

Reports

- [How do I create a report?](#)
- [How do I add a column to an existing report?](#)
- [How do I add a filter to an existing report?](#)
- [How do I add a lens to an existing report?](#)

How do I create a report?

To create a report, make a class that extends `\Aero\Admin\Reports\Report`.

For example, let's make a basic stock report.

First, let's make the report class, this defines the base query for our report:

```
class StockReport extends \Aero\Admin\Reports\Report
{
    use \Aero\Admin\Reports\Traits\HasTables;
    use \Aero\Admin\Reports\Traits\HasFilters;
    use \Aero\Admin\Reports\Traits\CanExport;

    protected static $exports = [
        \Aero\Admin\Reports\Exporters\CSVExporter::class,
        \Aero\Admin\Reports\Exporters\JSONExporter::class,
        \Aero\Admin\Reports\Exporters\XMLExporter::class,
    ];

    protected static $tables = [
        // TODO
    ];

    protected static $filters = [
        // TODO
    ];

    protected function newQuery(): \Illuminate\Database\Eloquent\Builder
    {
        return \Aero\Catalog\Models\Variant::query()
            ->select([
                'variants.id',
                'variants.sku',
                'variants.stock_level',
            ])
            ->groupBy('variants.id');
    }
}
```

```
}
```

Next, let's make a table class, this defines the UI of the report:

```
class StockReportTable extends \Aero\Admin\Reports\ReportTable
{
    public function columns(): array
    {
        $positionStart = 0;
        $positionIncrement = 10;

        return [
            \Aero\Admin\Reports\ReportTableColumn::create('SKU', function ($row) {
                return $row->sku;
            })
            ->position($positionStart = $positionStart + $positionIncrement)
            ->setSearchAction(function ($query, $search) {
                $query->whereLower('sku', 'like', "%{$search}%");
            }),

            \Aero\Admin\Reports\ReportTableColumn::create('Stock', function ($row) {
                return $row->stock_level;
            })
            ->position($positionStart = $positionStart + $positionIncrement),
        ];
    }

    public function emptyMessage(): string
    {
        return 'No stock data';
    }
}
```

Next, let's make a stock level filter:

```
class MinMaxStockFilter extends \Aero\Admin\Filters\MinMaxAdminFilter
{
    public function handleMinMax($min, $max, $query): void
    {
        $query->whereBetween('stock_level', [$min, $max]);
    }
}
```

```

    }

    public function title(): string
    {
        return 'Stock Level';
    }
}

```

Finally, let's hook it all up on the report:

```

class StockReport extends \Aero\Admin\Reports\Report
{
    use \Aero\Admin\Reports\Traits\HasTables;
    use \Aero\Admin\Reports\Traits\HasFilters;
    use \Aero\Admin\Reports\Traits\CanExport;

    protected static $exports = [
        \Aero\Admin\Reports\Exporters\CSVExporter::class,
        \Aero\Admin\Reports\Exporters\JSONExporter::class,
        \Aero\Admin\Reports\Exporters\XMLExporter::class,
    ];

    protected static $tables = [
        \Aerocargo\Acme\Tables\StockReportTable::class,
    ];

    protected static $filters = [
        \Aerocargo\Acme\Filters\MinMaxStockFilter::class,
    ];

    protected function newQuery(): \Illuminate\Database\Eloquent\Builder
    {
        return \Aero\Catalog\Models\Variant::query()
            ->select([
                'variants.id',
                'variants.sku',
                'variants.stock_level',
            ])
            ->groupBy('variants.id');
    }
}

```

```
}
```

And register it so that it shows on the Reports index page:

```
\Aero\Admin\AdminReport::create(\Aerocargo\Acme\StockReport::class)  
  ->title('Stock Report')  
  ->summary('Stock report for all products');
```

How do I add a column to an existing report?

You can use the `addColumn()` method(s) to add a column to an existing report:

```
\Aero\Admin\Reports\Implementations\SalesReport::extend(function
(\Aero\Admin\Reports\Implementations\SalesReport $report, \Closure $next) {
    // Add it as the first column
    $report->table()->addColumn('Header', function ($row) {
        return 'Content';
    });

    // Add it after a specific column
    $report->table()->addColumnAfter('Header', function ($row) {
        return 'Content';
    }, 'name');

    // Add it before a specific column
    $report->table()->addColumnBefore('Header', function ($row) {
        return 'Content';
    }, 'name');

    return $next($report);
});
```

If you need to see the keys of the existing columns in order to position your new column appropriately, you can dump them out:

```
\Aero\Admin\Reports\Implementations\SalesReport::extend(function
(\Aero\Admin\Reports\Implementations\SalesReport $report, \Closure $next) {
    dd($report->table()->getColumns()->map->key());

    return $next($report);
});
```

How do I add a filter to an existing report?

You can use the `addFilter()` method to add a filter to an existing report:

```
\Aero\Admin\Reports\Implementations\SalesReport::extend(function
(\Aero\Admin\Reports\Implementations\SalesReport $report, \Closure $next) {
    $report->addFilter(\Aerocargo\Acme\Filters\MinMaxStockFilter::class);

    return $next($report);
});
```

How do I add a lens to an existing report?

You can use the `addLens()` method to add a lens to an existing report:

```
\Aero\Admin\Reports\Implementations\SalesReport::extend(function
(\Aero\Admin\Reports\Implementations\SalesReport $report, \Closure $next) {
    $report->addLens(\Aerocargo\Acme\Lenses\TotalRevenueLens::class);

    return $next($report);
});
```

```
class TotalRevenueLens extends \Aero\Admin\Reports\ReportLens
{
    protected function value(): \Aero\Cart\CartPrice
    {
        $totalRevenue = \DB::query()
            ->fromSub($this->query, 'results')
            ->selectRaw('SUM(results.revenue + results.tax) as total')
            ->value('total');

        $currency = request()->store()->currency;

        return new \Aero\Cart\CartPrice($currency, $totalRevenue * 100);
    }
}
```

Where the lens added extends `\Aero\Admin\Reports\ReportLens`:

```
class TotalRevenueLens extends \Aero\Admin\Reports\ReportLens
{
    protected function value(): \Aero\Cart\CartPrice
    {
        $totalRevenue = \DB::query()
            ->fromSub($this->query, 'results')
```

```
        ->selectRaw('SUM(results.revenue + results.tax) as total')
        ->value('total');

    $currency = request()->store()->currency;

    return new \Aero\Cart\CartPrice($currency, $totalRevenue * 100);
}
}
```

`$this->query` is a copy of the report query, the `value` method must return a string (or stringable).