# 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`:

```
\Aero\AccountArea\AccountArea\AccountArea::registerPage(\Aerocargo\Acme\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.

For example, here's how `aerocargo/wishlists` defines its page:

```
class WishlistPage extends \Aero\AccountArea\AccountAreaPage
{
    use \Aero\AccountArea\Traits\HasAccountAreaLink;

    protected static $steps = [
        Steps\AttachWishlist::class,
    ];

    protected static $middleware = [
        'account',
```

```
    ];

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

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

    static function route(): string
    {
        return __('wishlist');
    }

    static function routeName(): string
    {
        return 'account.wishlist';
    }

    public static function toLink(): array
    {
        return [
            'icon' => 'wishlist::icons.heart',
            'text' => 'Your Wishlist',
        ];
    }
}
```

You can add your page to the navbar using the `addLink()` method:

```
\Aero\AccountArea\AccountAreaPage::extend(function (\Aero\AccountArea\AccountAreaPage $page,
\Closure $next) {
    $page->addLink(\Aerocargo\Wishlist\Http\Responses\WishlistPage::class);

    return $next($page);
});
```

> For `addLink` to work, your page needs to use the `\Aero\AccountArea\Traits\HasAccountAreaLink` trait and implement the `toLink()` method (see the wishlist example above)

---

Revision #1
Created 2026-09-25 12:45:14 UTC by Ethan Payne
Updated 2026-09-25 12:53:15 UTC by Ethan Payne