List Customization

The following sections form a sort of cook book with recipes for customizing result lists in Xataface.

Adding Option to Filter Result LIst

Problem

You want to add a drop-down list for the user to select filters for the result set.

Solution

Use the `filter` directive in the fields.ini file on any field that you want the filter to be added for.

Field definition in fields.ini file using the filter directive.

```
[test_field]
  filter=1
```

Now, if you navigate to this table's list view, you'll see a drop-down list at the top of the results where you can select all of the distinct values in the `test_field` column.

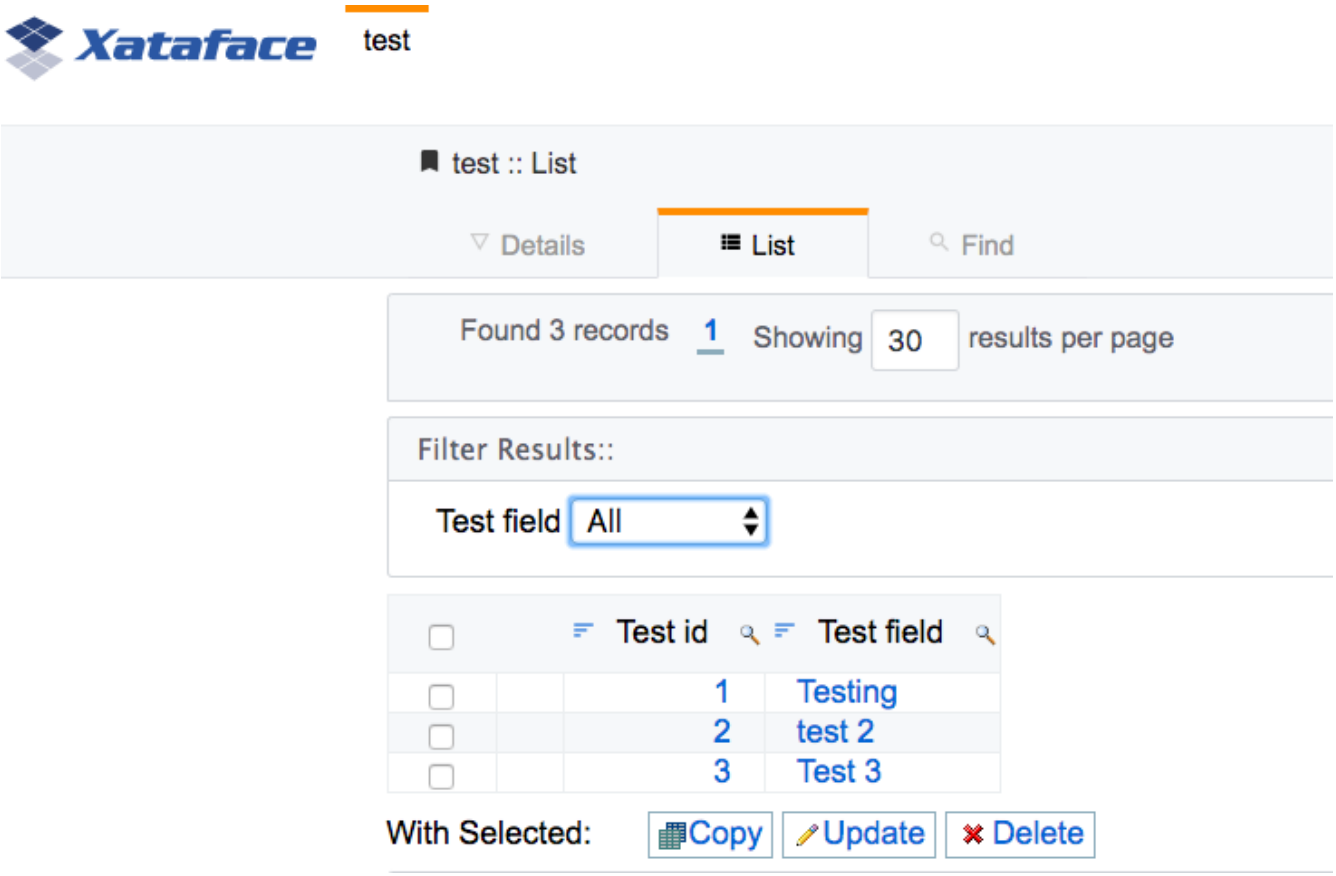


Figure 1. List View when a filter field defined.

