- getSiteNavigationHtml()
- getUserInfo()
- getUserInfo2()
- getWelcomeImageHtml()

JSON Object Fields

ⓘ **Applies to:** SuiteCommerce Web Stores

Shopping methods are implemented with JSON objects. **Set** methods can have JSON objects passed in as input parameter values and **get** methods return strings extracted from JSON objects. Note that parsing of returned JSON values is done when values are set to rows (for example, when cart items are built).

Tables for the following shopping method objects list the fields exposed for each. These fields are for use in web store carts and orders. For convenience, some fields are exposed here that can also be accessed through the existing SuiteScript API.

- order
- address
- analyticssettings
- checkoutsettings
- creditcard
- customer
- discounts_impact
- discounts
- fulfillmentPreferences
- giftcertificate
- item
- message
- orderitem
- ordersettings
- ordersummary
- payment
- paymentmethod
- promocode
- registrationsettings
- shipmethod
- sitecategory
- sitesettings
- status
- threedsecure
- touchpoints

order

This object contains other objects, and includes the complete set of data for a web store shopping order.

Object Name	Object Type	Description
billaddress	JSON object (type of address)	Billing address for order

Object Name	Object Type	Description
customer	JSON object	Current shopper information
customfields	JSON object	List of custom fields name/value pairs set for order
giftcertificate s	Array	Gift certificates applied to order
items	Array of orderitem s	Items added to order
payment	JSON object	Payment information for order
promocode s	Array	Promotion codes applied to order
purchaseNumber	JSON object	Purchase order number for the order
shipaddress	JSON object (type of address)	Shipping address for order
shipmethod	JSON object	Shipping method for order
status	JSON object	Status of order
summary	JSON object (type of ordersummary)	Summary information for order (subtotal, total, tax, etc.)

address

This object contains address information for a web store shopping order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
addressee	string	Person to which order is sent	yes	yes (reqd)
addr1	string	First line of address	yes	yes (reqd)
addr2	string	Second line of address	yes	yes
addr3	string	Third line of address	yes	yes
attention	string	Text for ATTN: line of address	yes	yes
city	string	City	yes	yes (reqd)
country	string	Country	yes	yes (reqd)
defaultbilling	string	Indicates whether address is default billing address Value should be T or F.	yes	yes
defaultshipping	string	Indicates whether address is default shipping address Value should be T or F.	yes	yes
internalid	string	Internal ID	yes	yes
isresidential	string	Indicates whether address is residential Value should be T or F.	yes	yes
phone	string	Phone number **Note:** Address forms on Reference Checkout 2.04 and earlier **require** the phone number field.	yes	yes
state	string	State for address	yes	yes (reqd)

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
zip	string	Zip code	yes	yes (reqd)

analyticssettings

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
clickattributes	string		yes	no
confpagetrackinghtml	string		yes	no
submitattributes	string		yes	no

checkoutsettings

This object contains information used for checkout.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
cancelurl	string	Full URL path for redirect after cancel action during order submission Used for PayPal Express support	no	yes
continueurl	string	Full URL path for redirect after order submission Used for PayPal Express support	no	yes
createorder	boolean	If value is set to T, backend submits current shopping order when Continue link is clicked Default value is F Used for PayPal Express support	no	yes
custchoosespaymethod	string	Value should be T or F.	yes	no
paymentauthorization	threedsecure object with fields	Settings to handle payment authorization Used for 3D Secure support	yes	yes
paymentmandatory	string	Indicates whether payment is mandatory Value should be T or F.	yes	no
paypalexpress	Object with fields	Used for PayPal Express support Includes the following fields: ■ *available* (boolean) (must be set to T to support PayPal Express) ■ *imageurl* (string)	yes	no
requiretermsand conditions	string	Indicates whether terms and conditions text is required Value should be T or F.	yes	no
saveccinfo	string	Indicates whether credit card should be saved by default	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		Value should be T or F.		
shippingaddrfirst	string	Indicates whether shipping address should be displayed first Value should be T or F.	yes	no
showpurchaseorder	string	Indicates whether purchase order should be displayed Value should be T or F.	yes	no
showsavecc	string	Indicates whether the Save Credit Card field should be displayed Value should be T or F.	yes	no
termsandconditions	string	Text of Terms and Conditions field	yes	no
type	string	Used for integration with third party checkout providers Valid values are: paypalexpress, google	no	yes

creditcard

This object contains credit card information.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
ccdefault	string	Indicates whether credit card is the default Value should be T or F.	yes	yes
ccexpiredate	string	Card expiration date	yes	no
ccname	string	Name on card	yes	yes
ccnumber	string	Card number Value returned by Get function is masked.	yes	yes
ccsecuritycode	string	Card security code Only needed if payment method is creditcard.	no	yes
customercode	string	Customer code	yes	yes
debtcardissueno	string	Card issue number for UK **Note**: Not exposed in Customer record.	yes	yes
expmonth	string	Card expiration month	yes	yes
expyear	string	Card expiration year	yes	yes
internalid	string	Internal ID	yes	yes
paymentmethod	Object of type paymentmethod	Payment method type **Note**: For credit card payments, this is the numerical code of the credit card type.	yes	yes
savecard	string	Indicates whether card should be saved	no	yes

