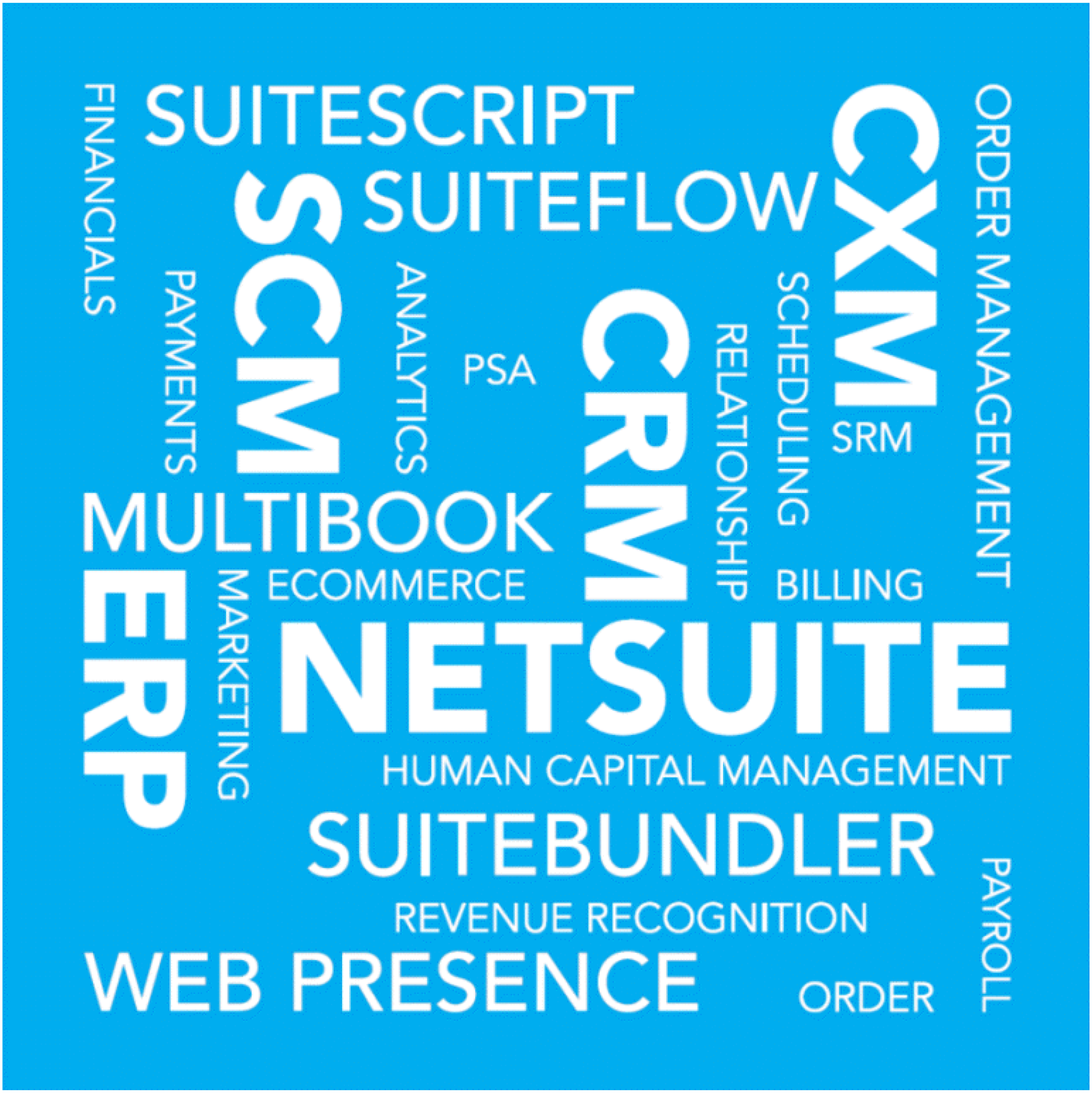
Commerce Search



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

SuiteCommerce Item Search 1

Search Settings Overview 1

Select Search Fields 3

Select Facet Fields 5

Select and Configure Sort Fields 5

Define Field Sets 6

Improve Relevance of your Search Results 10

Troubleshoot Unexpected Search Results 11

Improve the Performance of the Search Results Page 12

Search Index Overview 15

Item Search API Overview 17

The Base URL 18

Item Search API Input Parameters 19

Item Search API Output Response 26

Sample Item Search Queries and Results 29

Synonyms 34

SuiteCommerce Item Search

Applies to: SuiteCommerce Web Stores

When shoppers visit your SuiteCommerce website, you want them to quickly find the items that they are looking for with minimum navigation. The Item Search service speeds up this process by returning search results for the search query provided by the shopper.

The three main factors that impact the search results are:

- Your product catalogue
- Your item search settings
- The search query provided by the shopper

Although you cannot control the search query that your shoppers provide, you can make an educated guess of what they might search for and configure the search settings accordingly. For information on item search settings, see Search Settings Overview.

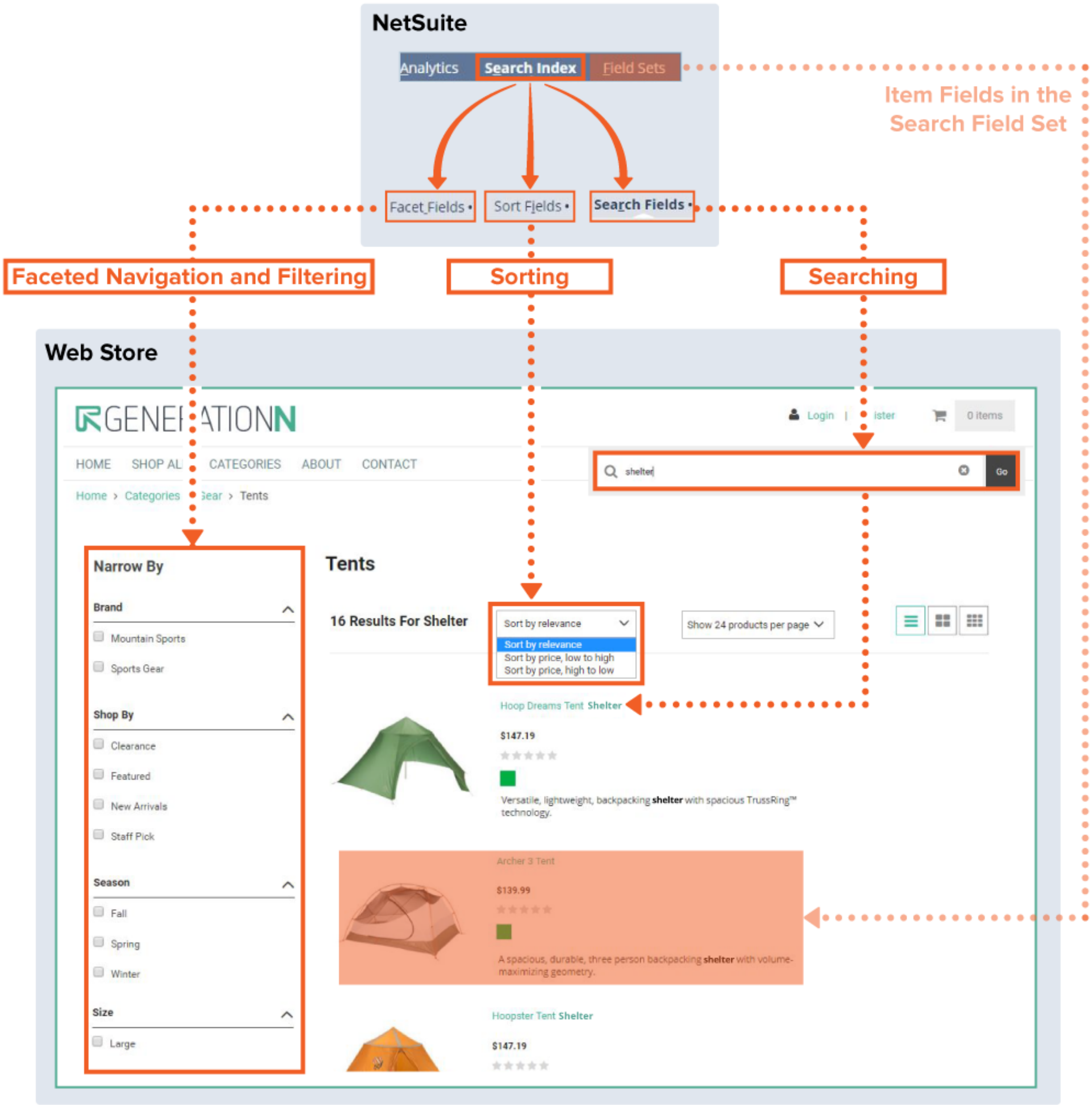
When you configure and save your item search settings for the first time, the Search Index is built. The Search Index is defined by the fields that you configure in your search settings. For information on the Search Index, see the help topic Search Index.

