PluginList filter control block

A filter is used to define the search query that will be run to gather a set of results for the list. A large number of possible individual parameters can be 'filtered' as set out in the table below and multiple filter control blocks can be applied in an additive manner ie the equivalent of an AND operation.

**Available filters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Boolean</th>
<th>Sample</th>
<th>Tiki Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>categories</td>
<td>Match on the object's directly associated categories</td>
<td>yes</td>
<td>1 OR 3</td>
<td>7</td>
</tr>
<tr>
<td>content</td>
<td>Full text search on all of the object's content</td>
<td>yes</td>
<td>hello world</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Additional field argument can be provided to search for other fields</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>instead of the default ones.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>contributors</td>
<td>user login to search for as the contributors for the content</td>
<td>yes</td>
<td>admin OR testuser</td>
<td>8</td>
</tr>
<tr>
<td>deepcategories</td>
<td>Match on the object's categories or their parents, providing deep searching</td>
<td>yes</td>
<td>1 OR 3</td>
<td>8</td>
</tr>
<tr>
<td>distance</td>
<td>filtering on object location by distance (Seems to work with Elasticsearch</td>
<td>no</td>
<td>{filter distance=&quot;10km&quot; lat=&quot;51.5287336&quot; lon=&quot;-0.3824611&quot;}</td>
<td>16.1</td>
</tr>
<tr>
<td></td>
<td>only - Tiki19.1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>exact</td>
<td>Exact text search</td>
<td>yes</td>
<td>hello world</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Use with the field argument to search for an exact title or field value.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can be used to look for empty tracker field</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>{filter field=&quot;tracker_field_foobar_text&quot; exact=&quot;&quot;}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>favorite</td>
<td>Show only the favorited objects of a certain user. The feature Wiki</td>
<td>no</td>
<td>{filter favorite=&quot;{{user}}&quot;}</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Argument Variables must be set at : Admin -&gt; Editing and Plugins -&gt; Wiki</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>argument variables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Boolean</td>
<td>Sample</td>
<td>Tiki Version</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>field</td>
<td>Basic filter by field contents For the full list of fields see Unified+Index#Fields</td>
<td>yes</td>
<td><code>{filter field=&quot;title&quot; content=&quot;foo&quot;},</code> <code>{filter field=&quot;tracker_id&quot; content=&quot;42&quot;},</code> <code>{filter field=&quot;tracker_field_foobar&quot; content=&quot;John Doe&quot;},</code> <code>{filter field=&quot;tracker_status&quot; content=&quot;o OR p&quot;},</code> <code>{filter field=&quot;blog_excerpt&quot; content=&quot;Some text&quot;}</code></td>
<td>7 (and onwards with more fields arriving most releases)</td>
</tr>
<tr>
<td>language</td>
<td>Search on the object's language (2-letter code) when available. unknown is used when unspecified.</td>
<td>yes*</td>
<td>en or fr OR unknown</td>
<td>7</td>
</tr>
<tr>
<td>multivalue</td>
<td>needed when filtering on array type fields like users' groups</td>
<td>no</td>
<td>Customers OR Staff</td>
<td>?</td>
</tr>
<tr>
<td>personalize</td>
<td>to filter down to own contributions without relying on argument variables</td>
<td>?</td>
<td>?</td>
<td>12</td>
</tr>
<tr>
<td>range</td>
<td>A date (unix timestamp) range Must have from and to arguments specified</td>
<td>no</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>range</td>
<td>In Tiki 9, from and to can now take now instead of a unix timestamp, and you can also set either from or to together with &quot;gap&quot; to specify the one relatively to the other</td>
<td>no</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>relation</td>
<td>Must have certain relation Additional field arguments objecttype and qualifier (the relationship) must be specified</td>
<td>no</td>
<td>ject_id &gt;</td>
<td>8</td>
</tr>
<tr>
<td>texrange</td>
<td>A text range Must have from and to arguments specified</td>
<td>no</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>type</td>
<td>Filter on object type</td>
<td>no</td>
<td><code>wiki page</code> (from Tiki 8 you can search for multiple types by providing a comma separated list) <code>trackeritem</code> to search trackers</td>
<td>7/8/9/11 or earlier</td>
</tr>
</tbody>
</table>

