

How do I create a custom admin filter?

To create a custom admin filter you must create a class that extends `\Aero\Admin\Filters\AdminFilter` and implement a `handle` method:

```
class OrderStatusAdminFilter extends \Aero\Admin\Filters\AdminFilter
{
    use \Aero\Admin\Filters\Traits\HasDefault;
    use \Aero\Admin\Filters\Traits\AdminFilterHelpers;

    protected $view = 'admin::filters.order-status';

    protected function selectedOptions(): array
    {
        return request()->input($this->key(), []) ?? [];
    }

    public function handle(\Symfony\Component\HttpFoundation\ParameterBag $parameters, $query)
    {
        $view = $parameters->get('view');

        if ($view === null && strlen($parameters->get('q', ''))) {
            $view = 'all';

            request()->request->set('view', $view);
        } elseif ($view === null) {
            $view = static::getDefault();
        }

        $selectedStatusIds = $view === 'incomplete'
            ? collect()
            : collect($parameters->get($this->key()))
                ->map(function ($value) {
                    return intval($value);
                });
    }
}
```

```

        })
        ->filter()
        ->values();

$query->when($view === 'incomplete', function ($query) {
    $query->whereNull('ordered_at');
})->when($view === 'outstanding' || $view === 'new', function ($query) use
($selectedStatusIds) {
    $outstandingStatusIds =
\Aero\Cart\Models\OrderStatus::forState(...setting('admin.outstanding_order_states'))-
>pluck('id');

    $statusIds = $selectedStatusIds->isNotEmpty()
        ? $outstandingStatusIds->intersect($selectedStatusIds)
        : $outstandingStatusIds;

    if ($statusIds->isNotEmpty()) {
        $query->whereIn('order_status_id', $statusIds);
    }
})->when($view === 'all', function ($query) use ($selectedStatusIds) {
    $query->whereNotNull('ordered_at');

    if ($selectedStatusIds->isNotEmpty()) {
        $query->whereIn('order_status_id', $selectedStatusIds);
    }
});
}

protected function viewData(): array
{
    $selectedStatuses = collect(request()->input($this->key()));
    $selectedView = request()->input('view', static::getDefault());

    $orderStatuses = \Aero\Cart\Models\OrderStatus::ordered()->get()
        ->map(function ($status) use ($selectedStatuses) {
            return [
                'id' => $status->id,
                'name' => $status->name,
                'state' => $status->state,
                'url' => $this->getUrlFor($status->id),
            ];
        });
}

```

```

        ];
    });

    return [
        'selectedStatuses' => $selectedStatuses,
        'statuses' => collect($orderStatuses)
            ->filter(function ($status) use ($selectedView) {
                switch ($selectedView) {
                    case 'all': return $status['state'] !== 'incomplete';
                    case 'new':
                    case 'outstanding': return in_array($status['state'],
setting('admin.outstanding_order_states')->toArray());
                }

                return false;
            })
            ->toArray(),
    ];
}

protected function baseName(): string
{
    return 'Status';
}

public function stateFields(): array
{
    return [$this->key(), 'view'];
}

protected static function defaultValue(): string
{
    return 'outstanding';
}
}

```

And then create a view for rendering the filter, in this case `admin::filters.order-status`:

```

<div class="border-b border-grey-light" data-filter-id="status">
    <button class="flex justify-between items-center w-full p-4" type="button"

```

```

onClick="toggleContent(this)">
    <span class="text-sm font-medium">{{ $title }}</span>
    <span class="[ js-accordion-arrow ] w-4 h-4 flex items-center justify-center transform
rotate-90">@include('admin::icons.caret-down')</span>
</button>

<div class="p-4 pt-2 list-reset [ js-accordion-content ] hidden">
    <div class="select w-full -mt-1">
        <select id="view_all" name="view" class="w-full" autocomplete="off"
onChange="updateUrlQuery('view', this.value)">
            <option value="outstanding" {{ request('view',
\\Aero\\Admin\\Filters\\Order\\OrderStatusAdminFilter::getDefault()) === 'outstanding' ||
request('view', \\Aero\\Admin\\Filters\\Order\\OrderStatusAdminFilter::getDefault()) === 'new' ?
'selected' : '' }} data-class="new">Outstanding Orders</option>
            <option value="all" data-class="all"
                @if(request('view',
\\Aero\\Admin\\Filters\\Order\\OrderStatusAdminFilter::getDefault()) == 'all') selected data-
filter-active="true" data-filter-group-name="View" data-filter-name="All Orders" data-filter-
key="view" @endif>All Orders</option>
            <option value="incomplete" data-class="incomplete"
                @if(request('view',
\\Aero\\Admin\\Filters\\Order\\OrderStatusAdminFilter::getDefault()) == 'incomplete') selected
data-filter-active="true" data-filter-group-name="View" data-filter-name="Incomplete Orders"
data-filter-key="view" @endif>Incomplete Orders</option>
        </select>
    </div>

    @if(count($statuses) > 0)
        <div class="pt-4 flex flex-col bg-white mt-2 overflow-hidden">
            <ul class="list-reset overflow-auto leading-loose" style="max-height: 216px">
                @foreach($statuses as $status)
                    <li class="mb-2 filter">
                        <label for="status-{{ $status['id'] }}"><label class="checkbox">
                            <input id="status-{{ $status['id'] }}" type="checkbox"
name="{{ $key }}" autocomplete="off" value="{{ $status['id'] }}" onclick="location.href =
'{{ $status['url'] }}? ' }}" ;"
                                @if($selectedStatuses->contains($status['id']))
checked data-filter-active="true" data-filter-group-name="{{ $title }}" data-filter-name="{{
$status['name'] }}" data-filter-url="{{ $status['url'] }}" @endif
                            > <span></span></label>

```

```
                {{ $status['name'] }}
            </label>
        </li>
    @endforeach
</ul>
</div>
@endif
</div>
</div>
```

Which receives the `viewData` defined by your filter type as well as the `options` defined by the base class (`$title` and `$key`).

Revision #1

Created 2026-09-22 11:40:42 UTC by Ethan Payne

Updated 2026-09-22 11:44:53 UTC by Ethan Payne