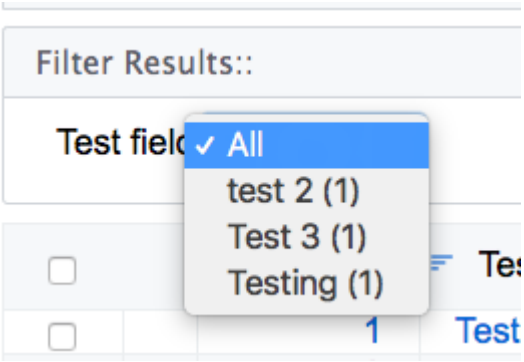


Figure 2. Expanding the filter list, you can see all of the distinct values for the test_field column.

If you select one of the options, it will filter the results to only show those results that match the filter.

Hiding Filter Counts

Problem

You want to hide the counts for each entry in the "filters" drop-down list on the list view. By default when you add a filter list via the `filter` directive, each row has the "count" displayed beside it:

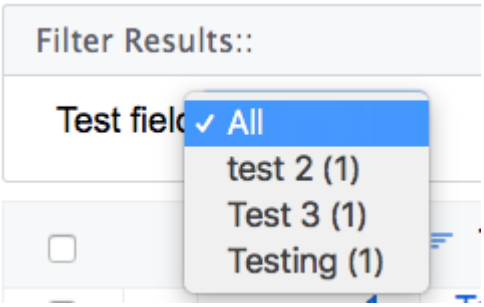


Figure 3. A result filter list. Notice each row has a (1) beside it indicating that there is 1 row matching that filter.

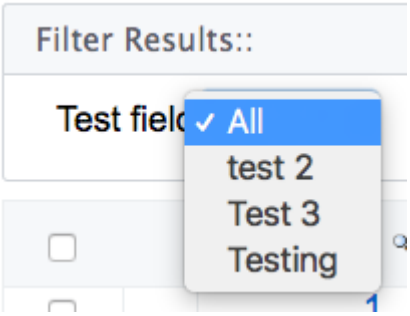
We want to hide this.

Solution

Use the `show_filter_counts` preferences directive to hide these counts as follows. In the `_prefs` section of the `conf.ini` file, add:

```
show_filter_counts=0
```

After this change, the filter lists will look like:



Sorting Filter Results

Problem

Filter lists are sorted by the "filtered" field in alphabetical order. You want to sort the filter lists on some other column.

Solution

Use the `filter.sort.` fields.ini directive to specify the field that the filter should sort on.

E.g.

Sort the "city" filter in descending order

```
[city]
  filter=1
  filter.sort="city desc"
```

Adding CSS Classes to Rows

Problem

You want to add custom CSS classes to the rows of the table in list view.

Solution

Implement the `css__tableRowClass()` method in the table delegate class. Have it return a string with the CSS classes you wish to add.

```php
<?php
class tables_MyTable {
    ...

    function css_tableRowClass(Dataface_Record $rec) {
        if ($record->val('status') == 'approved') {
            return 'status-approved another-css-class';
        } else {
            return 'another-css-class status-pending';
        }
    }
}
```

In the above example, on rows where the 'status' field is a approved you'll see something like:

```html
<tr class="status-approved another-css-class">
```

and in other rows you'll see

```html
<tr class="another-css-class status-pending">
```

Adding Row Actions

Problem

You want to add a button to each row of the list view to perform some action on that row.

Solution

Define an action with `category=list_row_actions` and `condition="$query['-table'] == 'tablename'`.

E.g.

Defining an action named "play_post" that is displayed in each row of the list view for the `_tmp_newsfeed` table.

```
[play_post]
    condition="$query['-table'] == '_tmp_newsfeed'" ①
    category=list_row_actions
    materialIcon=play_circle_outline ②
    label="Play"
    description="Play narration"
    url="{$record->getURL('-action=play_post')}" ③
    url_condition="$record" ④
    order=1
```

① Only display this action in the `_tmp_newsfeed` table.

② Uses a material icon for the action.

③ The `url` directive specifies the URL where the action should go when the user clicks on it. THe `$record` variable is a `Dataface_Record` object that encapsulates the current row. We call the `getURL()` method to get the URL for that record with the `play_post` action.

④ The `url_condition` directive is necessary to stop Xataface from trying to parse the `url` directive if `$record` is `null`. It is interpretted as a boolean expression. When it evaluates to a falsey value, it will skip parsing the `url` directive.