The search index process prepares your item records for use with the Item Search API. The Item Search API is an interface that helps you retrieve the indexed items. By specifying the input parameters with the Item Search API, you can retrieve a subset of indexed items in the JSON response. Web Developers can use this API to build facet filtering and keyword search on item data. For information on Item Search API, see the help topic Item Search API.

Search Settings Overview

View Related Video

You can enhance the item search capabilities of your website by configuring the item search settings with your preferences. The following illustration shows how item search settings relate to your web store:



- **Search Fields** influence the default sort order of the search results, which is relevance. For more information, see the help topic Search Fields.
- **Facet Fields** help a shopper narrow down the search results. For more information, see the help topic Facet Fields.
- **Sort Fields** define how results can be sorted beyond relevance to a search query. For more information, see the help topic Sort Fields for Search Results.
- **Field Sets** are a grouping of item fields taken from item records, used to display information to your shoppers. For more information, see the help topic Field Sets.

Prerequisites

Before you configure the item search settings, you should:

- Complete Item Configuration for Web Stores, which includes setting up item records for SuiteCommerce website, setting up item records for search index, and setting up item inventory in web stores.

- Review the FieldSetsList spreadsheet, which lists all the fields that are available for keyword search, faceting, sorting, and for use in field sets.
- Review the Sample Search Fields Setup.
- Read the best practices to Improve Relevance of your Search Results.
- Identify the item record fields that contain keywords, which shoppers might use to search for products on your website.
- Identify the item record fields that shoppers might use to efficiently narrow down the search results.
- Create necessary custom fields for use as facet fields.
- Identify the item record fields that shoppers might use to sort the search results.
- Create necessary custom fields for use as sort fields.
- Identify the item record fields that you want to include in each field set.

Select Search Fields

When a search is initiated in the web store, the system searches based on the type of match and order of the fields configured on the Search Fields subtab.

Select the Right Type of Match

Before you select the type of match for your Search Fields, review the usage of each type of match. Also, observe the definitions for each type of match as it would be applied to the item attribute text "wifi router":

Type of Match	Description	Usage
Exact	Matches the exact text of the item field contents. For example, if the shopper types “wifi router” in the search box, the search matches “**wifi router**.”	Exact match holds good in situations where a shopper might provide the exact search string when searching for a product. Ideal for the SKU and Online Name fields where there are chances that a shopper remembers the full name and provide the exact search string. **Note:** If you select the same search field multiple times and use it with different types of match, define exact match at the top of the list for that search field.
Keyword	Appears anywhere in the text. For example, if the shopper types “router” in the search box, the search matches “wifi **router**."	Keyword holds good for fields with a lot of text. Ideal for fields with description such as Detailed Description or in case of the Online Name field where shoppers might search for only part of the full item name.
Starts with	Behaves the same as Keyword search but only for the first 20 characters of every word in a field. For example, if the shopper types "wifi" in the search box, the search matches "**wifi** router."	You should use Starts With only if your site uses Type Ahead search and only use it on fields such as Online Name.
Starts With Exact	Matches the exact text from the beginning of a field but only up to the first 20 characters. For example, if the	Starts With Exact holds good in situations where a shopper provides the exact first

Type of Match	Description	Usage
	shopper types “wifi” or “wifi rou” in the search box, the search matches “**wifi router**.”	few characters (upto 20 characters) as the search string. Ideal for the SKU field where a shopper might provide only the first few characters of the SKU number. **Note:** If all your SKU numbers are less than 20 characters, it is possible to only use Starts With Exact for the SKU field instead of Exact.
Fuzzy	Matches similar text or misspellings depending on the length of the search term specified. Following rules are applicable when the type of match is Fuzzy: ■ Fuzzy matching is not applied for search term with three or fewer characters. ■ Search term with four or five characters will fuzzy match against one character being different. ■ Search term with six or more characters will fuzzy match against up to two characters being different. For example, if the shopper types “mifi” in the search box, the search matches “**wifi** router.”	Since fuzzy matches can return irrelevant items (skirt versus shirt) in the search results, use the Fuzzy type of match only if you have a valid use case. If you wish to include the Fuzzy type of match in your Search Settings, put it at the bottom of the list. Since the list of search fields is ranked in order, top to bottom, fuzzy matches near the top of the list can return irrelevant items at the top of the search results list.

Sample Search Fields Setup

NetSuite provides default search field settings to get you started. However, it is recommended that you reconfigure the Search Fields based on the data in your item records.

You can also consider the following search fields setup as a starting point for your environment:

Search Field	Item Record Field Name	Field ID	Search Type
Name	ITEM NAME/NUMBER	itemid	Starts With Exact
Online Name	WEB STORE DISPLAY NAME	storedisplayname	Exact
Online Name	WEB STORE DISPLAY NAME	storedisplayname	Starts With
Online Name	WEB STORE DISPLAY NAME	storedisplayname	Keyword
Detailed Description	DETAILED DESCRIPTION	storedetaileddescription	Keyword
Search Keywords	SEARCH KEYWORDS	searchkeywords	Keyword

To select the search fields

1. Complete the prerequisite tasks as described in Prerequisites.
2. Go to Setup > SuiteCommerce Advanced > Set Up Web Site.
3. Click **Edit** next to your SuiteCommerce Advanced site.
4. Click the **Search Index** sub tab and then click **Search Fields**.
5. If the default setting is relevant to your website, go to Step 8 without making any changes.
6. (Optional) If you wish to rearrange the search fields, you can drag and drop rows.
7. (Optional) If you wish to delete a search field, you can select the field and click **Remove**.

8. (Optional) If you wish to add a new search field:
    1. Select the search field from the **Search** dropdown.
    2. Select the match type for the Search Field from the **Type of Match** dropdown.
    3. Click **Add**.
9. Once you have finalized the Search Fields, click **Save**.

Select Facet Fields

Facets are a way of filtering the search results based on the values in the item fields. When a shopper clicks a facet value, the search results are reduced to only the items that have that value, and additional clicks continue to refine the search results.

Typically, a variety of product attributes are used as facets. For example, some of the popular facet fields for an apparels website are brand, gender, color, and size.

To select the facet fields:

1. Complete the prerequisite tasks as described in Prerequisites.
2. Go to Setup > SuiteCommerce Advanced > Set Up Web Site.
3. Click **Edit** next to your SuiteCommerce Advanced site.
4. Click the **Search Index** sub tab and then **Facet Fields**.
5. From the **Facet Field Name** dropdown, select the fields that you would like to use as facets and click **Add**.
6. (Optional) If you wish to delete a Facet Field from the list, you can select the field and click **Remove**.
7. Once you have finalized the Facet Fields, click **Save**.

