How do I extend an exisiting resource list?

Adding a New Column

To add columns to a resource list you can use the **addColumn**, **addColumnBefore**, or **addColumnAfter**methods from within the pipeline. These methods return a usual resource list column which allows you to chain on the other available column methods.

addColumn Method

This method accepts a column header and a closure that defines how the column will be rendered. A column added with this method will be added as the first column.

addColumnBefore Method

This method accepts the same parameters as the **addColumn**method but additionally accepts a third parameter, the key of the column this new column should be added before.

addColumnAfter Method

This method accepts the same parameters as the addColumn method but additionally accepts a third parameter, the key of the column this new column should be added after.

Finding the Current Column Keys

If you do not know the keys of the current columns you can use this code snippet to dump the current keys out.

```
<?php

namespace Acme\MyModule;

use Aero\Admin\ResourceLists\OrdersResourceList;
use Aero\Common\Providers\ModuleServiceProvider;
```

```
class ServiceProvider extends ModuleServiceProvider
{
   public function setup()
   {
       dd((app(OrdersResourceList::class))->getColumns()->map->key());
   }
}
```

```
<?php

namespace Acme\MyModule;

use Aero\Admin\ResourceLists\OrdersResourceList;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
   public function setup()
   {
       OrdersResourceList::extend(function (OrdersResourceList $list) {
           $list->addColumn('My Column', function($row) {
               return 'Example';
           });

           $list->addColumnBefore('Before Ref', function($row) {
               return $row->reference;
           }, 'order');

           $list->addColumnAfter('After Ref', function($row) {
               return $row->reference;
           }, 'order');
       });
   }
}
```

Adding a New Filter

To add a filter to a resource list you can use the **addFilter**method from within the pipeline. This method expects you to pass an admin filter class. You can create an admin filter class as shown in the article "How do I create a custom Admin Filter?

```
<?php

namespace Acme\MyModule;

use Acme\MyModule\Filters\CreatedAtAdminFilter;
use Aero\Admin\ResourceLists\OrdersResourceList;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
   public function setup()
   {
       OrdersResourceList::extend(function (OrdersResourceList $list) {
           $list->addFilter(CreatedAtAdminFilter::class);
       });
   }
}
```

Adding a New Sort By

To add a sort by to a resource list you can use the **addSortBy**method from within the pipeline. This method expects you to pass a **Aero\Admin\ResourceLists\ResourceListSortBy**. You can create a resource list sort by class as shown in the article "How do I create my own Resource List?

```
<?php

namespace Acme\MyModule;

use Aero\Admin\ResourceLists\OrdersResourceList;
use Aero\Admin\ResourceLists\ResourceListSortBy;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
   public function setup()
   {
```

```
        OrdersResourceList::extend(function (OrdersResourceList $list) {
            $list->addSortBy(ResourceListSortBy::create([
                'latest' => 'Latest',
                'oldest' => 'Oldest',
            ], function ($sortBy, $query) {
                return $sortBy === 'oldest' ? $query->orderBy('created_at') : $query->orderByDesc('created_at');
            }));
        });
    }
}
```

Revision #2
Created 2026-09-02 13:28:56 UTC by Michael Price
Updated 2026-09-02 13:37:01 UTC by Michael Price