

Vue

- [How do I use a custom vue component in the storefront?](#)

How do I use a custom vue component in the storefront?

1. You need to add the `assetLinks` function to your modules service provider so that your modules assets are publicly linkable. An example module service provider can be found here:

<https://github.com/aerocargo/admin-example-vue/blob/master/src/ServiceProvider.php>

2. You also need to make your component `.vue` file(s) and your `main.js` file that will register all of the components. An example javascript structure can be found here:

<https://github.com/aerocargo/admin-example-vue/tree/master/resources/js>

3. You need to run `php artisan aero:link` in your terminal to link the assets.
4. You then need to register the javascript file using the `Aero\Components\Facades\Component` facade `import` method:

```
\Aero\Components\Facades\Component::import('/modules/aerocargo/acme/components.js');
```

This facade is provided by the `aerocommerce/components` module.

5. Once you've got your files setup, have ran the link command and have built the javascript files (using npm: `npm run dev` or `npm run production`).

Usage

{% component %}

The `component` tag will render the component in line with the `aero.ssr.enabled` config value, if `true` then the component will be server-side rendered:

```
{% component "product" with {count: 100} csr %}  
<div>  
  <example-component :count="count" />  
</div>  
{% endcomponent %}
```

Using `csr` means that the component will always be client-side rendered.
Using `ssr` means the component will always be server-side rendered.

{% vue %}

The `vue` tag will render the component client-side by default:

```
{% vue with {count: 100} ssr %}  
<div>  
  <example-component :count="count" />  
  <another-example-component v-model="count" />  
</div>  
{% endvue %}
```

Using `ssr` means the component will be server-side rendered.