IMPORTANT	When the user clicks on this action, they will be directed to the URL `index.php?-table=_tmp_newsfeed&-action=play_post&…`. You need to make sure to implement this action handler in `actions/play_post.php`.

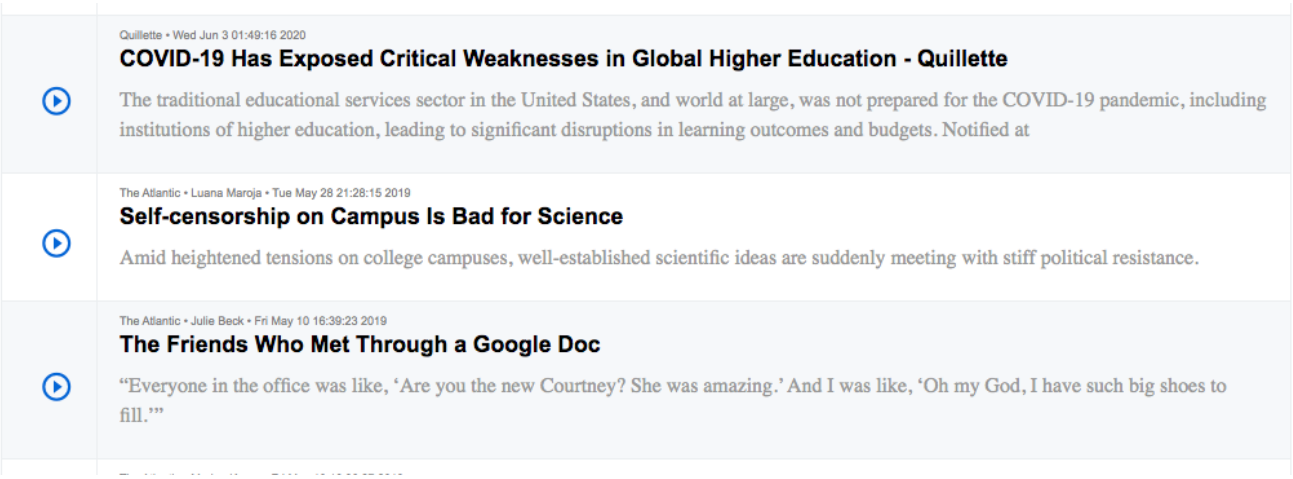


Figure 4. The "play_post" action appears in the left-most column of the list view

Customizing Row Action Styles

Problem

You want to customize the style on a particular action

Solution

Use the `class` directive on the action to specify a custom CSS class on the `<a>` tag of the action. Then use this CSS class to target that button specifically from your stylesheet.

```
[play_post]
    condition="$query['-table'] == '_tmp_newsfeed'"
    category=list_row_actions
    materialIcon=play_circle_outline
    label="Play"
    description="Play narration"
    order=1
    class=large-button
```

Then in your CSS file you can target this action directly:

```
.large-button {
    font-size: 150%;
}
```

Adding Javascript Row Actions

Problem

You want to trigger a Javascript function when the user clicks on a row action instead of just directing the user to a URL.

Solution

Use the `onclick` directive instead of the `url` directive.

See [javascript-action] for an introduction to Javascript actions with a detailed example using the `onclick` handler. The only thing we need to add to make our Javascript action useful, is the ability to retrieve the record ID of the current row. There are two ways to do this:

1. Use the `$record` variable inside the `onclick` directive to obtain details about the record, and add them as parameters to your Javascript function.

e.g.

```
[myaction]
    category=list_row_actions
    onclick="window.playPost('{$record->val('post_id')}');"
```

An alternative way is to make user of the `xf-record-id` attribute that can be found on the `<tr>` tag of the row in list view. If you look at the resulting HTML source of the list view, and drill down to the individual rows of the table, you'll see something like:

```
<tr class="listing odd " xf-record-id="_tmp_newsfeed?post_id=73">
...
```

We can use this inside our Javascript function, as follows. First we pass `this` as an argument to our function. `this` will refer to the `<a>` tag that was clicked.

```
[myaction]
    category=list_row_actions
    onclick="window.playPost(this);"
```

Javascript file definine the function that we want to call from our `playPost` function

```
(function(){
    var $ = jQuery;
    window.playPost = playPost;

    function playPost(el) {
        if (!$(el).attr('xf-record-id')) {  ①
            var trTag = $(el).parents('[xf-record-id]').first();
            if (trTag.length == 0) {
                return new Promise(function(resolve, reject){ ②
                    reject('Not found');
                });
            } else {
                el = trTag; ③
            }
        }
        return new Promise(function(resolve, reject) { ④
            // Do the actual playing here, and either call resolve() or reject()
            // when done.
        });

    }
})();
```

① Check to see if the HTML element that element contains the `xf-record-id` attribute. If it doesn't we need to walk up the DOM until with find an element that does.