Select and Configure Sort Fields

Using a sort field, a shopper can sort the search results by that field instead of the default relevance field. It may not always be obvious on which field you should or should not sort, but you need to ensure that all item records contain data for the fields you choose as sort fields, especially if you select an item record field where values are optional. Items with empty fields are placed last when sorting by that field. For example, when sorting by price, the search service will place items that have no price last.

> **Note:** Relevance is the default sort field, and you cannot remove Relevance from the list. But you can select other item fields to use as sorting options for your site. By default, the `onlinecustomerprice` field is added as an additional sort option, and additional configuration is not required to support sorting by price. SuiteCommerce Advanced is pre-configured to display two pricing options: Sort by price, low to high and Sort by price, high to low.

Adding a Sort Field under Search Index enables sorting for that field. You can then use the sort field with the Item Search API but that sort field is not visible in the web store until you configure it.

To add the sort fields:

1. Complete the prerequisite tasks as described in Prerequisites.
2. Go to Setup > SuiteCommerce Advanced > Set Up Web Site.
3. Click **Edit** next to your SuiteCommerce Advanced site.

4. Click the **Search Index** subtab and then **Sort Fields**.
   - **Field Name** — (Required) Select an item record field. Select item attributes that you want shoppers to use for sorting search results.
   - **Field ID** — (View only field) Displays the field's internal ID.
   - **Sort Order** — Choose ascending or descending order. This is the order in which products are sorted in search results.

Configuring a Sort Field ensures that the sort field is available to the shoppers in the SuiteCommerce Advanced web store.

To configure the sort fields

1. Select the domain to configure at Setup > SuiteCommerce Advanced > Configuration.
2. In the SuiteCommerce Configuration record, navigate to the **Search** tab and then click the **Result Sorting** subtab.
3. Define your sort options. For details, see the help topic Result Sorting Subtab.

   For example, if a custom field, `custitem41` has been configured as a sort field in the web site record, a line item for that field must be added to the Desktop, Phone, and Tablet sections before the selection is available on the web store in the applicable browser.

To configure sort fields for pre-Vinson SCA implementations, the `sortOptions` array in the `SC.Shopping.Configuration.js` configuration file must be extended to include the new sort field. For guidance on extending your implementation, see the help topic Customize and Extend Core SuiteCommerce Advanced Modules.

For example, if a custom field `custitem41` has been configured as a sort field in the web site record, a line item for that field must be added to the `sortOptions` array before the selection is available on the web store.

```
sortOptions: [
   {id: ’relevance:asc’, name: _(’Sort by relevance’).translate(), isDefault: true}
,   {id: ’onlinecustomerprice:asc’, name: _(’Sort by price, low to high’).translate()}
,   {id: ’onlinecustomerprice:desc’, name: _(’Sort by price, high to low ’).translate()}
,   {id: ’custitem41:desc’, name: _(’Sort by best seller, low to high ’).translate()}
]
```

Define Field Sets

You can expose information from the item records in NetSuite to your website by defining field sets. Once you define the field sets, a web developer can use the Item Search API to access field values in the field sets.

For example, you might create a field set for the product details page, which includes fields that display product name, product images, product description, product availability details, and price. You could also create a different field set for the search results page, which includes fields that just display an image thumbnail, product name, and price.

> **Note:** Field Sets are made public on the internet. Since any field type can be selected for use in field sets, you must consider which fields you choose to expose.

You can leverage the Field Set Setup Script to quickly populate your site with required Field Sets. To run a script to create the field sets, see Import Field Sets.

You can also extend SuiteCommerce Advanced by defining your own field sets on the Web Site Setup page. To define your own field sets, follow the inline procedure.

To define your own field sets:

1. Complete the prerequisite tasks as described in Prerequisites.
2. Go to Setup > SuiteCommerce Advanced > Set Up Web Site.
3. Click **Edit** next to your SuiteCommerce Advanced site.
4. Click the **Field Sets** subtab.
5. In the **Name** field, enter a name for the new Field Set. This name is only displayed internally on the Field Sets list page and in dropdown lists on the Web Site Setup page. Name each Field Set based on the product views you create for your site.

   > ⚠ **Important:** The Field Set to return item details is typically named **details**. If a different name is used, you must update the `searchApiMasterOptions` property. See Search Results Subtab for more details on configuring this property.

6. In the **Field Set ID** field, enter an ID, which you need to later use with the Item Search API as an input parameter to specify the field set. The Field Set ID must be alphanumeric, lowercase, and contain no spaces or special characters. The Field Set ID is displayed internally, on the Field Sets list page and in dropdown lists on the Web Site Setup page.
7. The **Record Type** field is pre–populated with the value Item as only item records are supported.
8. The **Description** field is an optional field where you can enter a description of this field set. The description only shows on Field Set records, the Field Set list page, and on the Web Site Setup page.
9. To populate the **Fields Included in Field Set** field, click the Set icon next to this field. A Field Set window opens.
10. In the Field Set window, select the item record fields from the Field Name dropdown one by one and click Add. Once you have selected all the fields, click Submit. The values of the **Field ID** and **Field Type** fields are auto populated for the selected field. You can use the Field ID to refer to specific fields when writing SuiteScript. This is also the field ID returned in the JSON response when using the Item Search API. The Field Type refers to the NetSuite field type. Field types provide information about the type of data web developers can expect in the JSON response to Item Search API queries.
11. After you have selected fields for this field set, click **Submit**. The Field Set window closes.
12. Click **Add** to add the new field set.
13. Click **Save** on the Web Site Setup page to save your changes.

Defining field sets is the final task in configuring SuiteCommerce Advanced item search settings. When you click Save on the Web Site Setup page after defining field sets, the item search indexing process is triggered. For more information, see the help topics Search Index and Checking Search Index Job Status.

Field Set Reference

We have provided a Microsoft Excel worksheet listing all the fields available for field sets. You can use this list to find information such as, field IDs and field types. The list also identifies fields that are available for faceting, sorting, and keyword search. In addition to the fields on the item records, there are additional fields that include derived values. These fields are identified on the worksheet as Synthetic fields.

To access the worksheet, click this link: FieldSetsList.xls

Note: Depending on the features enabled in your account, some of the fields in the spreadsheet may or may not appear in your account.

Derived Fields for Field Sets

In addition to the fields on the item records, there are additional fields that include derived values. These fields are known as derived fields or synthetic fields. You can use these derived fields in field sets to retrieve additional information.

Derived Field Name	Field ID	Description
Available for Backorder	Isbackorderable	Displays a Boolean value, which indicates whether an item can be back ordered when it is out of stock. One of the following conditions must be met for an item to back orderable: - The item's out of stock behavior is set to **Allow back orders with no out-of-stock message** or **Allow back orders but display out-of-stock message**. - The item's out of stock behavior is set to **Default** and the site is configured to display out of stock items.
Available for Purchase	ispurchasable	Displays a Boolean value, which indicates whether an item is in stock or back orderable.
Commerce Categories	commercecategory	Displays the Name, ID, URLs, and URL fragment aliases of each Commerce Category containing the item. **Note:** The same details are also displayed for each category in the primary path to the item when navigating through the category hierarchy.
Correlated Items (Detail)	correlateditems_detail	Displays the details of all the correlated items associated with an item. The `correlateditems` fieldset defines the fields to be returned for each correlated item. **Note:** Even if you have defined the default field set `correlateditems` as described in Import Field Sets, you can overwrite the default field set with any other field set using the Item Search API parameter `correlateditems_fieldset`. For more information, see Item Search API Input Parameters.
Default Category (Detail)	defaultcategory_detail	Displays the path to the item's default Site Builder category on the current site with the help of label and URL component fields.
Default Item Ship Method	defaultitemshipmethod	Displays the default shipping method for the specified item.
Display Out-of-Stock Message	showoutofstockmessage	Displays a Boolean value, which indicates whether a message should be displayed to indicate that the item is out of stock. The item must be out of stock and one of the following conditions must be met for the out of stock message to be displayed: - The item's out of stock behavior is set to **Allow back orders with no out-of-stock message** or **Allow back orders but display out-of-stock message**.