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		Value should be T or F.		
validfrom	string	Card valid from date **Note**: Not exposed in Customer record.	yes	no
validfrommon	string	Card valid from month for UK	yes	yes
validfromyear	string	Card valid from month for UK	yes	yes

customer

This object contains information about the logged in customer.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
addressbook	List of JSON address objects	Address information	yes	no
balance	string	Customer's accounts receivable balance due, shown in customer's currency.	yes	no
campaignsubscriptions	List of JSON subscription objects	Information on campaign subscriptions for the given customer.	yes	yes
companyname	string	Customer company name	yes (for Checkout and Shopping domains)	yes (for Checkout domain only)
creditcards	List of JSON creditcard objects	Credit card information	yes	no
creditholdoverride	string	Customer's credit hold status.	yes	no
creditlimit	string	Maximum currency amount the customer is allowed to accrue in outstanding receivables.	yes	no
email	string	Customer email address Required in Set function for customer registration, guest registration, and login.	no	yes
emailsubscribe	string	Email subscription Value should be T or F.	yes	yes
firstname	string	Customer first name	yes	yes
internalid	string	Internal ID	yes	no
isperson	boolean	If false this customer is of customer type. If true this customer is of the type individual.	yes	no
lastname	string	Customer last name	yes	yes
middlename	string	Customer middle name	yes	yes
name	string	Customer name	yes	yes

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		⚠ **Important:** As of Version 2013 Release 1, this field is deprecated and included for backwards compatibility only. The firstname, lastname, and middlename fields are the preferred fields for getting and setting customer names.		
partner	string	Partner	yes	yes
password	string	Password Required in Set function for customer registration and login.	no	yes
password hint	string	Hint	no	yes
paymentterms	JSON object of fields	Payment terms associated with the customer	yes	no
phoneinfo	JSONobject of fields	Returns or sets the phone number information for a customer. For example: {"phone":"65460780", "fax" : "40540", "altphone" : null}	yes	yes

discounts_impact

This object contains information about the overall discount impact for a given item.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
discounts	Array of JSON object of type discounts	Discounts applied to the order.	yes	no
total_amount	number	Total currency amount of the discount applied to the order.	yes	no
total_amount_formatted	string	Discount amount formatted with currency symbol.	yes	no

discounts

This object contains information about the per line discount impact for a given item.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
promotion_couponcode	string	The name of the coupon code. This corresponds to what the user applies during the checkout process. For example, user might apply SUMMER or FREESHIP.	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		Note: If you manually add the discount without using a coupon code, the value of promotion_couponcode is Empty.		
promotion_name	string	Name of the associated promotion. For example, name of the promotion can be 10% Summer Sale. **Note:** If you manually add the discount without using a coupon code, the value of promotion_name is Empty.	yes	no
promotion_id	integer	ID of the associated promotion. **Note:** If you manually add the discount without using a coupon code, the value of promotion_id is Null.	yes	no
discount_item	string	Name of the associated discount item. For example, name of the discount item can be Ten Percent.	yes	no
discount_item_id	integer	ID of the associated discount item.	yes	no
amount	number	Currency amount of the discount applied to the order as a result of applying this coupon code.	yes	no
amount_formatted	string	Discount amount formatted with currency symbol.	yes	no
is_auto_applied	boolean	Indicates whether the promocode was automatically or manually applied for the current order. Value should be true or false.	yes	no

fulfillmentPreferences

This object contains information about the fulfillment choices for a web store shopping order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
fulfillmentChoice	string	Indicates the type of fulfilment, and its value should be **ship** or **pickup**. **Note:** The value **pickup** can be used only when the Store Pickup feature is enabled.	yes (reqd)	yes (reqd)
pickupLocationId	integer	Indicates the internal ID of the pickup location, and is available only if the fulfillment choice is **pickup**.	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
shippingAddressId	integer	Indicates the shipping address ID, and is available only if the fulfillment choice is **ship**. **Note:** This can only be used only when the Multiple Ship To feature is enabled.	yes	no
shippingMethodId	integer	Indicates the shipping method ID, and is available only if the fulfillment choice is **ship**. **Note:** This can only be used only when the Multiple Ship To feature is enabled.	yes	no

giftcertificate

This object contains information about gift certificates in the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
amountapplied	number	Amount applied to order	yes	no
amountremaining	string	Remaining gift certificate amount after applied to order	yes	no
giftcertcode	string	Gift certificate code	yes	yes
originalamount	number	Original amount of gift certificate	yes	no

