

How do I create a custom page in the account area?

To create and register a custom account area page you need to use the `**Aero\AccountArea\AccountArea::registerPage()` helper method. This method expects you to pass a class that extends **Aero\AccountArea\AccountAreaPage**.

```
<?php

namespace Acme\MyModule;

use Acme\MyModule\AccountArea\Pages\CustomPage;
use Aero\AccountArea\AccountArea;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        AccountArea::registerPage(CustomPage::class);
    }
}
```

Your account area page class must implement the **title**, **route**, and **routeName** methods that all should return a string. It's common practice to use [Laravel Localization](#) for the title and route so that they can change depending on the language used. You should not use this for the routeName as the routeName is used in code to refer to the route and therefore should never change.

You should also define a protected static steps and middleware array and a protected sections array. The static steps array can be used to define any extra response steps required for your page. The middleware array can be used to define any middleware required for your page (the most useful being 'account' which means users need to be logged in to view your page). The sections array defines the views that make up your page.

```
<?php

namespace Acme\MyModule\AccountArea\Pages;

use Aero\AccountArea\AccountAreaPage;

class CustomPage extends AccountAreaPage
{
    protected static $steps = [
        Steps\AttachWishlist::class,
    ];

    protected static $middleware = [
        'account',
    ];

    protected static $sections = [
        'navbar' => 'account-area::sections.navbar',
        'page-header' => 'account-area::partials.page-header',
        'wishlist' => 'wishlist::account-area-wishlist',
    ];

    static function title(): string
    {
        return __('My Custom Page');
    }

    static function route(): string
    {
        return __('custom-page');
    }

    static function routeName(): string
    {
        return 'acme.my-module.custom-page';
    }
}
```

Adding your Page to the Navbar

To add your page to the navbar you need to make sure your page class uses the **Aero\AccountArea\Traits\HasAccountAreaLink** trait and then add the link to the navbar using the **addLink**, **addLinkBefore**, or **addLinkAfter** methods from within the pipeline.

The **Aero\AccountArea\Traits\HasAccountAreaLink** trait makes you implement a **toLink** method that returns an array of data used to render your link. The array should have an icon and text key.

```
public static function toLink(): array
{
    return [
        'icon' => 'my-module::icon',
        'text' => 'My Page',
    ];
}
```

```
<?php

namespace Acme\MyModule;

use Acme\MyModule\AccountArea\Pages\CustomPage;
use Aero\AccountArea\AccountAreaPage;
use Aero\Common\Providers\ModuleServiceProvider;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        AccountAreaPage::extend(function ($page) {
            $page->addLink(CustomPage::class);
        });
    }
}
```

Revision #1

Created 2026-09-02 13:27:07 UTC by Michael Price

Updated 2026-09-02 13:27:07 UTC by Michael Price