Derived Field Name	Field ID	Description
		■ The item's out of stock behavior is set to **Default** and the site is configured to display out of stock items.
In Stock	isinstock	Displays a Boolean value, which indicates whether an item should be considered to be in stock depending on the item type and item availability.
Item Carrier	itemcarrier	Displays the carrier used to ship an item. For example, UPS, FedEx, and so on.
Item Images (Detail)	itemimages_detail	Lists the URL and alt text of all the images associated with the item. **Note:** The default URL for each image is displayed, not the URL of the resized image. For information on image file naming convention, see the help topic Renaming Image Files with an Item Identifier.
Item Options (Detail)	itemoptions_detail	If an item has item options, this field returns all the item options, their values in case of list-type options, and their default values. For example, for gift certificate items, this field returns the sender name, recipient name, recipient email address, and message. For matrix items, this field indicates whether the item is a matrix parent or child. Also, for matrix child items, the field returns the internal ID of the matrix parent item.
Matrix Child Items	matrixchilditems_detail	Displays the details of all the matrix child items that exist for the matrix parent item, including which item option values define each child item. The `matrixchilditems` fieldset defines the fields to be returned for each matrix child item. **Note:** Even if you have defined the default field set `matrixchilditems` as described in Import Field Sets, you can overwrite the default field set with any other field set using the Item Search API parameter `matrixchilditems_fieldset`. For more information, see the help topic Item Search API Input Parameters.
Price (Formatted) and Price Level (Formatted)	pricelevel*_formatted	Each price field has a formatted counterpart. For example, the field ID of the Base Price field is `pricelevel1`. The field ID of the derived field Base Price (Formatted) is `pricelevel1_formatted`. The regular price field returns a numeric value. The `*_formatted` variant of the price field returns the formatted price with a currency symbol along with appropriate number of decimal places based on regional rounding conventions.
Price for Current Customer (Detail)	onlinecustomerprice_detail	Returns a JSON object with all the available information about the item's current price. If a customer is logged in, the price displayed on the website applies to the logged-in customer, including special pricing and quantity pricing. If the item has quantity pricing, the price schedule is returned. If the item is a donation item, information defining the donation parameters such as minimum and default amount are included. This field consists of the following subfields: ■ onlinecustomerprice - Price for the current customer, or in case of matrix items, this is the lowest child item price.

Derived Field Name	Field ID	Description
		- onlinecustomerprice_formatted - Formatted version of onlinecustomerprice - priceschedule - Pricing for specified quantities of the item, which can be set up on the item record. - Nopricemessage – This message is displayed when the price is unavailable for an item. - Maximumvariableamount – This subfield is displayed if the item is a donation item and it displays the maximum donation amount. - Showdefaultvariableprice - This subfield is displayed if the item is a donation item and it displays the item's show default amount option.
Related Items (Detail)	relateditems_detail	Displays the details of all the related items associated with an item. The *relateditems* fieldset defines the fields to be returned for each related item. **Note:** Even if you have defined the default field set `relateditems` as described in Import Field Sets, you can overwrite the default field set with any other field set using the Item Search API parameter `relateditems_fieldset`. For more information, see the help topic Item Search API Input Parameters.

Improve Relevance of your Search Results

Following are some of the best practices for improving the relevance of your search results:

- For items displayed in the web store, ensure that the items records are complete. At least specify values for all the fields of the item record that are being used as search fields, facet fields, and sort fields.
- Your settings on the Search Fields tab influences the default sort order of the search results. Therefore, identify item record fields that contain keywords that customers might use to find products, and then select those fields on the Search Fields subtab. For more information, see Sample Search Fields Setup.
- The top row on the Search Fields list shows the item field and the type of match that result in highest relevance ranking. Therefore, define a search field at the top of the list whose value is most likely to be searched by shopper. For example, define Online Name (Field ID: storedisplayname) or Name (Field ID: itemid) at the top of the list.
- If you select the same search field multiple times and use it with different types of match, define exact match at the top of the list for that search field.
- Use the Starts With type of match for the Online Name search field only if your site uses Type Ahead Search.
- Since fuzzy matches can return irrelevant items (shirt versus skirt, or cat versus hat) in the search results, use the Fuzzy type of match only if you have a valid use case. If your intention is to return accurate ordering of search results (for example, return all items that match **Shirt** at the top of list), then do not include fuzzy match fields at the top of the list of Search Fields. Since the list of search fields is ranked in order, top to bottom, fuzzy matches near the top of the list can return irrelevant items at the top of the search results list.
- Avoid adding excessive search fields as it can have performance implications and result in less relevant search results.

- Only certain fields on item records are supported for faceted navigation based on the field type. For more information, see Field Set Reference.
- Avoid adding excessive Facets or Facet Values. For more information, see the help topic SEO Page Generator Best Practices.
- You must limit the length of facet field values to 200 characters. If the facet field value exceeds the maximum character length, the search index fails to rebuild.
- If Google Analytics is enabled in your account, you can see analysis of the search terms used by your shoppers in due course. In your Google Analytics account, you can navigate to Behaviour > Site Search > Search Terms to view all the search terms. By analyzing these search terms, you may identify areas in you Search Settings that you can tweak to provide even more relative search results to the shoppers visiting your website.

Troubleshoot Unexpected Search Results

Search results are driven by the items in your item catalog, data in your item record, configurations in your search settings, and the search query that is provided. Based on these factors, you might get any of the following in your search results:

- Relevant items are returned at the top
- Irrelevant items are returned before relevant items
- Irrelevant items are returned intermittently
- Relevant items are not returned
- No items are returned

Relevant items at the top implies that you have configured the right fields in your search settings for the items published in your web store. Therefore, no action is required at this point in time.

However, if Google Analytics is enabled in your account, you can analyze the search terms used by your shoppers to refine your Search Settings as described in Improve Relevance of your Search Results.

Irrelevant items might be returned before relevant items because of your Search Settings. Modify you Search Settings to see if you can get better search results.

Irrelevant items might be returned intermittently because of stemming, which can be sub-optimal in some situations. Stemming is a process of reducing words to their root form, and words are stemmed at both index and query time. Consequently, when two or more words have the same stem, and when the type of match for the search field is set to Keyword, irrelevant items might be returned intermittently in the Search Results along with relevant items.

For example, generator, generic, and general are all stemmed and stored as `gener` in the Search Index, and all these items are returned in the search results when you search for generator.

If **relevant items are not returned** even though the items being searched are published in your web store, you might have added a different field under Search Fields or you might not have assigned the right type of match for the selected Search Field.

For items to be searchable:

- The field on the item record and the field in our search fields sublist should be the same. For example, the Web Store Display Name field on the Item Record is represented as Online Name in the Search Fields dropdown. Therefore, to ensure that you have selected the intended field, always verify the Field ID of a field using the FieldSetsList spreadsheet.

- Item fields that you add under Search Fields should have values assigned to them in the item record.
- The type of match that you assigned for the selected Search Field should be appropriate for the data in your item record and the search query provided.

If **no items are returned** in the search results even though you have configured the right fields in your search settings, and the item being searched is published in your web store, it could be because the Item Search API is not returning items. To troubleshoot this issue, rebuild the search index manually as described in Manually Rebuilding the Search Index.

