

How do I create a custom admin filter?

To create a custom admin filter you need to create a class that extends **Aero\Admin\Filters\AdminFilter** and implements the handle method.

handle Method

This method is executed on every request made to a report or resource list with your filter and accepts 2 parameters. The first parameter is a **Symfony\Component\HttpFoundation\ParameterBag**, and the second parameter is the query.

render Method

This method defines how the filter is rendered. By default it will return a view (that you can set with the protected **\$view** property) and pass in the ****options()** and ****viewData()** arrays.

stateFields Method

This method returns an array of names for the parameters this filter uses in the request. It's important that if you use any parameters outside of the default parameter (which uses the ****\$this->key()** for the name) that you override this method and return them in the array. This method is used to gather all of the fields that should be reset to remove your filter.

options Method

This method returns an array of data that is passed to the view by the render method. If you're creating a new filter that may be used by many classes it is a good idea to use this method instead of the viewData method so that classes that extend your class can use the viewData method if required.

viewData Method

This method returns an array of data that is passed to the view by the render method.

Here is the code for our date range admin filter implementation:

```
<?php

namespace Aero\Admin\Filters;

use Symfony\Component\HttpFoundation\ParameterBag;

abstract class DropdownAdminFilter extends AdminFilter
{
    protected $view = AdminFilterTypes::DROPDOWN;

    protected function selectedOption()
    {
        return request()->input($this->key());
    }

    protected function options(): array
    {
        return [
            'title' => $this->title(),
            'key' => $this->key(),
            'options' => $this->dropdowns(),
            'selected' => $this->selectedOption(),
        ];
    }

    public function handle(ParameterBag $parameters, $query)
    {
        if (($selected = $parameters->get($this->key())) && $selected != '') {
            $this->handleDropdown($selected, $query);
        }
    }

    abstract protected function handleDropdown($selected, $query);

    abstract protected function dropdowns(): array;
}
```

