Search Criteria Reference

Criteria are the filters for Content and Location Search, for generic use of API Search see Search Criteria and Sort Clauses .

A Criterion consist of two parts just like SortClause and FacetBuilder:

- The API Value: `Criterion`
- Specific handler per search engine: `Criterion Handler`

`Criterion` represents the value you use in the API, while `CriterionHandler` deals with the business logic in the background translating the value to something the Search engine can understand.

> Implementation and availability of a handler typically depends on search engine capabilities and limitations, currently only Legacy (SQL) Search Engine exists, and for instance its support for FullText and Field Criterion is not optimal and it is advised to avoid heavy use of these until future search engine arrives.

Common concepts for most Criteria

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

- `target`: Exposed if the given Criterion supports targeting a specific sub field, example: FieldDefinition or Meta Data identifier
- `value`: The value(s) to *filter* on, this is typically a `scalar` or `array` of `scalars`.
- `operator`: Exposed on some Criteria
    - All operators can be seen as constants on `eZ\Publish\API\Repository\Values\Content\Query\Criterion\Operator`, at time of writing: `IN`, `EQ`, `GT`, `GTE`, `LT`, `LTE`, `LIKE`, `BETWEEN`, `CONTAINS`
    - Most Criteria do not expose this and select `EQ` **or** `IN` depending on if value is `scalar` **or** `array`
    - `IN` & `BETWEEN` always acts on an `array` of values, while the other operators act on single `scalar` value
- `valueData` : Additional value data, required by some Criteria, MapLocationDistance for instance

> In the Legacy search engine, the field index/sort key column is limited to 255 characters by design.
> Due to this storage limitation, searching content using the eZ Country Field Type or Keyword when there are multiple values selected may not return all the expected results.

List of Criteria

The list below reflects Criteria available in the `eZ\Publish\API\Repository\Values\Content\Query\Criterion` namespace (it is also possible to make a custom Criterion):

Only for LocationSearch

Criterion	Constructor arguments description
`Location\Depth`	`operator` (`IN, EQ, GT, GTE, LT, LTE, BETWEEN`), `value` being the Location depth(s) as an integer(s).
`Location\IsMainLocation`	Whether or not the Location is a Main Location `value` (`Location\IsMainLocation::MAIN, Location\IsMainLocation::NOT_MAIN`).

`Location\Priority`	Priorities are integers that can be used for sorting in ascending or descending order. What is higher or lower priority in relation to the priority number is left to the user choice. `operator` (GT, GTE, LT, LTE, BETWEEN), `value` being the location priority(s) as an integer(s).

Common

Criterion	Constructor arguments description
`ContentId`	`value` being scalar(s) representing the Content id.
`ContentTypeGroupId`	`value` being scalar(s) representing the Content Typ eGroup id.
`ContentTypeId`	`value` being scalar(s) representing the Content Type id.
`ContentTypeIdentifier`	`value` being string(s) representing the Content Type identifier, example: "article".
`DateMetadata`	`target` (`DateMetadata`::MODIFIED , `DateMetadata`::CREATED), `operator` (IN, EQ, GT, GTE, LT, LTE, BETWEEN), `value` being integer(s) representing unix timestamp.
`Field`	`target` (FieldDefinition identifier), `operator` (IN, EQ, GT, GTE, LT, LTE, LIKE, BETWEEN, CONTAINS), `value` being scalar(s) relevant for the field.
`FieldRelation`	`target` (FieldDefinition identifier), `operator` (IN , CONTAINS), `value` being array of scalars representing Content id of relation. *Use of `IN` means relation needs to have one of the provided ID's, while `CONTAINS` implies it needs to have all provided id's.*
`FullText`	`value` being the string to search for, `properties` is array to set additional properties for use with future search engines like Solr/ElasticSearch.
`LanguageCode`	`value` being string(s) representing Language Code(s) on the Content (*not* on Fields), `match AlwaysAvailable` as boolean.
`LocationId`	`value` being scalar(s) representing the Location id.
`LocationRemoteId`	`value` being string(s) representing the Location Remote id.
`LogicalAnd`	A `LogicalOperator` that takes `array` of other Criteria, makes sure all Criteria match.
`LogicalNot`	A `LogicalOperator` that takes `array` of other Criteria, makes sure none of the Criteria match.

LogicalOr	A `LogicalOperator` that takes `array` of other Criteria, makes sure one of the Criteria match.
MapLocationDistance	`target` (FieldDefinition identifier), `operator` (`IN, EQ, GT, GTE, LT, LTE, BETWEEN`), `distance` as float(s) from a position using `latitude` as float , `longitude` as float as arguments
MatchAll	*No arguments, mainly for internal use when no* `filter` *or* `query` *is provided on* `Query` *object .*
MatchNone	*No arguments, mainly for internal use by* Block ingLimitation*.*
ObjectStateId	`value` being string(s) representing the Content ObjectState id.
ParentLocationId	`value` being scalar(s) representing the Parent's Location id
RemoteId	`value` being string(s) representing the Content Remote id.
SectionId	`value` being scalar(s) representing the Content Section id.
Subtree	`value` being string(s) representing the Location id in which you can filter. *If the Location Id is /1/2/20/42, you will filter everything under 42.*
UserMetadata	`target` (`UserMetadata ::OWNER , UserMetadata ::GROUP , UserMetadata :: MODIFIER`), `operator` (`IN, EQ `), `value` being scalars (s) representing the User or User Group id(s).
Visibility	`value` (`Visibility ::VISIBLE , Visibility ::HIDDEN`), *Note: This acts on all assigned locations when used with ContentSearch, meaning hidden content will be returned if it has a location which is visible. Use LocationSearch to avoid this.*