Improve the Performance of the Search Results Page

You can improve the performance of search result pages in SuiteCommerce Advanced by tweaking the field sets.

Prerequisites

You have:

- Complete understanding of Item Search API, including the usage of Item Search API Input Parameters.
- Complete understanding of Field Sets.

Background

Fields such as `Matrix Child Items (Detail)`, `Related Items (Detail)`, and `Correlated Items (Detail)` use the default field sets and retrieve a large number of item fields for each item. Consequently, the performance of pages that retrieve lot of matrix child items, related items, or correlated items degrade.

For example, the `Matrix Child Items (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
```

Solution

When retrieving item fields for the search result pages, you can specify an alternate field set to be loaded instead of the default field set using the Item Search API input parameters.

Use the following Item Search API parameters to override the default field set with any other field set:

- For the `Matrix Child Items (Detail)` field, use the Item Search API parameter `matrixchilditems_fieldset`.
- For the `Related Items (Detail)` field, use the Item Search API parameter `relateditems_fieldset`.

- For the `Correlated Items (Detail)` field, use the Item Search API parameter `correlateditems_fieldset`.

Example

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 `matrixchilditems_mini`, which has only four item fields. For more information, see the help topic To define fields sets:.

In the following example, fewer matrix child item fields are returned as we have overwritten the default field set with a field set `matrixchilditems_mini` (using the Item Search API parameter `matrixchilditems_fieldset`):

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

Search Index Overview

The Search Index powers item search by ensuring that the search times are reduced and requests are more targeted. The Search Index picks up the item attributes that you want to expose on your site including the fields selected for search, faceted navigation, sorting, and field sets. The search index is first built when you click Save on the Web Site Setup page after selecting the item search settings for the first time. Whenever you make changes to your item record or item search settings, the search index is rebuilt factoring in the changes that you made.

> **Note:** Deleting a SuiteCommerce Advanced Web Site deletes the index for the site.

To learn more about the Search Index, refer to the following sections:

- Events that Trigger Search Index Rebuild
- Events that Trigger Search Index Update
- Check the Search Index Job Status
- Rebuild the Search Index Manually

Events that Trigger Search Index Rebuild

Any of the following changes in your NetSuite account trigger a search index rebuild:

- Changing facets, search, and sort options (on the Search Index subtab of the Web Site Setup page).

  > **Note:** When you remove a search field and save your changes, the search index is rebuilt only if you have removed all types of matches for that particular search field.

- Adding or removing languages.
- Adding or removing Subsidiary accounts.
- Changing site currencies when the site has currency-based fields defined as facets or sort options. For example, fields such as price or cost.
- Changing the preference, **Include Out Of Stock Items in Web Store** on the Web Site Setup Page.

> **Note:** Saving an item record triggers an index **update**, not an index rebuild. For information on events that triggers a search index update, see Events that Trigger Search Index Update.

Events that Trigger Search Index Update

Any of the following changes in your NetSuite account trigger a search index update:

- Item record changes involving fields that are configured in your search settings.

> **Note:** This includes changes on item records as a result of using a CSV import.

- Item Availability
- Item Pricing
- Currency Exchange Rates
- URL Components for Facets
- Web Site Tabs
- Site Builder Categories
- Commerce Categories

For information on events that trigger the search index to rebuild, see Events that Trigger Search Index Rebuild.

Check the Search Index Job Status

You can use the Job Status page to check the status of the search index process. Navigate to Setup > SuiteCommerce Advanced > Set Up Web Site, and click View in the Search Index Status column.

The Job Status page shows elapsed time as the item search index is processing. Click **Refresh** to update information on the page. The search index process consists of the following tasks:

1. Dropping the search index
2. Updating search index, whichs includes, adding, updating, and deleting items
3. Rebuilding the search index core

> **Note:** Changes to the item record trigger an index update. The Records Finished column on the Job Status page always indicates the total number of records being processed.

The job names on the Job Status page describe whether the search index is being built for the first time, rebuilt or updated. The job names also provide information about why the index is being updated.

The following table lists all possible job names; your NetSuite configuration determines which of these you will see.

Job Name	Description
Create	The full search index is being built for a newly created site.
Reindex for site config change	The full search index is being rebuilt because changes were made to the site configuration.
Index regeneration	The full search index is being rebuilt because an ad hoc rebuild has been triggered, or a feature has been enabled or disabled that requires a new search index to be built.
Index is live	The newly rebuilt search index is being made available for search queries.
Feature Change	The search index is being updated because a feature was toggled, for example, Commerce Categories.
Item Type Feature Change	The search index is being updated because a feature related to item type was toggled, for example, the Download Items and Gift Certificates features.
Item Update	The search index is being updated because one or more items were added or updated.

Job Name	Description
Item Removal	The search index is being updated because one or more items were removed from the webstore.
Commerce Category Update	The search index is being updated because a Commerce Category was added, updated, or deleted.
Site Builder Category Update	The search index is being updated because a Site Builder Category was added, updated, or deleted.
Synonyms Updated	The full search index is being rebuilt due to the addition, modification, or removal of Synonyms.

Rebuild the Search Index Manually

Even though certain changes that you make in your NetSuite account automatically trigger the search index to rebuild, you can manually rebuild the search index for your SuiteCommerce Advanced Site. For information on events that trigger the search index to rebuild, see Events that Trigger Search Index Rebuild.

Manual rebuild of the search index is typically done when you want to force a rebuild, and not wait for the changes that you made in your NetSuite account to automatically trigger the rebuild.

After you initiate the search index rebuild process, it may take up to 10 minutes to start rebuilding the search index. The time it takes to complete the process is relative to the number of items being indexed.

> **Note:** There is a limit to the number of times you can request a rebuild of the search index. In a 24-hour period, you are allowed only two requests.

To manually rebuild the search index for your SuiteCommerce Advanced Site:

1. Go to Setup > SuiteCommerce Advanced > Set Up Web Site.
2. Click **Edit** next to your website.
3. On the Web Site Setup page, choose Actions > Rebuild Search Index.

   A popup message appears stating that there is a limit to the number of times you can request a rebuild of the search index.
4. In the popup message dialog box, click **OK** to rebuild the search index for this website.
5. (Optional) Open the Job Status page to check the status of the rebuild process. For more information, see Check the Search Index Job Status.

Item Search API Overview

The Item Search API is a configuration driven API, which is determined by your Search Settings. For example, items returned in the API response data can be filtered by the facets you defined on the Search Index subtab when you set up Search Settings. For more information, see Search Settings Overview.

The Item Search API uses a REST/JSON style of communication, and supports searching, sorting, faceting, pagination, or any combination of these as they can operate independently. Web Developers can use this API to build facet filtering and keyword search on item data.

Note: To use the Item Search API, you must have a SuiteCommerce Advanced website, and you must have configured item search settings for your site.

The Item Search API returns data from the item records that have been indexed. However, some items might not be returned depending on the item availability in a given subsidiary at the time of indexing. Items are retuned in the Item Search API response 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 a Commerce 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).

For more information about the search index, see the help topic Search Index.

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

To learn more about the Item Search API, refer to the following sections:

- The Base URL
- Item Search API Input Parameters
- Item Search API Output Response
- Sample Item Search Queries and Results

Important: If you are using an earlier release of SuiteCommerce Advanced, you require a patch to ensure that the Item Search API response is cached in the CDN. For more information, see the help topic Item Search API Response Data not Cached.

The Base URL

The Item Search API 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 base URL has the following format:

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

The components of the base URL is described here:

- **Custom Domain** – A unique URL is required, which must be a custom domain. For example, `www.mywebstore.com` is a custom domain.
- **Items** – This API is designed to query item records in your NetSuite account. Note that `/items` in the URL is part of the path and it is required. You cannot assign a value to it.
- **Parameters** – 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 (&). Parameters are typically name-value pairs, such as `fieldset=details`. For more information, see the help topic Item Search API Input Parameters.

> **Note:** JSONP (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 as described in Callback Function, and then create the variable name. For more information about the jQuery API, see api.jquery.com/jQuery.getJSON/.

Item Search API Input Parameters

The Item Search API Input Parameters are used to construct query strings that return item record attributes in a JSON response. To create a query string for the Item Search API, you should use the input parameters with the base URL `http://www.mywebstore.com/api/items?{parameters}`. You can also use multiple parameters with the base URL by separating them with an ampersand (`&`).

> **Note:** Some parameters when used with the base URL just return the item IDs of the matching items. However, if you wish to retrieve additional item details, you can use the parameters listed under the Retrieve Additional Item Details section with these parameters.

The base URL when used without any parameters retrieves the item IDs of all the items. To retrieve only the desired items and item details, you can specify the following input parameters:

Parameters	Usage

`c`
`n`
`region`
	c (optional)

`id`	id (optional)
`url`	url (optional)
`q`	q (optional)

`commercecategoryid`
`commercecategoryurl`
`commercecategoryname`
	commercecategoryid (optional)

`fields` `fieldset`	fields (optional)

`matrixchilditems_fieldset`
`correlateditems_fieldset`
`relateditems_fieldset`
	correlateditems_fieldset (optional)

`limit` `offset`	limit (optional)

`<field_name>`
`<numeric_facet_field_ID>.ranges`
	facet.exclude (optional)

Parameters	Usage

`<numeric_facet_field_ID>.from`
`<numeric_facet_field_ID>.to`	

`include` `facet.exclude`	include (optional)
`sort`	sort (optional)
`pricelevel`	pricelevel (required)
`currency`	currency (optional)
`language` `country`	country (optional)
`callback`	callback (optional)
`ssdebug`	debug (optional)

Retrieve Items by NetSuite Account Number, SuiteCommerce Site ID, Subsidiary ID

- `c` — You can use the parameter `c` to specify the NetSuite account and retrieve all the items in the specified NetSuite account. For example, if your NetSuite account is 3925062, you can use the following query string to retrieve all the items that belongs to your NetSuite account:

  ```
  http://www.mywebstore.com/api/items?c=3925062
  ```

- `n` — You can use the parameter `n` to specify the ID of your SuiteCommerce website and retrieve all the items in the specified SuiteCommerce website. For example, if the ID of your SuiteCommerce website is 3, you can use the following query string to retrieve all the items that belongs to this SuiteCommerce website:

  ```
  http://www.mywebstore.com/api/items?n=3
  ```

- `region` — You can use the parameter `region` to specify the Subsidiary in a NetSuite OneWorld account and retrieve all the items in the specified Subsidiary. For example, if your Subsidiary ID is 4, you can use the following query string to retrieve all the items that belongs to this Subsidiary:

  ```
  http://www.mywebstore.com/api/items?region=4
  ```

You can also use these parameters together to further refine the search results. For example, if have more than one SuiteCommerce Website in a NetSuite account and you would like to retrieve items belonging to a particular site, you can use the following query string:

```
http://www.mywebstore.com/api/items?c=3925062&n=3
```

> **Note:** If you specify an incorrect NetSuite account number, an incorrect SuiteCommerce Site ID, an incorrect or inactive Subsidiary ID, or if your website is not reachable, no items are returned and an error message is displayed.

Retrieve Items by Item ID

You can use the parameter `id` to specify the item record ID and retrieve information for a particular item. For example, you can use the following query string to retrieve the item with Item ID 123:

```
http://www.mywebstore.com/api/items?id=123
```

To return multiple items in response data, list multiple item IDs separated by commas. For example, you can use the following query string to retrieve the items with Item ID 123, 456, 789:

```
http://www.mywebstore.com/api/items?id=123,456,789
```

> **Note:** If the item is marked as Inactive or if the Display in Web Store box is not selected on the item record, no items are returned. If you specify an invalid item ID, an error message is displayed. In case of Matrix items, active child items are returned only if the parent item is active and the Display in Web Store box is selected.

Retrieve Items by Item URL

You can use the parameter `url` to specify the value in the URL Component field on the item record. The url parameter returns the Item ID of item that matches the URL component value in the request.

```
http://www.mywebstore.com/api/items?url=Vok-500-Cellular-Phone
```

You can specify other parameters along with the `url` parameter to retrieve other item details. For example, you can use the `fields` parameter to retrieve associated images:

```
http://www.mywebstore.com/api/items?url=Vok-500-Cellular-
Phone&fields=itemimages_detail
```

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

```
http://www.mywebstore.com/api/items?url=Vok-500-Cellular-Phone&fieldset=details
```

> **Note:** If the item is marked as Inactive or if the Display in Web Store box is not selected on the item record, no items are returned. If you specify an invalid item URL, an error message is displayed. In case of Matrix items, active child items are returned only if the parent item is active and the Display in Web Store box is selected.

Retrieve Items by Keyword Search

You can use the parameter `q` to retrieve all the items that match the specified keyword. For example, `q=jeans` searches for that keyword in the fields listed on the Search Fields subtab.

```
http://www.mywebstore.com/api/items?q=jeans
```

> **Note:** Matching items are returned only if the specified keyword exists in the search fields that you selected on the Web Site Setup page. For information on selecting Search Fields, see the help topic Search Fields.

To do a multi-term search, specify two or more terms in the URL separated by whitespace. For example, `q=denim jeans` searches for these keywords in the fields listed on the Search Fields subtab.

```
http://www.mywebstore.com/api/items?q=denim jeans
```

> **Note:** Relevance is the default sort order when you use the q parameter to query the Item Search API. Your preferred sort order specified on the Sort Fields subtab is overridden. For information on selecting and configuring Sort Fields, see Select and Configure Sort Fields.

Retrieve Items by Commerce Categories

- `commercecategoryid` — You can use the parameter `commercecategoryid` to retrieve items that belong the specified Commerce Category ID.

  ```
  http://www.mywebstore.com/api/items?commercecategoryid=12345
  ```

- `commercecategoryurl` — You can use the parameter `commercecategoryurl` to retrieve items that belong the specified Commerce Category URL.

  ```
  http://www.mywebstore.com/api/items?commercecategoryurl=/apparel/partywear/
  shoes
  ```

- `commercecategoryname` — You can use the parameter `commercecategoryname` to retrieve items that belong the specified Commerce Category.

  ```
  http://www.mywebstore.com/api/items?commercecategoryname=Summer Sale
  ```

  This parameter is dependent on the Commerce Category facet field being defined in the Web Site Setup record. For more information, see the help topic Add Commerce Categories to Website Search Index.

> **Note:** For a sample search query and response on Commerce Categories, see the help topic Sample Search Query for Returning Categories.

Retrieve Additional Item Details

If you do not specify any input parameters with the Item Search API base URL, only the Item IDs of the items are returned. You can use the following parameters to retrieve additional item details:

- `fields` — You can use the parameter `fields` to retrieve additional item record fields in the API response. Your request can include a valid field ID, or multiple field IDs separated by commas to return multiple fields.

  ```
  http://www.mywebstore.com/api/items?fields=custitem1,upccode
  ```

- `fieldset` — You can use the parameter `fieldset` to retrieve all the item fields associated with a field set.

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

> **Note:** 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.

Flexibility in Specifying the Field Sets of Related Items

- `matrixchilditems_fieldset` — You can use the parameter `matrixchilditems_fieldset` to specify the field set that overrides the reserved `matrixchilditems` field set. The `matrixchilditems` field set is used by the `matrixchilditems_detail` derived 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`.