② If we didn't find **any** elements with the xf-record-id attribute, we'll just return a promise that rejects.

③ If we found an element with `xf-record-id` we will just use this element instead of the one that was passed into the method. Since our action is called with `window.playPost(this)`, it will always be passing the `<a>` tag to the `playPost()` method, and the `a` tag doesn't have the attribute. The parent `<tr>` tag has the attribute, so this is where we crawl up to.

④ We perform our action on the provided element. In this case, I'm returning a Promise to get us prepared for performing asynchronous actions cleanly.

Making Row Actions Toggleable

Problem

You want to add an action to each row of the list view that can be toggled between two different states. For example, we have functionality to add and remove rows from a playlist. If the record is currently "on" the playlist, we want the action to display "Remove from playlist". If the record is not on the playlist, we want the action to display "Add to playlist".

Solution

Use two different actions: "add_to_playlist" and "remove_from_playlist" and conditionally show

either action depending on whether the record is currently "on" the playlist.

I'll include two different recipes here to achieve this:

1. A fully server-side solution using the `condition` directive.
2. A server **and** client-side solution using Javascript, CSS, and AJAX to add and remove rows from the playlist.

The 2nd option is more complex but yields a better user experience.

Solution 1: Using `condition` directive

We can define our actions as follows: (And assume that our table has an "on_playlist" field that indicates whether or not a record is on the playlist currently.

```
[add_to_playlist]
    condition="$query['-table'] == '_tmp_newsfeed' and $record and !$record-
>val('on_playlist')"
    category=list_row_actions
    label="Add to Playlist"
    materialIcon="playlist_add"
    order=2
    onclick="window.addToPlaylist(this)"


[remove_from_playlist]
    condition="$query['-table'] == '_tmp_newsfeed' and $record and $record-
>val('on_playlist')"
    category=list_row_actions
    materialIcon="remove_from_queue"
    order=3
    onclick="window.removeFromPlaylist(this)"
    label="Remove from Playlist"
```

The key here is in the `condition` directives of these actions. The `add_to_playlist` is set to appear only when we are on the `_tmp_newsfeed` table **AND** the record is not on the playlist. The `remove_from_playlist` action is set to appear only when the record is on the playlist.

IMPORTANT

In both actions we need to ensure that `$record` exists before calling `$record->val('on_playlist')` otherwise the application will crash in cases where there is NO record in the current context. I.e. We need to have

```
$record and $record->val('on_playlist')
```

and not just

```
$record->val('on_playlist')
```

There is a lot hidden in this solution inside the `addToPlaylist()` and `removeFromPlaylist()` Javascript functions. These are responsible for actually adding and removing records from the playlist. See Using AJAX To Modify Row Records for an example using AJAX to do this.

Solution 2: Using CSS to Show/Hide Actions

The first solution relies on the `condition` directive to show or hide our actions. However, this directive is processed on the server-side, so we would need to reload the whole page if we wanted to update state. We can offer a better user experience by using CSS to show/hide the actions.

The gist of this solution is to:

1. Add a CSS class, `on-playlist`, to the `<tr>` tag (i.e. each record row) to indicate whether that record is currently on the playlist.
2. Add CSS classes to the two actions, so that we can easily target them from a stylesheet.
3. Add custom CSS to show/hide actions depending on whether the `<tr>` tag includes the `on-playlist` CSS class.
4. The `addToPlaylist()` function removes the `on-playlist` CSS class from the `<tr>` tag, and the `removeFromPlaylist()` adds it.

Adding the CSS class to the `<tr>` tag:

TIP See Adding CSS Classes to Rows for more details on adding CSS classes to rows.