item

This object contains basic information about an item that can be included in the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
featureddescription	string	Featured description	yes	no
internalid	string	Internal ID for item Unique identifier that should be used when adding an item to the cart	yes	yes (reqd)
isavailable	string	Indicates whether item is available for web store order Value should be T or F.	yes	no
isdonationitem	string	Indicates whether item is donation item Value should be T or F.	yes	no
isdropshipitem	string	Indicates whether item is a dropship item Value should be Yes or No.	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
isspecialorderitem	string	Indicates whether item is a special order item Value should be Yes or No.	yes	no
isstorespecial	string	Indicates whether item is on special in the web store Value should be T or F.	yes	no
istaxable	string	Whether the item is taxable Value should be Yes or No.	yes	no
itemtype	string	Type of the item, such as inventory, kit/package, etc.	yes	no
nopricemessage	string		yes	no
nopricemessage2	string		yes	no
outofstockmessage	string		yes	no
outofstockmessage2	string		yes	no
outofstocknobackorder message	string		yes	no
pagetitle	string		yes	no
pagetitle2	string		yes	no
quantityavailable	string	Number of items available for ordering	yes	no
rate	string	Unit price	yes	no
rate_formatted	string	Unit price formatted with currency symbol	yes	no
storedetaileddescription	string	Store detailed description	yes	no
storedescription	string	Store description	yes	no
storedisplayimage	string		yes	no
storedisplayname	string	Store display name	yes	no
storedisplayname2	string	Second store display name	yes	no
storedisplaythumbnail	string		yes	no
freegiftpromotionid	string	ID of the free gift promotion this item belongs to	yes	no

message

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
text	string (with HTML markup)	Text of message		
type	string	Type of message ▪ Confirmation ▪ Error ▪ Information		

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		- Warning		

orderitem

The orderitem object is a subtype of item.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
amount	string	Currency amount for items	yes	no
amount_formatted	string	Amount formatted with currency symbol	yes	no
internalid	string	Internal ID for item Unique identifier that should be used when adding an item to the cart	yes	yes (reqd)
name	string	Display name	yes	no
options	JSON object	List of name/value pairs	yes	yes
orderitemid	string	Unique ID for an item in an order	yes	yes
promotionamount	string	Amount after any line-level discounts	yes	no
promotionamount_formatted	string	Amount after any line-level discounts formatted with currency symbol	yes	no
promotiondiscount	string	Amount of discount related to promotions	yes	no
promotiondiscount_formatted	string	Amount of discount related to promotions formatted with currency symbol	yes	no
quantity	string	Number of items in order	yes	yes
rateschedule	string	Price schedule for quantity pricing	yes	no
shipaddress	string	ID of shipping address for item	yes	yes
shipcarrier	string	Shipping carrier	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		Note: Valid values that can be set for shipcarrier are `ups` or `noups`.		
shipmethod	JSON object of type shipmethod	ID of shipping method for item Used when the Multiple Shipping Routes feature is enabled	yes	yes
taxtype1	string	Type of the item tax. For example, GST/VAT/WET	yes	no
taxrate1	string	Rate of the tax in percentage	yes	no
taxtype2	string	For Canada only. Currently returns `PST`.	yes	no
taxrate2	string	For Canada only. The rate of PST tax in percentage when applicable	yes	no
tax1amt	string	The total tax amount for the line item.	yes	no
fulfillmentPreferences	JSON object of type fulfillmentPreferences	Fulfillment preference for item	yes	no
discounts_impact	JSON object of type discounts_impact	Per line discount impact for the line item.	yes	no

Working with Taxes

In the United States, per item taxes are not applied to an order. Instead tax is applied to the entire order. The following tax fields are only used when countries require per line tax:

- taxtype1
- taxrate1
- taxtype2
- taxrate2

Since every line can have a different tax type or tax rate you should define the taxes per order line. When a line is not taxed, omit those fields.

Note: You should also check the value of `pricesincludevat` in the `sitesettings` object to verify whether taxes are included in the price. These fields are only required when tax is not included in the price.

