# Pages


- What are the default pages in the account area?
- How do I create a custom page in the account area?
- How do I add a section to an existing account area page?

# What are the default pages in the account area?

| Class | Route | Description |
| --- | --- | --- |
| `AccountLoginPage` | `login` | A page where users can log in. |
| `AccountRegisterPage` | `register` | A page where guests can register. |
| `AccountPasswordForgotPage` | `password.forgot` | A page where users can send a password reset email to their email. |
| `AccountPasswordResetPage` | `password.reset` | A page where users can reset their password. |
| `AccountOrdersPage` | `account.orders` | A page that lists all of a user's orders and lets them filter/search. |
| `AccountOrderViewPage` | `account.view-order` | A page that shows more information about an order. |
| `AccountAddressesPage` | `account.addresses` | A page that lists all of the users addresses. |
| `AccountAddressNewPage` | `account.new-address` | A page where users can create a new address, |
| `AccountAddressEditPage` | `account.edit-address` | A page where users can edit one of their addresses. |
| `AccountDetailsPage` | `account.details` | A page where users can update their account details and change their password. |
| `AccountInvoicePage` | `account.view-invoice` | A simple invoice for an order - currently not linked to from anywhere. |
| `AccountOverviewPage` | `account` | The first page a user sees after they log in - currently redirects to the orders page. |

> The above classes reside in the `Aero\AccountArea\Http\Responses` namespace.

# 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)

# How do I add a section to an existing account area page?

You can use the `addSection()` method(s) to add a section to an existing form:

```
\Aero\AccountArea\Http\Responses\AccountLoginPage::extend(function
(\Aero\AccountArea\Http\Responses\AccountLoginPage $page, \Closure $next) {
    // Add it as the first section
    $page->addSection('acme', 'acme::section');

    // Add it after a specific section
    $page->addSectionAfter('acme', 'acme::custom', 'login-form');

    // Add it before a specific section
    $page->addSectionBefore('acme', 'acme::custom', 'login-form');

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

If you need to see the keys of the existing columns in order to position your new column appropriately, you can dump them out:

```
\Aero\AccountArea\Http\Responses\AccountLoginPage::extend(function
(\Aero\AccountArea\Http\Responses\AccountLoginPage $page, \Closure $next) {
    dd($page->getSections()->keys()->toArray());

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