* AND matching won't make much sense, single-value field
Boolean Syntax

When applying an individual filter more complex logic can also be applied using a Boolean syntax - that applies to all aspects of Unified Search - e.g. when filtering on content you could exclude all items not containing the word "water" by using:

```
{filter content="NOT water"}
```

Empty or Non-empty

To filter for a empty or non-empty fields, the syntax might depend on the Unified Index engine.

- When using "MySQL Full Text Search" or "Elasticsearch" as the Unified Search Engine, you can use these syntaxes:
  - to include all non-blanks non-NULL values:
    ```
    {filter content="NOT "}
    ```
  - to include all blanks OR NULL values:
    ```
    {filter exact=""}
    ```
  - Note: it's not possible at the moment to perform an OR or AND filter operation where one of the elements is blank and the other is not, e.g.: " OR 1".

Peculiarities for Elasticsearch

When using Elasticsearch as the Unified Search Index, you need to be aware of a few peculiarities:

1. To filter by exact value use the tracker field perm name with the tracker_field_ prefix
2. To filter by exact EMPTY value (i.e. missing value), use the _text version of the permanent name for that field where available.
   - Mysql index works with non _text version too but elastic does not. The reason is that Radio button, dropdown and a number of other fields use the normal permName field as identifier and add a _text version which is a plain text.
   - In Elasticsearch, this means that identifiers are exact-matches - they are not analyzed by search index and wildcard searches (like the empty one) does not work on them. Plain text, on the other hand, support wildcard searches, are analyzed and thus, work for exact="" search.
3. When you use a field (let's say radio button, or dropdown, for instance) with a value "No", you will get some problems when search for that value in Elasticsearch since "No" is a stopword in Elasticsearch.
   - This is defined in the "Control Panel > Search" where you can exclude it to not be a stopword.
   - Being a stopword means that for analyzed string fields like tracker_field_myPermanentName_text, searching for stopword will result in nothing and searching for an empty value will show those fields containing a stopword. So, instead of seeing all empty ones, you will see all empty ones and all the ones that have a stopword in them - like all the "No" ones. This is undesired, so if you have such a field and need to filter on a stopword, please update the list of stopwords to exclude that specific value.
Examples

1.1.1. Basic

Basic usage

{LIST()}

{pagination max="10"}
{filter content="hello world"}
{filter categories="1 AND 2"}
{filter language="en or unknown"}
{filter type="wiki page"}
{filter range="modification_date" from="2014-01-01" to="last week"}
{LIST}

The date filters use strftime to convert various common formats to Unix epoch. Unlike what the documentation previously said, this has always been the case.

1.1.2. Date ranges

Parameters to, from and gap:

- The idea is that you specify the value in to as the last day you want that filter to work on.
  - For instance: 1 week ago or "-1 week" - both should work with php's strtotime() parsing.
- Then, the gap should cover the whole previous date range that you need. If you want, for instance, items older than 1 week, you can use a number there as big as you can, for instance 40 years (of course not going beyond 1970 as this is the start of the UNIX era and strtotime will return odd results).
- And the idea of parameter from is that you specify the first day you want that filter to work on.

Note that the upper boundary of the range is not included in the search

In Tiki 15, the date filters are based on seconds and the following are possible:

Tiki 15+

{filter range="tracker_field_myDate" gap="86400" to="now"} {* Within the last day *}

From Tiki 16+, the date filter has been "humanised" and the following date filters are possible:

Tiki 15+

{filter range="tracker_field_myDate" gap="1 day" to="now"} {* Within the last day *}

{filter range="tracker_field_myDate" gap="10 years" to="-1 week"} {* Older than one week *}

{filter range="tracker_field_myDate" gap="1 week" to_field="tracker_field_myDeadline"} {* 1 week before the deadline *}

{filter range="modification_date" from="-10 years" gap="1 year"} {* Between 9 and 10 years ago *}

{filter range="modification_date" gap="1 year" to="10 years"} {* In between 9 and 10 years from now *}

{filter range="modification_date" gap="1 year" from="now"}
You can also filter using human value like this month, next month, etc.

