

Sort Clauses reference

Intro

SortClauses are the *sorting options* for Content and Location Search in "Platform stack", for generic use of API Search see [Search Criteria and Sort Clauses](#)

SortClause consists of two parts just like [Criterion](#) and [FacetBuilder](#):

- The API Value: "*SortClause*"
- Specific handler per SearchEngine: "*SortClausesHandler*"

The *SortClause* represent the value you use in the API, while *SortClauseHandler* deals with the business logic in the background translating the value to something the Search engine can understand.

Implementation and availability of a handler sometimes depends on SearchEngine capabilities/limitations.

Common concepts for all Sort Clauses

For how to use each and every SortClause see list below as it depends on the SortClause Value constructor, but in *general* you should be aware of the following common concept:

- `sortDirection`: The direction to perform the sort, either `Query::SORT_ASC` (*default*) or `Query::SORT_DESC`

List of Sort Clauses

The list below reflects Sort Clauses available in `eZ\Publish\API\Repository\Values\Content\Query\SortClause` namespace (it is also possible to [make custom SortClause](#)):

Arguments starting with "?" are optional.

Only for LocationSearch

Sort Clause	Constructor arguments description
<code>Location\Depth</code> <code>>= 5.3/2014.03</code>	? <code>sortDirection</code>
<code>Location\Id</code> <code>>= 5.3/2014.03</code>	? <code>sortDirection</code>
<code>Location\IsMainLocation</code> <code>>= 5.3/2014.03</code>	? <code>sortDirection</code>
<code>Location\Depth</code> <code>>= 5.3/2014.03</code>	? <code>sortDirection</code>
<code>Location\Priority</code> <code>>= 5.3/2014.03</code>	? <code>sortDirection</code>
<code>Location\Visibility</code> <code>>= 5.3/2014.03</code>	? <code>sortDirection</code>

Only for ContentSearch

Sort Clause	Constructor arguments description
<code>LocationDepth</code> DEPRECATED IN 5.3/2014.03	Use <code>LocationSearch</code> with <code>Location\Depth</code> , since <code>LocationDepth</code> can give unpredictable results when content has several locations.

LocationPath DEPRECATED IN 5.3/2014.03	
LocationPathString DEPRECATED IN 5.3/2014.03	
LocationPriority DEPRECATED IN 5.3/2014.03	Use LocationSearch with Location\Priority, since LocationPriority can give unpredictable results when content has several locations.

Common

Sort Clause	Constructor arguments description
ContentId	?sortDirection
ContentName	?sortDirection
DateModified	?sortDirection
DatePublished	?sortDirection
Field	typeIdentifier as string, fieldIdentifier as string, ?sortDirection, ?languageCode >= 5.3/2014.01 as string
MapLocationDistance >= 5.3/2014.03	typeIdentifier as string, fieldIdentifier as string, latitude as float, longitude as float, ?so rtDirection, ?languageCode as string
SectionIdentifier	?sortDirection
SectionName	?sortDirection