ordersettings

This object contains general settings for the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
itemdisplayorder	string	Ordering of items in an order	yes	no
outofstockbehavior	string	Behavior for out of stock items	yes	no
upselldisplay	string	■ Related Items Only ■ Upsell Items ■ Related Items and then Upsell Items ■ Upsell Items and then Related Items	yes	no

ordersummary

This object contains summary information for the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
discountedsubtotal	number	Item subtotal - any applicable discount	yes	no
discountedsubtotal_formatted	string	Item subtotal - any applicable discount formatted with currency symbol	yes	no
discountrate	string	Raw discount	yes	no
discounttotal	number	Total amount of discounts applied	yes	no
discounttotal_formatted	string	Total amount of discounts applied formatted with currency symbol	yes	no
giftcertapplied	number	Total amount of gift certificates applied	yes	no
giftcertapplied_formatted	string	Total amount of gift certificates applied formatted with currency symbol	yes	no
handlingcost	number	Amount of handling cost	yes	no
handlingcost_formatted	string	Amount of handling cost formatted with currency symbol	yes	no
itemcount	number	Returns the quantity of items in the cart. ⚠ **Important:** Use this argument with the `getOrderSummary` method for the most optimized call. Note that the value returned equals the total quantity of items in the cart which may differ from the total lines in the cart. For example, if a shopper has three of the same items in the cart, the call returns 3.	yes	no
shippingcost	number	Amount of shipping cost	yes	no
shippingcost_formatted	string	Amount of shipping cost formatted with currency symbol	yes	no
subtotal	number	Total of quantity * rate per line item	yes	no
subtotal_formatted	string	Total of quantity * rate per line item formatted with currency symbol	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
tax2total	number	Total secondary tax on taxable elements of the order (items + shipping + handling) - effect of any discounts	yes	no
tax2total_formatted	string	Total secondary tax on taxable elements of the order (items + shipping + handling) - effect of any discounts formatted with currency symbol	yes	no
taxondiscount	number	For VAT countries, currency amount of the tax rate * the discount amount	yes	no
taxondiscount_formatted	string	For VAT countries, currency amount of the tax rate * the discount amount formatted with currency symbol	yes	no
taxonhandling	number	Currency amount of the tax rate * taxable handling amount	yes	no
taxonhandling_formatted	string	Currency amount of the tax rate * taxable handling amount formatted with currency symbol	yes	no
taxonshipping	number	Currency amount of the tax rate * taxable shipping amount	yes	no
taxonshipping_formatted	string	Currency amount of the tax rate * taxable shipping amount formatted with currency symbol	yes	no
taxtotal	number	Total tax on taxable elements of the order (items + shipping + handling) - effect of any discounts	yes	no
taxtotal_formatted	string	Total tax on taxable elements of the order (items + shipping + handling) - effect of any discounts formatted with currency symbol	yes	no
totalcombinedtaxes	number	A grand total of all the tax components (taxtotal + tax2total) associated with an order	yes	no
totalcombinedtaxes_formatted	string	A grand total of all the tax components (taxtotal + tax2total) associated with an order formatted with currency symbol	yes	no
total	number	Order subtotal + shipping, handling, and all tax, less the effect of any discounts	yes	no
total_formatted	string	Order subtotal + shipping, handling, and all tax, less the effect of any discounts formatted with currency symbol	yes	no

payment

This object contains payment information for the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
creditcard	JSON creditcard object	Payment information when credit card is selected as the payment method	yes	yes

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		Note: See Formatting Values for Credit Card Payments for an example of how to format this information.		
paymentmethod	string	ID of the payment method selected by customer	yes	yes
paymentterms	string	Payment terms selected by customer Value should be CreditCard or Invoice. **Note**: Set this value to **Invoice** to use an invoice as payment. See Using an Invoice as Payment for an example.	yes	yes
paypal	JSON object	PayPal information if enabled	yes	no
status	JSON status object	Embedded status object used for web store business logic validation	yes	yes

Formatting Values for Credit Card Payments

The following example code illustrates how to format details about credit card payments:

```
function service(request,response)
{

var payment = {
'paymentmethod': '5' ,
creditcard: {
  'ccname': 'Herb Joseph',
  'ccnumber': '4111111111111111', 'expmonth' : '09',
  'expyear': '2013',
  'paymentmethod': '5'
  }
};

var order = nlapiGetWebContainer().getShoppingSession().getOrder();
order.setPayment(payment)

};
```

Using an Invoice as Payment

The following example code illustrates how to use an invoice as a payment method:

```
function service(request,response)
{
                    var returnval = null;

                    try
                    {
                                        var payment = {
                                                            paymentterms : 'Invoice'
                                        };
                                        nlapiGetWebContainer().getShoppingSession().getOrder().
setPayment(payment);
                    }
                    catch (e)
                    {
                                        var nle = nlapiCreateError(e);
                                        returnval = {status : 'error', reasoncode : nle.getCode
(), message : nle.getDetails()};
                    }
                    response.writeLine(JSON.stringify(returnval));
}
```

paymentmethod

This object includes payment method information for the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
internalid	string	Internal ID	yes	no
iscreditcard	string	Indicates whether payment method is a credit card Value should be T or F.	yes	no
ispaypal	string	Indicates whether payment method is PayPal Value should be T or F.	yes	no
name	string	Name of the payment method	yes	no
paypalemailaddress	string	Primary paypal email address	yes	no
merchantid	string	Internal ID of the merchant account linked with the payment method	yes	no
isexternal	string	Indicates whether payment method is handled outside of NetSuite Value should be T or F.	yes	no
imagesrc	string	Returns the URL of the image file used to represent the payment method.	yes	no

