

OpenStore Search Operation

The search operation uses hidden fields to define what the SQL search filter should be. These fields pass a tag XML to the server, which builds the SQL based on the hidden field data.

An example of this can be seen in “ClassicAjax.search.cshtml” and “ClassicAjax.searchadvanced.cshtml”

Example hidden field with search tag:

```
<input id="search3" type="hidden" update="save" value="<tag id='search' action='like'
search='genxml/textbox/searchtext' sqlcol='NB3.ProductRef' sqloperator='or' />" />
```

The field tags are created in order then the SQL needs to be built.

The XML tag can have a number of attributes and some are optional:

action – Action to be done.
search – search field xpath.
sqlfield – sql xpath field. (slow on large DB)
sqlcol – sql column name. (recommended)
searchfrom – range search.
searchto – range search.
sqltype – sql data type.
sqloperator – sql operation. AND, OR, etc.
sqlinject – sql text to inject.

The sqlcol can make use of the NBrightIdx table, which speeds up searches on fields that are included in the table. The table is referenced in the SQL SPROC by using [NB3]. Normal columns in the NBrightBuy table are referenced by using [NB1].

```
SELECT [ItemId]
      , [ProductRef]
      , [ProductName]
      , [Manufacturer]
      , [Summary]
      , [SEOName]
      , [TagWords]
      , [SEOPageTitle]
      , [FromPrice]
      , [Qty]
      , [TypeCode]
      , [Lang]
      , [Visible]
FROM [dbo].[NBrightBuyIdx]
```

The XML tags have an action attribute, which tells the server what SQL statement should be built.

The action attributes can be:

sqloperator

Adds a SQL operator to the statement, (defined in the “sqloperator” attribute AND, OR, etc.)

Example:

```
<input id="search1" type="hidden" update="save" value="<tag id='search' action='sqloperator' sqloperator='and' />">
```

sqlinject

Adds SQL static text to the statement.

Example:

```
<input id="search1" type="hidden" update="save" value="<tag id='search' action='sqlinject' sqlinject='' />">
```

open

Adds an opening bracket.

Example:

```
<input id="search1" type="hidden" update="save" value="<tag id='search' action='open' sqloperator='and' />">
```

close

Adds a closing bracket.

Example:

```
<input id="search7" type="hidden" update="save" value="<tag id='search' action='close' />">
```

equal

Builds a SQL test text for an EQUAL operation. Uses the attributes: sqlfield, sqltype, search, sqlcol.

Example 1:

```
<input id="search3" type="hidden" update="save" value="<tag id='search' action='equal' search='genxml/textbox/searchtext' sqlcol='NB3.ProductName' sqloperator='or' />">
```

Example 2:

```
<input id="search3" type="hidden" update="save" value="<tag id='search' action='equal' search='genxml/textbox/searchtext' sqlfield='genxml/lang/genxml/textbox/txtproductname' sqloperator='or' />">
```

not

Builds a SQL test text for an NOT operation. Uses the attributes: sqlfield, sqltype, search, sqlcol.

Example:

```
<input id="search3" type="hidden" update="save" value="<tag id='search' action='not' search='genxml/textbox/searchtext' sqlcol='NB3.ProductName' sqloperator='or' />">
```

like

Builds a SQL test text for an LIKE operation. Uses the attributes: sqlfield, sqltype, search, sqlcol.

Example:

```
<input id="search3" type="hidden" update="save" value="<tag id='search' action='like' search='genxml/textbox/searchtext' sqlcol='NB3.ProductRef' sqloperator='or' />">
```

range

Builds a SQL test text for an RANGE operation. Uses the attributes: sqlfield, sqltype, searchfrom, searchto, sqlcol.

Example:

```
<input id="search3" type="hidden" update="save" value="<tag id='search' action='range'
sqltype='decimal(10,2)' searchfrom=' genxml/textbox/searchfrom' searchto='
genxml/textbox/searchto' sqlcol='NB3.FromPrice' sqloperator='or' />">
```

cats

Create SQL for a multiple category search. The idea is you use a checkboxlist (@CategorySelectList RazorToken) to display the categories and then any selected from the list are used for the search.

Example:

```
<input id="search3" type="hidden" update="save" value="<tag id='search' action='cats'
search=' genxml/checkboxlist/cats' sqloperator='or' />">
```

cat

Create SQL for a single category search. Add “{*criteriacatid*}” token to the statement. categoryid is sometimes not known or different when criteria is first built. So, we use a token and replace when we know the categoryid have been assigned (selected).

Debugging

An important part of building this search criteria is to know what is being created, so you can adjust the tokens as required.

The Navigation Data (and the SQL created) is kept in a temporary file on the server. The easiest way to find this file is to just look at the dates of the files in the NBStoreTemp folder. But you can look at the cookies on the browser to find the name of the file if you're unsure.

WARNING: Doing this kind of dynamic build of SQL is often difficult and frustrating, so start with the basics and then build onto it.