

General

General

- [How do I set up a slug validator?](#)
- [How do I add a route to the CSRF exception list?](#)
- [What queues does Aero use?](#)
- [How to Set the Page Title for the Admin](#)

How do I set up a slug validator?

You can control if a collection of slugs is valid or not by setting your own validator:

```
\Aero\Store\Routing\Slugs::setValidator(function ($slugs, $resolver) {  
    if ($resolver !== 'listings') return true;  
  
    if ($slugs->count() < 3 && $slugs->where('model_type', 'tag')->count()) return false;  
  
    return true;  
});
```

How do I add a route to the CSRF exception list?

If you do not want your route to require a valid [CSRF token](#)

Code

To add a route to the exceptions list through code you need to add to the static **VerifyCsrfToken::\$exceptions** array.

```
<?php

namespace Acme\MyModule;

use Aero\Common\Providers\ModuleServiceProvider;
use Aero\Store\Http\Middleware\VerifyCsrfToken;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        VerifyCsrfToken::$exceptions[] = '/my-route';
    }
}
```

Configuration

The **routing** configuration file has a **csrf_exceptions** array where you can list routes for the exceptions list.

You can publish the **routing** configuration file with this command:

```
php artisan vendor:publish --provider="Aero\Routing\Providers\RoutingServiceProvider"
```

Once this file is published you can view it by navigating to **config/aero/routing.php** in your project. You can edit the **csrf_exceptions** array directly in this file.

You may need to clear the config cache for your changes to take effect.

```
php artisan config:clear
```

What queues does Aero use?

Queues are used so that HTTP requests can remain snappy and more intensive code can be executed later. You can [learn more about queues here](#)

Queue	Description
search	This queue has search jobs. The majority of these jobs will be responsible for reindexing listings.
email	This queue has email jobs. Whenever an email is sent and queued in your application, it will be in this queue.
subscriptions	This queue has subscription jobs. The majority of these jobs are responsible for creating and processing subscriptions.
default	This queue has any jobs that are not in one of the other queues. These jobs can be anything from order created event handlers to customer created event handlers.

Aero uses four queues so that you can have different workers completing different tasks without a risk of locks or jobs being handled more than once. It's important to note that some queues should be considered more important than others (for example, processing events such as an order created event is more important than updating your search index). In a production environment you should use tools like [Supervisor](#) to manage your queue workers.

You can use the queue work command to run all of or only specific queues (if you adjust the list of queues in the command). It's important to note that the order of the queues in the command defines their priority.

How to Set the Page Title for the Admin

By default the page title for every admin page is generated from the route name. If you would like to manually set the page title you can set a title specifically for your route or you can return an `adminPageTitle` variable to your view (this includes extending a response builder and adding this to the data).

To set the title manually for a specific route you can use the `Aero\Admin\Http\Middleware\SetTitle::addTitle()` method. This method expects a route name and a page title.

```
<?php

namespace Acme\MyModule;

use Aero\Admin\Http\Middleware\SetTitle;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        SetTitle::addTitle('admin.dashboard', 'The Dashboard Title');
    }
}
```