Note: You cannot set an invoice as a payment method. Instead, set invoice as the paymentterms value for the payment object. See Using an Invoice as Payment for an example.

promocode

This object contains promotion code information for the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
discountamount	string	Currency amount of the discount applied to the order.	yes	no
errormsg	string	Returns the error message when a promo code is not valid.	yes	no
internalid	string	Internal ID	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
isvalid	string	Indicates whether promocode is valid for the current order. Value should be T or F.	yes	no
promocode	string	The value of the Coupon Code ID. This corresponds to what the user sets during the checkout process.	yes	yes
promocodeid	string	The value of the Promocode ID. This corresponds to the promocode displayed in the NetSuite UI. **Note:** The value of this field is not changed when a coupon code is modified.	yes	yes
purchase_discount_amount	string	Currency amount of the discount applied to the order.	yes	no
purchase_discount_amount_forn	string	Discount amount formatted with currency symbol.	yes	no
shipping_discount_amount	string	Currency amount of the discount applied to the shipping cost.	yes	no
shipping_discount_amount_form	string	Discount shipped amount formatted with currency symbol.	yes	no
promotion_name	string	Name of the promotion associated with the applied coupon.	yes	no
discount_name	string	Name of the discount associated with the applied promotion.	yes	no
discount_type	string	Indicates the type of discount, and its value should be PERCENTAGE or FLAT.	yes	no
discount_rate	string	Rate of the discount in percentage if the type of discount is PERCENTAGE and currency amount of the discount if the type of discount is FLAT.	yes	no
is_auto_applied	boolean	Indicates whether the promocode was automatically or manually applied for the current order. Value should be true or false.	yes	no
applicability_status	string	Indicates whether the promocode is actually applied to the current order or not. Value should be one of the following: ■ APPLIED — Indicates that the promocode is applied to the current order and you get a discount if the promocode is valid. It also indicates that the type of promotion is SuitePromotion. ■ NOT_APPLIED — Indicates that the promocode does not give you a discount even though it is added to the current order. It also indicates that the type of promotion is SuitePromotion. ■ NOT_AVAILABLE — Indicates that the applicability status information is not	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
		available. You get this value in case of Standard promotions.		
applicability_reason	string	Specifies the reason for not applying the promocode. Value should be one of the following: - DISCARDED_BEST_OFFER — Indicates that the promocode is not applied as there is another promocode or a combination of promocodes that is giving you a better discount. - CRITERIA_NOT_MET — Indicates that the promocode is not applied as it does not meet the criteria. For example, if your order does not meet the minimum order amount requirement. - NOT_AVAILABLE — Indicates that the promocode is applied or the promotion is a Standard promotion.	yes	no
promotion_type	string	Indicates the type of promotion.	yes	no

registrationsettings

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
companyfieldmandatory	string	Indicates whether Company field is mandatory Value should be T or F.	yes	no
registrationallowed	string	Indicates whether registration is allowed Value should be T or F.	yes	no
registrationanonymous	string	Indicates whether registration is anonymous Value should be T or F.	yes	no
registrationmandatory	string	Indicates whether registration is required Value should be T or F.	yes	no
registrationoptional	string	Indicates whether registration is optional Value should be T or F.	yes	no
showcompanyfield	string	Indicates whether to show Company field on registration form	yes	no

shipmethod

This object contains shipping method information for the order.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
name	string	Name of the shipping method	yes	no
rate	string	Rate of the shipping method	yes	no
shipcarrier	string	Shipping carrier ID **Note:** Valid values that can be set for shipcarrier are `ups` or `noups`.	yes	yes
shipmethod	string	Shipping method ID	yes	yes

sitecategory

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
canonicalurl	string		yes	no
categorythumbnailhtml	string		yes	no
description2	string	The value from the web site category record for the field Brief Description.	yes	no
internalid	string	Internal ID of the site category.	yes	no
itemfields	string	An array of field names returned for each category.	yes	no
itemid	string	Name of site category	yes	no
metataghtml	string	The value from the web site category record for the field Meta Tag Html.	yes	no
pagetitle	string		yes	no
pagetitle2	string		yes	no
searchkeywords	string	The value from the web site category record for the field Search Keywords.	yes	no
storedetaileddescription	string		yes	no
storedetaileddescription2	string		yes	no
storedisplayimage	string		yes	no
storedisplayimagehtml	string		yes	no
storedisplaythumbnail	string		yes	no
storeurl	string		yes	no
urlcomponent	string		yes	no
urlcomponentaliases	array		yes	no
welcomepagetitle	string		yes	no