1.1.2.1. Unix timestamps

Unix timestamps may also be used. In Tiki 9+, the following date filters are possible:

```plaintext
{filter range="modification_date" from="1317427200" to="now"}
{filter range="modification_date" gap="31536000" to="now"}
{filter range="modification_date" from="now" gap="31536000"}
```

1.1.3. Numeric ranges

Note that the upper boundary of the range is not included in the search

When you use Elasticsearch as the unified search index, you can reuse the same parameters to, from and gap that are shown above for Date ranges:

```plaintext
{filter range="tracker_field_myPermanentName" from="40" gap="100"}
```

This will display items which have the field with permanent name "myPermanentName" with values starting at 40 and up to 100 numbers higher (from 40 to 140).

1.1.4. Tracker with categories

Filter in a specific tracker

```plaintext
{LIST()}
{list max="10"}
{filter content="hello world"}
{filter categories="1 AND 2"}
{filter content=5 field=tracker_id}
{LIST}
```

1.1.5. Tracker and logged user items

Filter in a specific tracker and the logged user item(s) - single user selection

```plaintext
{LIST()}
{list max="1"}
{filter content="2" field="tracker_id"}
```
1.1.6. Tracker item status

Filter by tracker item status

{LIST()
  {filter content="2" field="tracker_id"}
  {filter field="tracker_status" content="o OR p"}
}{LIST}

1.1.7. Tracker items with empty field

Filter tracker items having an empty field (exact match)

{LIST()
  {filter content="2" field="tracker_id"}
  {filter field="tracker_field_fieldPermanentName" exact=""}
}{LIST}

1.1.8. Wiki pages with partial match

Filter wiki pages with a partial match with some text string such as ‘foo’

{LIST()
  {filter type="wiki page"}
  {filter field="title" content="foo"}
}{LIST}

1.1.9. Wiki pages with partial match and spaces in string

Filter wiki pages with a partial match with some text string containing spaces such as ‘foo bar’

{LIST()
  {filter type="wiki page"}
  {filter field="title" content="\"foo bar\""}
}{LIST}

1.1.10. Wiki pages with exact match and spaces in string

Filter wiki pages with an exact match with some text string containing spaces such as ‘foo bar’

{LIST()
  {filter type="wiki page"}
  {filter field="title" exact="foo bar"}
}{LIST}

1.1.11. Wiki pages with boolean partial matches and spaces
in strings

Filter on literal text with spaces in

{LIST()}
  {filter content=""hello world\" OR "goodbye world\""}
{LIST}

1.1.12. Users in specific group(s)

Filter users in a specific group

{LIST()}
  {filter field="object_type" content="user"}
  {filter field="groups" multivalue="Customers OR Staff"}
{LIST}

1.1.13. Related items

In this example we want to find all objects related to trackeritem itemId 42 related using the qualifier thing.related.content. Note we need to add .invert to the end of it because they are all related via a Relations field on this tracker, so the relation from their point of view is "inverted"

Related objects

{LIST()}
  {filter relation="42" objecttype="trackeritem" qualifier="thing.related.content.invert"}
{LIST}

Objects related to a wiki page

{LIST()}
  {filter relation="page name" objecttype="wiki page" qualifier="thing.related.content"}
{LIST}

All the LIST Plugin control blocks

- PluginList pagination or list control block
- PluginList filter control block
- PluginList output control block
- PluginList format control block
- PluginList display control block
- PluginList sort control block
- PluginList advanced output control block
- PluginList aggregate control block
- PluginList overview about control blocks parameters and values
- LIST - Troubleshooting The List Plugin
- LIST - Graphical User Interface
- LIST - Hacks and Fun

Alias names of this page:
listfilter | listfilters | list filter | list filters | listfilterrange | filterrange