  ```
  http://www.mywebstore.com/api/items?
  fieldset=search&matrixchilditems_fieldset=mini
  ```

- `correlateditems_fieldset` — You can use the parameter `correlateditems_fieldset` to specify the field set that overrides the reserved `correlateditems` field set. The `correlateditems` field set is used by the `correlateditems_detail` derived 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`.

  ```
  http://www.mywebstore.com/api/items?
  fieldset=search&correlateditems_fieldset=mini
  ```

- `relateditems_fieldset` — You can use the parameter `relateditems_fieldset` to specify the field set that overrides the reserved `relateditems` field set. The `relateditems` field set is used by the `relateditems_detail` derived 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`.

  ```
  http://www.mywebstore.com/api/items?
  fieldset=search&relateditems_fieldset=mini
  ```

Pagination of Query Results

You can use the parameter `offset` in combination with another parameter `limit` to paginate Item Search API query results. The `offset` parameter specifies the offset from the first result and the `limit` parameter specifies the number of items that can be displayed on each page. For example, you can retrieve query results from the eleventh item to the fifteenth item using the following query string:

```
http://www.mywebstore.com/api/items?offset=10&limit=5
```

You can click next at the bottom of the query result to increment the offset parameter by the value of limit. Note that the value of limit remains unchanged. In the above example, when you click next, the value of offset changes to 15 and the next five items are displayed. Also, you can click prev at the bottom of the query result to decrease the value of the offset parameter by the value of limit. In the above example, when you click prev, the value of offset changes to 5 and the previous five items are displayed.

> **Note:** The default value of the offset parameter is 0 and that of the limit parameter is 50. Therefore, when you do not specify offset and limit, the first 50 items are displayed in the query results.

Facet and Filter Items

- `<facet_field_name>` — You can specify the facet field name and facet values to retrieve the item IDs of the items associated with the specified facets and facet values. For example, you can filter items for the specified facets and facet values:

  ```
  http://www.mywebstore.com/api/items?color=red,blue&gender=women
  ```

  If you are unsure about the facet value or if special characters are included in the facet value you want to specify in the query, you can call the Search API requesting all facets. For example, you can request all facets using the following query string:

  ```
  http://www.mywebstore.com/api/items?pricelevel=5&include=facets
  ```

  You can 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.

- `<numeric_facet_field_ID>.ranges` — You can use the parameter `<numeric_facet_field_ID>.ranges` to define range buckets for numeric facets fields such as price. Use the suffix ranges on the field ID to perform the search. For example, you can retrieve an array of range buckets with the number of items in each bucket using the following query string:

  ```
  http://www.mywebstore.com/api/items?pricelevel5.ranges=1-9.99,10-19.99,20-*.
  ```

  The range buckets in this example are 1 to 9.99, 10 to 19.99, and 20 or more.

  > **Note:** 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.

- `<numeric_facet_field_ID>.from&<numeric_facet_field_ID>.to` — To specify a range of values to search for on item records, you can use the parameters `<numeric_facet_field_ID>.from` and `<numeric_facet_field_ID>to`. Use the suffix from and to on the field ID to perform the search. Both from and to are required, and excluding one results in an error.

  For example, you can retrieve all the items with a price between 10 and 20 (both inclusive) using the following query string:

```
http://www.mywebstore.com/api/items?pricelevel5.from=10&pricelevel5.to=20
```

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

Include or Exclude Item Attributes

- `include` — You can use the parameter `include` to specify an item attribute in the API request and retrieve additional data in the JSON response. For example, the following sample query includes facet fields in the JSON response:

  ```
  http://www.mywebstore.com/api/items?include=facets
  ```

- `facet.exclude` — You can use the parameter `facet.exclude` to exclude fields from faceting in Item Search API calls. Excluding facet fields that are not essential can help speed up the API calls. The `facet.exclude` parameter accepts comma delimited facet field names.

  In the following example, the API call specifies a facet value for filtering (`custitem_categories=Kitchen-Utensils`), while excluding the field from faceting (`facet.exclude=custitem_categories`).

  ```
  http://www.mywebstore.com/api/items?
  include=facets&fieldset=search&language=en&country=US&currency=USD&pricelevel=5&c=1234567&n=2&
  ```

Sort Items

You can use the parameter `sort` to define the sort order of search results on your website. By using the field ID you want to sort by in the API request, you can retrieve 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 sort orders are ascending (`asc`) and descending (`desc`).

For example, you can sort the items by their store display name using the following query string:

```
http://www.mywebstore.com/api/items?sort=storedisplayname:asc
```

Items are sorted by category when the value of `commercecategory` and either `commercecategoryid` or `commercecategoryurl` are specified. 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 Sort Fields for Search Results.

Specify Price Level

You can use the parameter `pricelevel` to specify the price level ID for items. 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`.

```
http://www.mywebstore.com/api/items?pricelevel=4&include=facets
```

> **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, then the value of Alternate Price defaults to the Online Price. If Online Price is also not defined for the item, then the value of Alternate Price defaults to the Base Price.

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. 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. Consequently, the custom price (`pricelevel=4`) is displayed to this customer for all the products in the web store.

> **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.

Specify Currency

Prices are usually displayed in the default currency that you have set up for your website. In an account that uses the Multiple Currencies feature, you can use the parameter currency with the ISO Country Code to specify the currency format in which you want to display prices for the items.

```
http://www.mywebstore.com/api/items?
fields=onlinecustomerprice_formatted,pricelevel5&currency=USD
```

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.

> **Note:** If you specify a currency that is not associated with your website or if the Online box is not checked for that currency on the Web Site Setup record, an error message is returned.

Specify Language and Country

A shopper on your website can use their local language to search and browse products on your website. The translated text that shoppers see in product views is based on the translated text you set up on item records. The JSON response data includes item attributes and field set fields in the languages you have set up in your NetSuite account. You must add the corresponding fields to a field set and also check that the field set is included in the Reference ShopFlow SuiteApp. By default, items are returned with fields containing the default language. You can use the language and country parameter to specify the translated text you set up in NetSuite.

In the following example, the search results are displayed in Spanish:

```
http://www.mywebstore.com/api/items?fieldset=search&country=ES&language=es
```

> **Note:** You must always use the language and country parameters together. The country code that you specify should be in uppercase and the langue code should be in lowercase.

All ISO country and language codes are valid if the language is set up for use on your website. To affect the language displayed in a product view, the site template must be localized. For details on localizing site templates and using multiple languages, see the help topic Localization.

Callback Function

You can use the parameter `callback` to define the JavaScript callback function in order to wrap the JSON response.

For example, you can create a variable name `mycallback` by using the callback parameter as shown in the following sample query:

```
http://www.mywebstore.com/api/items?callback=mycallback
```

Retrieve Debug Information

You can use the parameter `ssdebug` and set its value to `true` to retrieve the debug information in the JSON response.

```
http://www.mywebstore.com/api/items?ssdebug=true
```

Item Search API Output Response

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, Ideal For and Collection. Each facet has two facet values.

```
facets: [
      {
         id: "Ideal For",
         values: [
               {
                  url: "Men",
                  label: "Men"
               },
               {
                  url: "Women",
                  label: "Women"
               }
            ]
         },
         {
{
         id: "Collection",
         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** – These links support pagination and are automatically returned in response data. You can also build the URL for the next set of results using the limit and offset parameters returned by the API. You can click next href in the query results to increment the offset parameter by the value of limit and view the next set of results. You can also click prev href in the query results to decrease the value of the offset parameter by the value of limit. Note that the value of limit remains unchanged.

```
links:    [
      {
            rel: "next",
            href:  "http://www.mywebstore.com/api/items?pricelevel=5&offset=10&limit=5&fieldset
=search&include=facets"
      },
 ],
```

- **corrections** – This property displays the past and present values for facet URL components in the Item Search API request. If you have not entered any aliases in NetSuite for facets or facet values you used in the past, then the corrections property does not return any 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.