sitesettings

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
analytics	JSON analyticssettings object	Analytics settings	yes	no
checkout	JSON checkoutsettings object	Checkout settings	yes	no
cookiepolicy	string	Returns the URL for the cookie policy file in the NetSuite file cabinet.	yes	no
currencies	Array of JSON objects with fields (internalid, name, isdefault, symbolplacement)	Ordered list of currencies supported by the site `isdefault` field is a Boolean, with a value of **T** for default currency, **F** for others. `symbolplacement` uses the constants: SYMBOL_BEFORE_NUMBER = 1 and SYMBOL_AFTER_NUMBER = 2	yes	no
defaultshipcountry	string	Default ship to country	yes	no
defaultshippingmethod	string	Default shipping method	yes	no
displayname	string	Site name	yes	no
imagesizes	Array of JSON objects with image resize definitions (resizeid, resizeh, resizew)	Each object has the following properties: ▪ maxwidth–maximum resized image width in pixels ▪ maxheight–maximum resized image height in pixels ▪ urlsuffix–suffix that should be added to the image URL to return a resized image ▪ internalid–internal system ID of image resize definition. ▪ name–value from the Image Resize ID field (image resize definition) as entered on the Web Site Setup page.	yes	yes
iswebstoreoffline	boolean	Retruns T or F depending on whether the web store is online or offline.	yes	no
iswsdk	boolean	Returns T or F depending on the settings of the WSDK checkbox on the site record (tab setup) and the WSDK feature settings for a site.	yes	no
languages	Array of JSON objects with fields (internalid, name, isdefault)	Ordered list of languages for the site isdefault field is a Boolean, with a value of T for default language, F for others	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
loginallowed	string	Indicates whether login is allowed Value should be T or F.	yes	no
loginrequired	string	Indicates whether login is required to access the site Value should be T or F.	yes	no
order	JSON ordersettings object	Order settings	yes	no
paymentmethods	array of paymentmethod objects	Payment methods	yes	no
pricesincludevat	string	Indicates whether prices include VAT Value should be T or F.	yes	no
registration	JSON registrationsettings object	Registration settings	yes	no
requireloginforpricing	string	Indicates whether login is required to display item prices Value should be T or F.	yes	no
requireshippinginformation	boolean	Indicates whether shipping information needs to be collected Value should be T or F.	yes	no
shipallcountries	string	Indicates whether shipping is supported to all countries Value should be T or F.	yes	no
shippingrequired	string	Indicates whether shipping is required for the site Value should be T or F.	yes	no
shiptocountries	Array of strings	Countries where shipping is supported	yes	no
showextendedcart	string	Indicates whether extended cart should be displayed Value should be T or F.	yes	no
showshippingestimator	string	Indicates whether shipping estimator should be displayed Value should be T or F.	yes	no
sitetype	string	Indicates the type of web site. Value should be ADVANCED or STANDARD. ADVANCED indicates the site is a SuiteCommerce Advanced site where STANDARD indicates it is a Standard web site built with the NetSuite Site Builder.	yes	no
subsidiaries	Array of JSON objects with fields (internalid, name, isdefault)	Ordered list of subsidiaries for the site isdefault field is a Boolean, with a value of T for default subsidiary, F for others	yes	

Field Name	Field Type	Description	In Get Fun ctions?	In Set Fun ctions?
wsdkcancelcarturl	string	For use with an External Catalog Site (WSDK). Returns the Cancel Cart URL.	yes	no
wsdkcancelcheckouturl	string	For use with an External Catalog Site (WSDK). Returns the Cancel Checkout URL.	yes	no
wsdkcancelloginurl	string	For use with an External Catalog Site (WSDK). Returns the Cancel Login URL.	yes	no
wsdkcompletecheckouturl	string	For use with an External Catalog Site (WSDK). Returns the Complete Checkout URL.	yes	no
wsdkcompleteloginurl	string	For use with an External Catalog Site (WSDK). Returns the Complete Login URL.	yes	no

status

Each JSON object returned by a shopping function has an embedded status object that provides business logic validation of the web store order. The caller of a shopping function, in addition to checking the request status, should also check the status object to determine whether the object can be submitted to the backend and whether there are messages to display.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
code	string	Code of the status Following values are supported: ▪ error = object contains error ▪ incomplete = object is not complete, so cannot be handled by backend ▪ ready = object is complete and ready to be submitted to backend ▪ success = object successfully processed by backend		
fieldstatuses	array of status objects	Status for object fields		
messages	array of message objects	Status messages		
reasoncode	string	Code of possible reason for the current status		

subscriptions

This object contains information used for campaign subscriptions.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
internalid	string	The internalid of the campaign subscription.	yes	no

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
name	string	Name of the customer.	yes	no
subscribed	boolean	Indicates whether the customer is currently subscribed to this email campaign category.	yes	yes
description	string	Description of the campaign.	yes	no

threedsecure

This object contains information used for 3D Secure payment integration.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
noredirect	string	Default is F If set to T, when 3D Secure authorization is required, instead of a redirect, a status object that includes fields for 3D secure integration is returned.	no	yes
servicehtml	string	Complete HTML for IFrame that triggers 3D Secure service	yes	no
termurl	string	URL for page or handler where 3D Secure POST is sent	no	yes
threedsecurekey	string	Key used by backend to identify currenct 3D Secure data object	yes	yes

touchpoints

Each field in this object stores the URL for a custom page that is integrated to the web store at a selected entry point. These URLs are defined on the Touch Points subtab of the web application record. See the help topic Selecting Supported Touch Points.

