How do I extend existing reports?

Reports can be extended using a pipeline from your modules service providers setup method or your projects app service providers boot method. You can read more about pipelines here (link).

Adding a Column

To add columns to a table you can use the **addColumn**, **addColumnBefore**, or **addColumnAfter** methods from within the pipeline. These methods return a usual report table 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 when visiting the report.

```
<?php

namespace Acme\MyModule;

use Aero\Admin\Reports\Implementations\SalesReport;
```

```
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
   public function setup()
   {
       SalesReport::extend(function (SalesReport $report) {
           dd($report->table()->getColumns()->map->key());
       });
   }
}
```

```
<?php

namespace Acme\MyModule;

use Aero\Admin\Reports\Implementations\SalesReport;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
   public function setup()
   {
       SalesReport::extend(function (SalesReport $report) {
           $report->table()->addColumn('My Column', function ($row) {
               return 'example';
           })
           ->setExportContent(function ($row) {
               return 'export content';
           });

           $report->table()->addColumnBefore('Before SKU', function ($row) {
               return $row->sku;
           }, 'sku');

           $report->table()->addColumnAfter('After Sku', function ($row) {
               return $row->sku;
           }, 'sku');
```

```
        });
    }
}
```

Adding a Lens

To add a lens to a report you can use the **addLens**method from within the pipeline. This method expects you to pass a report lens class. You can create a report lens class as shown in the article "

How do I add Lenses to my custom Report?

```
<?php

namespace Acme\MyModule;

use Acme\MyModule\Reports\Lenses\TotalRevenueLens;
use Aero\Admin\Reports\Implementations\SalesReport;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        SalesReport::extend(function (SalesReport $report) {
            $report->addLens(TotalRevenueLens::class);
        });
    }
}
```

Adding a Filter

To add a filter to a report 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 add Admin Filters to my custom Report?

```
<?php

namespace Acme\MyModule;
```

```
use Aero\Admin\Filters\Order\OrderStatusAdminFilter;
use Aero\Admin\Reports\Implementations\OrderBreakdownReport;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
   public function setup()
   {
       OrderBreakdownReport::extend(function (OrderBreakdownReport $report) {
           $report->addFilter(OrderStatusAdminFilter::class);
       });
   }
}
```

Revision #2
Created 2026-09-02 13:27:27 UTC by Michael Price
Updated 2026-09-02 13:36:59 UTC by Michael Price