

How do I add lenses to my custom report?

You need to create a class for your lens that extends **Aero\Admin\Reports\ReportLens** and implements the required public value function. The value function will return a string used to display your lens.

```
<?php

namespace Acme\MyModule\Reports\Lenses;

use Aero\Admin\Reports\ReportLens;

class TotalOrdersLens extends ReportLens
{
    protected function value()
    {
        return $this->query->count();
    }
}
```

You can optionally override the **render** method to change the view that your lens uses. The default **render** method:

```
public function render()
{
    return view('admin::reports.lens', ['title' => $this->title(), 'content' => $this->value()]);
}
```

You can also additionally override the **title** method to change the title shown on the frontend. By default this method returns a title that is generated from the lens class name.

Registering a Lens with your Report

To register your newly created lens with your report you need to ensure the report uses the **Aero\Admin\Reports\Traits\HasLens** trait and that your lens class is in the reports protected static \$lenses array.

```
<?php

namespace Acme\MyModule\Reports;

use Acme\MyModule\Reports\Lenses\TotalOrdersLens;
use Aero\Admin\Reports\Report;
use Aero\Admin\Reports\Traits\HasLens;
use Aero\Cart\Models\Order;
use Illuminate\Database\Eloquent\Builder;

class OrdersReport extends Report
{
    use HasLens;

    protected function newQuery(): Builder
    {
        return Order::with([
            'status', 'items', 'currency', 'discounts', 'payments.method',
        ]);
    }

    protected static $lenses = [
        TotalOrdersLens::class,
    ];
}
```

Revision #1

Created 2026-09-02 13:27:25 UTC by Michael Price

Updated 2026-09-02 13:27:25 UTC by Michael Price