Field Name	Field Type	Description	In Get Functions?	In Set Functions?
checkout	string	URL for starting the checkout process	yes	no
continueshopping	string	URL for continue shopping page	yes	no
customercenter	string	URL for customer center	yes	no
home	string	URL for home page	yes	no
login	string	URL for customer login	yes	no
logout	string	URL for customer logout	yes	no
register	string	URL for customer registration	yes	no
serversync	string	URL for sync shopper status between shopping and checkout domain	yes	no
viewcart	string	URL for viewing the shopping cart	yes	no
welcome	string	URL for welcome page	yes	no

Serversync Touchpoint

ⓘ **Applies to:** SuiteCommerce Advanced | Mont Blanc

You can customize your implementation of SuiteCommerce Advanced to use the serversync touchpoint. This touchpoint maintains the identity of the user and cart contents when a user opens items in other tabs or windows. This is useful during email marketing campaigns, when users open a separate instance of the application from a link in an email.

Using the serversync touchpoint requires customizing your implementation of SuiteCommerce Advanced to extend or override the following SuiteCommerce Advanced source files:

- CheckoutApplication@X.X.X\SuiteScript\checkout.environment.ssp
- Header@X.X.X\JavaScript\Header.Profile.View.js
- ProductList@X.X.X\SuiteScript\ProductList.Model.js
- Profile@X.X.X\SuiteScript\Profile.Model.js

> ⚠ **Important:** Making changes to core JavaScript source files or changing any vital functionality of the application can make migrating to future releases difficult. Before making changes to SuiteCommerce Advanced, see the help topics Customize and Extend Core SuiteCommerce Advanced Modules and Best Practices for Customizing SuiteCommerce Advanced

In the case of the serversync touchpoint, NetSuite has provided a customization patch that you can use.

> ⚠ **Important:** If you have already made customizations to your implementation of SuiteCommerce Advanced, use caution when applying these custom modules.

To apply the serversync touchpoint custom modules:

1. Click the link to download the serversync zip file.

   Serversync.zip

2. Extract the files, then place the Serversync directory and all contents in the Modules > extensions directory for your implementation of SuiteCommerce Advanced.

   > ⓘ **Note:** If you have already made customizations to the checkout.environment.ssp SuiteScript file in the CheckoutApplication@X.X.X module, run a diff between the new custom version of checkout.environment.ssp and your existing override, incorporate NetSuite's customizations into your custom module, and remove the EnvironmentCustom@1.0.0 folder from your new Serversync directory.
   >
   > Customizations you have made to the Header, ProductList, and Profile modules are only affected if you have modified the same methods and properties that have been extended in this patch. Run a diff on the appropriate files and, if you have made customizations in the same methods and properties, incorporate NetSuite's changes into your custom module.

3. Go to Modules > extensions > Serversync to locate the distro.json provided with the patch.
4. Open the new distro.json file and locate the appropriate Serversync entries for CheckoutApplication, Header, ProductList, and Profile.
5. Copy the Serversync entries and paste them into the list of modules in your existing distro.json file.

> ⚠ **Important:** If you have already made any customizations to the CheckoutApplication module, or customizations to the same methods and properties of the Header, ProductList, or Profile modules in your implementation of SuiteCommerce Advanced, do not use the Serversync entries from the new distro.json.

6. Open a command prompt from your source directory and deploy to NetSuite.

> ⓘ **Note:** Since these customizations modify files stored as SSP libraries, changes are not visible in your local environment. You must deploy your custom modules directly to NetSuite. See the help topic Deploy to NetSuite for more information.

SSP Application Governance

Applies to: SuiteCommerce Web Stores

SSP applications have a governance model in accordance with SuiteScript governance model, with the same goal of optimizing performance. SSP application governance is based on usage units. Each shopping method consumes a system-defined number of processing units, and each SSP application script can execute a system-defined number of units. If the number of allowable units is exceeded by an SSP application script, the script is terminated.

The limit for SSP applications is 1000 units per script.

