

How do I configure the admin edit page?

By default your setting group is editable through the admin by a generated edit page. You cannot remove this generated edit page but you can stop users from being able to edit the settings using it. It's important to note that if you disable editing through this edit page that users can still edit the settings manually through the storage json file.

To make your setting group not editable you should call the **notEditable** method when defining your group.

```
<?php

namespace Acme\MyModule;

use Aero\Common\Facades\Settings;
use Aero\Common\Providers\ModuleServiceProvider;
use Aero\Common\Settings\SettingGroup;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        Settings::group('acme-my-module', function (SettingGroup $group) {
            $group->string('denied_message');
            $group->array('allow_list');
            $group->boolean('allow_list_enabled');
            $group->notEditable();
        });
    }
}
```

Validation Rules

These are helper methods that map to their equivalent [Laravel Validation Rules](#)

Rule	Works For
<code>\$group->string('key')->required();</code>	All
<code>\$group->string('key')->between(\$min, \$max);</code>	String, Integer, Float, Array, Encrypted
<code>\$group->string('key')->min(\$min);</code>	String, Integer, Float, Array, Encrypted
<code>\$group->string('key')->max(\$max);</code>	String, Integer, Float, Array, Encrypted
<code>\$group->string('key')->size(\$size);</code>	String, Integer, Float, Array, Encrypted
<code>\$group->string('key')->greaterThan(\$size);</code>	String, Integer, Float, Array, Encrypted
<code>\$group->string('key')->lessThan(\$size);</code>	String, Integer, Float, Array, Encrypted
<code>\$group->string('key')->in(\$needles);</code>	String, Integer, Float
<code>\$group->string('key')->email();</code>	String, Encrypted
<code>\$group->string('key')->uuid();</code>	String, Encrypted
<code>\$group->string('key')->url();</code>	String, Encrypted
<code>\$group->string('key')->ip();</code>	String, Encrypted
<code>\$group->string('key')->startsWith(\$needle);</code>	String, Encrypted
<code>\$group->string('key')->startsWithOneOf(\$needles);</code>	String, Encrypted
<code>\$group->float('key')->step(\$step);</code>	Float
<code>\$group->date('key')->beforeToday();</code>	Date
<code>\$group->date('key')->before(\$date);</code>	Date
<code>\$group->date('key')->afterToday();</code>	Date
<code>\$group->date('key')->after(\$date);</code>	Date

Methods

Method	For	Description
<code>\$group->title(\$title);</code>	-	Set the title shown in the list of settings (by default this is generated from the group name).
<code>\$group->summary(\$summary);</code>	-	Set the summary shown in the list of settings.
<code>\$group->notEditable();</code>	-	Disable editing through the generated admin page.

Method	For	Description
<code>\$group->string('key')->label(\$label);</code>	All	Set the label for the edit field (by default this is generated from the settings key).
<code>\$group->string('key')->hint(\$hint);</code>	All	Add a helpful hint to the edit field.
<code>\$group->string('key')->textarea();</code>	String	Make the string field appear as a textarea.
<code>\$group->string('key')->wysiwyg();</code>	String	Make the string field appear as a wysiwyg.
<code>\$group->eloquent('order_status', OrderStatus::class)->searchField(\$field);</code>	Eloquent	Define the field used for searching in the searchable select.
<code>\$group->eloquent('order_status', OrderStatus::class)->searchFields(\$field);</code>	Eloquent	Define multiple fields used for searching in the searchable select (first field is the value).
<code>\$group->array('key')->associative();</code>	Array	Define the array as an associative array - an array that has a key and value.
<code>\$group->array('key')->definition(\$definition);</code>	Array	Set a custom definition for your array (link:section below this).

Complex Array Definitions

If you have a complex array (an array of arrays that have multiple keys), you need to define the contents of the array and any validation for each key.

Here is a code example that defines an `additional_checkboxes` array that is an array of arrays. The child arrays have a text key (that has validation ensuring it is a string and required) and a required key (that has validation ensuring it is a boolean).

Although this example uses the string, required, and boolean rules you can use any [Laravel Validation rule](#)

```
<?php

namespace Acme\MyModule;

use Aero\Common\Facades\Settings;
use Aero\Common\Providers\ModuleServiceProvider;
use Aero\Common\Settings\SettingGroup;

class ServiceProvider extends ModuleServiceProvider
{
```

```
public function setup()
{
    Settings::group('acme-my-module', function (SettingGroup $group) {
        $group->array('additional_checkboxes')->definition([
            'text' => ['string', 'required'],
            'required' => ['boolean']
        ]);
    });
}
```

Revision #1

Created 2026-09-02 13:29:10 UTC by Michael Price

Updated 2026-09-02 13:29:10 UTC by Michael Price