How do I extend responses?

Adding additional code to run on the response is as simple as calling the **extend()**method on the response builder class. This is typically done from within a **Service Provider**. If the code is unique to a particular store it could be placed in the boot method of the **AppServiceProvider**. If it is to be included as part of a module, then it can be added to the setup method of the module's Service Provider.

For example, if a module needs to send a variable **foo**to the **product.twig**view file for it to be outputted in the HTML, the **ProductPage**response builder class should be extended in the module's **Service Provider**as shown below:

```
<?php

namespace Acme\MyModule;

use Aero\Common\Providers\ModuleServiceProvider;
use Aero\Store\Http\Responses\ProductPage;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        ProductPage::extend(function ($page) {
            $page->setData('foo', 'bar');
        });
    }
}
```

When passing a **Closure**to the extend method, the **$page**argument is the response builder instance that will process the response.

For more advanced use cases, you can pass the response builder on for further processing in order to return the **Illuminate\Http\Response**instance. This is useful in situations when you may wish to modify the response before it is passed to the browser. A common use case for this is to add cookies to the response:

```
\Aero\Store\Http\Responses\ProductPage::extend(function ($page, $next) {
    $response = $next($page);

    $response->cookie('foo', 'bar');

    return $response;
});
```

Alternatively, a fully qualified class name can be passed to the **extend()**method, which reduces bloat in the **Service Provider**and provides a better indication of what the code is responsible for. When doing so, the class must implement a **handle()**method:

```
<?php

namespace App\Http\Extensions;

class AddFooVariableToProductPage
{
    public function handle($page)
    {
        $page->setData('foo', 'bar');
    }
}
```

```
\Aero\Store\Http\Responses\ProductPage::extend(AddFooVariableToProductPage::class);
```

Note that this code will only evaluate when Aero routes a request to the particular response builder, i.e. code that is set to run on the **ProductPage**will not be processed on a request for the **Homepage**.

Revision #1
Created 2026-09-02 13:29:04 UTC by Michael Price
Updated 2026-09-02 13:29:04 UTC by Michael Price