Method from the table delegate class that causes the `<tr>` tag to have the `on-playlist` CSS class if the record is on the playlist.

```
function css__tableRowClass(Dataface_Record $rec = null) {
    if ($rec->val('on_playlist')) {
        return 'on-playlist';
    }
    return '';
}
```

Adding CSS classes to the two actions:

TIP See [custom-row-actions-styles] for more details on using the `class` directive to customize action styles.

```
add_to_playlist]
    condition="$query['-table'] == '_tmp_newsfeed'"
    category=list_row_actions
    label="Add to Playlist"
    materialIcon="playlist_add"
    order=2
    onclick="window.addToPlaylist(this)"
    class="add-to-playlist" ①

[remove_from_playlist]
    condition="$query['-table'] == '_tmp_newsfeed'"
    category=list_row_actions
    materialIcon="remove_from_queue"
    order=3
    onclick="window.removeFromPlaylist(this)"
    label="Remove from Playlist"
    class="remove-from-playlist" ②
```

① We add the `add-to-playlist` CSS class to the `<a>` tag for the "add" action.

② We add the `remove-from-playlist` CSS class to the `<a>` tag for the "remove" action.

Defining Styles to Show/Hide Actions

Now that we have our CSS actions defined, we'll be dealing with HTML like:

```
<tr class="on-playlist">
  ....
  <a class="remove-from-playlist" ...> ... </a>

  <a class="add-to-playlist"...> ...</a>
```

Now we can target our `<a>` tags with the following CSS directives in our stylesheet.

```
/* Hide add-add-to-playlist when tr has on-playlist class */
tr.on-playlist .add-to-playlist {
    display:none;
}

/* Hide remove-from-playlist by default */
span.row-actions .remove-from-playlist {
    display:none;
}

/* Show remove-from-playlist when tr has on-playlist class */
tr.on-playlist .remove-from-playlist {
    display:block;
}
```

Adding/Removing CSS classes using Javascript

As we've seen above, our actions will trigger the `addToPlaylist()` and `removeFromPlaylist()` functions respectively. We just need add/remove the CSS class as appropriate here.

e.g.

```
function addtoPlaylist(el) {
    if (!$(el).attr('xf-record-id')) {
        var trTag = $(el).parents('[xf-record-id]').first();
        if (trTag.length == 0) {
            return new Promise(function(resolve, reject){
                reject('Not found');
            });
        } else {
            el = trTag;
        }
    }
    $(el).addClass('on-playlist'); ①
    return new Promise(function(resolve, reject) {
        // Do the actual playing here, and either call resolve() or reject()
        // when done.

        addToPlaylistImpl(el.attr('xf-record-id')).then(function(data) {
            // success
            resolve(data);
        }).catch (function(data) {
            // Failed
            $(el).removeClass('on-playlist'); ②
            reject(data);
        });;
    });
}
```

```
function removeFromPlaylist(el) {
    if (!$(el).attr('xf-record-id')) {
        var trTag = $(el).parents('[xf-record-id]').first();
        if (trTag.length == 0) {
            return new Promise(function(resolve, reject){
                reject('Not found');
            });
        } else {
            el = trTag;
        }
    }
    $(el).removeClass('on-playlist');③
    return new Promise(function(resolve, reject) {
        // Do the actual playing here, and either call resolve() or reject()
        // when done.

        removeFromPlaylistImpl(el.attr('xf-record-id')).then(function(data) {
            // success
            resolve(data);
        }).catch (function(data) {
            // Failed
            $(el).addClass('on-playlist'); ④
            reject(data);
        });;
    });
}
```

① Optimistically add the 'on-playlist' CSS class when user clicks the "add" button. This will toggle the actions immediately.

② If the action failed, we revert the CSS class back, by removing the "on-playlist" CSS class.

③ Optimistally remove the 'on-playlist' CSS class when user clicks the "remove" button. This will toggle the actions immediately.

④ If the action failed, we revert the CSS class back, by re-adding the "on-playlist" CSS class.

Using AJAX To Modify Row Records

Problem

You need to perform a server-side action when the user clicks on a row action. E.g. From Making Row Actions Toggleable, we needed to add and remove the row record from the playlist.

Solution

Create a custom action handler that performs the the function, and returns JSON, and write a Javascript function that calls this AJAX action.

TODO: Need to write this solution

Specifying Default Sorting

Problem

You want the records in a particular table to be sorted on a particular column by default.

Solution

Use the `table.default_sort` property of the fields.ini file.

Using the `table.default_sort` property in the fields.ini file to sort records in descending order on the date_posted field.

```
table.default_sort=date_posted desc
```

TIP See [default-related-sort] to learn how to set the default sort order on a related tab.

Rendering List View as a Grid

Problem

You want to display the "list" view as a grid for one or more tables in your app.

Solution

Add `list_template=@grid` to the beginning of your table's `fields.ini` file.

This will cause the list to use the `actions/list/grid.html` template for rendering the list view.