

Plugin Tracker Query Template

Introduced in [Tiki10](#)

Use this [wiki plugin](#) to generate forms from a tracker. Currently only able to list data, the TrackerQueryTemplate plugin simply obtains data from a tracker and allows an editor the ability to list the tracker data as he sees fit.

Parameters

Generate a form from tracker data

Introduced in Tiki 10. Required parameters are in **bold**.

[Go to the source code](#)

Preferences required: feature_trackers, wikiplugin_trackerquerytemplate

Parameters	Accepted Values	Description	Default	Since
(body of plugin)		Wiki Syntax, with variables from tracker query.		
tracker	text	The name of the tracker to be queried, or if byname="n", the tracker ID.		10.0
andfilters	text	Apply "and" filters to fields. Format: field:value;field:value;field:value , where field may be the field name or ID depending on the setting for the byname parameter.		10.0
likefilters	text	Apply "like" filters to fields. Format: field:value;field:value;field:value , where field may be the field name or ID depending on the setting for the byname parameter.		10.0
orfilters	text	Apply "or" filters to fields. Format: field:value;field:value;field:value , where field may be the field name or ID depending on the setting for the byname parameter.		10.0
debug	(blank) y n	Turn tracker query debug on (off by default).		10.0
getlast	(blank) y n	Retrieve only the last item from the tracker.		10.0
render	(blank) y n	Render as needed for trackers (default).	y	10.0
byname	(blank) y n	Use the tracker name instead of tracker ID in the tracker parameter. Also use the field name instead of field ID in the filter parameters. Set to Yes (y) to use names (default) or No (n) to use IDs.	y	10.0
itemid	digits	Item id of tracker item		10.0

itemids	digits	Item id of tracker items, separated with comma	11.0
	separator: ,		

Features

- Supports nesting, where values in settings are from the parent tracker, and those in the body are from the current tracker
- When using byname="y" (default), variables are accessed `$field name$` (rendered) and `$~field name$` (unrendered)
- When byname="n" , variables are `{ $f_id }` (rendered) and `{ $~f_id }` (unrendered)
- Filters are in the format `field:value;field:value;field:value;`

Simple Example

Here the:

- tracker id = 1
- username field id = 1
- email field id = 2

```
{TRACKERQUERYTEMPLATE(tracker="1" orfilters="1:sam;1:bob;")} Username: { $f_1 } Email: { $f_2 }
{TRACKERQUERYTEMPLATE}
```

Nested Example

Here we use the byname parameter, which can be more intuitive. **NOTICE** how `$~username$` comes from tracker named "User Tracker" and not "User Sports Tracker", thus aiding in complex joining scenarios.

```
{TRACKERQUERYTEMPLATE(tracker="User Tracker" byname="y" orfilters="username:sam;username:bob;")} Username:
$username$ Email: $email$ Sports: {TRACKERQUERYTEMPLATE(tracker="User Sports Tracker" byname="y"
orfilters="username:$~username$")} *$sport$ {TRACKERQUERYTEMPLATE} {TRACKERQUERYTEMPLATE}
```

What is nesting?

Nesting is when an object is embedded in the same type of object, a visual:

- root
 - nested in root
 - nested in previous nested

or in the case of TrackerQueryTemplate nested 2 times:

```
{TRACKERQUERYTEMPLATE()} {TRACKERQUERYTEMPLATE()}
{TRACKERQUERYTEMPLATE()} {TRACKERQUERYTEMPLATE} {TRACKERQUERYTEMPLATE}
{TRACKERQUERYTEMPLATE}
```