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

```
http://www.mywebstore.com/api/items?pricelevel=5&fieldset=search&include=facets&custitem_catego
ry=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. You can troubleshoot errors related to currency mismatch by verifying the details in the locale section. When you are using the Multiple Currencies feature, you should always select the correct default locale on the currency record. For example, select New Zealand (English) as the default locale for New Zealand Dollar.

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

  For more information, see the help topic Localization.

- **volatility** – This refers to the CDN cache settings for your SuiteCommerce Advanced implementation. CDN caching can be set for a shorter period of time or a longer period of time. The length of time that a certain type of content is cached on CDN depends on the value of TTL (Time To Live) in CDN, which could be one of the following:
  - Unique — This asset is not cached.
  - Short — This asset may change frequently, so cache it for five minutes.
  - Medium — This asset may or may not change frequently, so cache it for two hours.
  - Long — This asset is not expected to change frequently, so cache it for seven days.

> **Note:** Changing the CDN cache 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.
- **warnings** – If you have specified a parameter in the query that the API does not recognize, the parameter is ignored and a warning message is displayed in the API response. In the following Item Search API response, a warning message is displayed indicating that the non-existent parameter foo has been ignored.

```
"warnings": {
      "processeduri":"http://www.mywebstore.com/api/items?fieldset=search&foo=&include=facets"
,
      "ignoredparams": [
            "foo"
      ]
}
```

Sample Item Search Queries and Results

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

- 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 Returning Item Quantity

If you use the Multi-Location Inventory feature, you can access item inventory per location using fields specifically designed to return item location details. You can use these fields to retrieve the Item Quantity details:

- Quantity Available (quantityavailable_detail)
- Quantity Backordered (quantitybackordered_detail)
- Quantity Committed (quantitycommitted_detail)
- Quantity On Hand (quantityonhand_detail)
- Quantity On Order (quantityonorder_detail)

By using these fields with the Item Search API, you can retrieve the item inventory data. You can also create a field set that includes these fields, and then use that field set in a query to the Item Search API. For information on creating field sets, see Define Field Sets.

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 item quantity information is written to the JSON response, which includes internal ID of the location and quantity back ordered for each item:

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

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 the help topic 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 listed in the subsequent table.
    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 following table 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

Synonyms

The Synonyms feature lets you define variants of terms in your product catalog. With synonyms, shoppers can use familiar search terms and receive expanded results. A search for a term, also searches for that term's synonyms. Therefore, synonyms eliminate the need for you to try and include all terms and their possible variants in item records.

For example, the shopper searches for **pants**. However, **trousers** is more commonly used in your catalog. For **pants** to yield any results, you could add it as a keyword for every relevant item, but that is a time-consuming task. With Synonyms, you make one entry to define **pants** as a synonym for **trousers**. When visitors search for **pants** the search results returns **pants** and **trousers**.

> **Note:** When you define search synonyms, make sure the terms are included in fields you configured as search fields with a Keyword match type. This is necessary because only indexed search fields are included in the search.
>
> For example, consider you define purple, magenta, violet, and lavender as synonyms so shoppers can find the nearest shade of the item. You can use the defined search synonym only if the product color is specified in the Product Description field, and you configured the Product Description field as a search field with a Keyword match type. Alternatively, you can create a custom Color field to specify the color of the product, and define the custom Color field as a search field.

Types of Search Synonyms

You can define two types of search synonyms in NetSuite:

One-way Synonym — A one-way synonym works in one direction to produce search results.

For example, if candy is a one-way synonym to sweets (`sweets -> candy`), searching for sweets triggers a search for its one-way synonym candy. However, searching for candy does not trigger a search for sweets. If you want the other way around to work too, you need to define sweets and candy as group synonyms.

Group Synonym — A group synonym is a collection of synonyms that produce the same search results for all the terms defined in the group.

For example, if sneakers, trainers, and shoes are defined as group synonyms (`sneakers, trainers, running shoes`), searching for any one of these terms triggers a search for all its synonyms.

Define Search Synonyms in NetSuite

To define a one-way synonym:

1. Go to Setup > SuiteCommerce Advanced > One-way Synonyms.
2. On the One-way Synonyms page, click **New One-way Synonym**.
3. On the One-way Synonym page, enter the search term in the **Searching For** box.

   > **Note:** The term you specify here should be the search term your shopper might search for but does not find relevant results. For example, you can enter pants in the **Searching For** box.

4. In the **Shows These Synonyms In Results** box, enter one or more synonyms separated by a comma.

   > **Note:** The terms you specify here should be the synonymous terms you widely use in your catalog for the same product. For example, you can enter jeans and trousers in the **Shows these synonyms in Results** box.

5. If you use multiple websites, select the site from the list for which this one-way synonym should be applied.
6. If your site uses multiple languages, select the language from the list for which this one-way synonym should be applied.
7. (Optional) If you do not want to enable the one-way synonym, check the **Inactive** box.
8. (Optional) If you want to add more one-way synonyms, click **Save & New** and repeat Steps 3–7.
9. Click **Save**.

> **Note:** When you click Save, your search index automatically rebuilds to reflect these changes. When the search index is rebuilt successfully with these changes, the Search Index Job Status page displays **Synonyms Updated** under the Job Name column.

To define a group synonym:

1. Go to Setup > SuiteCommerce Advanced > Group Synonyms.
2. On the Group Synonyms page, click **New Group Synonym**.
3. On the Group Synonym page, enter a comma-separated list of synonymous terms in the **Synonyms** box.

> **Note:** For example, you can enter handbag, purse, shoulder bag in the **Synonyms** box.

4. If you use multiple websites, select the site from the list for which this group synonym should be applied.
5. If your site uses multiple languages, select the language from the list for which this group synonym should be applied.
6. (Optional) If you do not want to enable the group synonym, check the **Inactive** box.
7. (Optional) If you want to add more group synonyms, click **Save & New** and repeat Steps 3–7.
8. Click **Save**.

After you define search synonyms, you can view and edit the synonyms on the One-way Synonyms/Group Synonyms page. These pages display the defined synonyms and their status. If there are too many entries on the page, you can filter out the synonyms you are looking for by specifying the Search Term and Synonyms, Site, Language, Last Modified Date, and Status in the respective fields.

To view, edit, or delete a defined synonym:

1. Go to Setup > SuiteCommerce Advanced > One-way Synonyms/Group Synonyms.
2. You can view the defined synonyms on the One-way Synonyms/Group Synonyms page.
3. (Optional) To filter the list of defined synonyms:
   1. Expand **Filters**.
   2. Use the following fields to filter the list of defined synonyms:
      - Search Term and Synonyms
      - Site
      - Language
      - Last Modified Date
      - Status
4. (Optional) To edit a synonym definition, click **Edit** next to the synonym record.

   > **Note:** To edit multiple search synonym records, use the Edit toggle button.

5. (Optional) To delete a defined synonym, click **Edit** next to the synonym record and then click **Delete** under the Actions menu.

   > **Note:** To delete multiple search synonym records, use the Edit toggle button and then click **Delete Record** under the New column.

Best Practices for Defining Search Synonyms

The following list contains best practices for defining Search Synonyms:

- Test a few synonyms on Sandbox or a Test Site to see how the addition of search synonyms affects the relevance of your search results.
- To improve performance, define fewer synonyms that return relevant results rather than an exhaustive list of possible synonyms.
- If you use multiple websites or multiple languages, create multiple sets of synonyms to address the relevant customer base. For example, since different terms are used to describe the same item in American and British English, you can define such terms as synonyms for your US and UK websites.