The units consumed per method are as follows:

- Shopping Session Methods
  - nlobjShoppingSessionImpl.logIn(20)
  - nlobjShoppingSessionImpl.registerCustomer(20)
  - nlobjShoppingSessionImpl.registerGuest(20)
  - nlobjShoppingSessionImpl.setShopperCurrency(10)
  - nlobjShoppingSessionImpl.setShopperLanguageLocale(10)
  - nlobjShoppingSessionImpl.setShopperSubsidiary(10)
  - nlobjShoppingSessionImpl.getPaymentMethods(10)
  - nlobjShoppingSessionImpl.getCountries(10)
  - nlobjShoppingSessionImpl.getShipToCountries(10)
  - nlobjShoppingSessionImpl.getStates(10)
  - nlobjShoppingSessionImpl.getItemFieldValues(10)
  - nlobjShoppingSessionImpl.getSiteCategoryFieldValues(10)
  - nlobjShoppingSessionImpl.getInformationItemFieldValues(10)
  - nlobjShoppingSessionImpl.getMediaItemFieldValues(10)
- Customer Methods
  - nlobjWebStoreCustomerImpl.addAddress(10)
  - nlobjWebStoreCustomerImpl.updateAddress(10)
  - nlobjWebStoreCustomerImpl.removeAddress(10)
  - nlobjWebStoreCustomerImpl.addCreditCard(10)
  - nlobjWebStoreCustomerImpl.updateCreditCard(10)
  - nlobjWebStoreCustomerImpl.removeCreditCard(10)
  - nlobjWebStoreCustomerImpl.getAddressBook(10)
  - nlobjWebStoreCustomerImpl.getAddress(10)
  - nlobjWebStoreCustomerImpl.getCreditCards(10)
  - nlobjWebStoreCustomerImpl.getCreditCard(10)
- Order Methods
  - nlobjWebStoreOrderImpl.addItems(20)
  - nlobjWebStoreOrderImpl.addItem(10)
  - nlobjWebStoreOrderImpl.removeItem(10)

- nlobjWebStoreOrderImpl.updateItemQuantity(10)
- nlobjWebStoreOrderImpl.updateItemQuantities(20)
- nlobjWebStoreOrderImpl.removeAllItems(10)
- nlobjWebStoreOrderImpl.estimateShippingCost(20)
- nlobjWebStoreOrderImpl.applyPromotionCode(5)
- nlobjWebStoreOrderImpl.removePromotionCode(5)
- nlobjWebStoreOrderImpl.applyGiftCertificate(5)
- nlobjWebStoreOrderImpl.removeAllGiftCertificates(5)
- nlobjWebStoreOrderImpl.setShippingAddress(10)
- nlobjWebStoreOrderImpl.setShippingMethod(5)
- nlobjWebStoreOrderImpl.getAvailableShippingMethods(20)
- nlobjWebStoreOrderImpl.getAvailableShippingMethod(10)
- nlobjWebStoreOrderImpl.submit(20)

Note: There is not a concurrency limit for SSP applications but they are not recommended for high volume catalog pages or to deliver content that should come from a cache.

SuiteScript API & SuiteCommerce

Applies to: SuiteCommerce Web Stores

SuiteScript is a JavaScript API that enables you to programmatically access most NetSuite records, custom records, and event/trigger points. This API is a standard NetSuite feature. For complete details on all of SuiteScript including all APIs and the usage of those APIs, see the help topic SuiteScript 1.0.

SuiteCommerce Advanced uses SuiteScript to access data stored in an NetSuite record. This enables SuiteCommerce Advances to access data that is not accessible from the Item Search API or the Commerce API.

SuiteCommerce Advanced uses SuiteScript in the following contexts:

- Backend models: all backend models are combined into a file called ssp_libraries.js. This file is stored in the root directory of the SSP application.
- Services: all services in SuiteCommerce Advanced are deployed to the `services` directory within the SSP application.

Being installed within the SSP application enables backend models and services to access the SuiteScript API to access data stored in NetSuite records.

The Case application module, for example, uses SuiteScript to access information stored in a NetSuite case record. When creating a new case, `Case.Model` uses the following code:

```
var case_record = nlapiCreateRecord(’supportcase’);
```

The string passed to the `nlapiCreateRecord` method is the internal identifier for the case record in NetSuite. Calling `nlapiCreateRecord` creates a new case record in NetSuite. It also returns a copy of the object. This object is stored in the `case_record` variable.

Storing a record as an object enables you to call additional methods on the object containing the case record. For example, the following code calls the `getField` method that returns the value of the record field passed to the method.:

```
var category_field = case_record.getField(’category’);
```

In this example, `getField` returns the value of the category field.

Other modules use the `nlapiLoadRecord` method to assign a NetSuite record to a variable containing an object. You can then call methods like `getField` on the object to retrieve the values of specific fields within the record.

In another context, a model may call the `nlapiDeleteRecord` method to remove a record from NetSuite.