

How do I use Vue in the admin?

When in an admin's Blade view file you can use any Vue directive (such as `v-for`, `v-if`, `@click` etc).

Putting Vue into Dev Mode

Putting Vue into dev mode allows you to use the Vue dev tools. You can put the Admin Vue into dev mode by putting this code snippet in your Blade view:

```
@push('scripts')
<script>
    window.AeroAdmin.vue.use({
        install(Vue) {
            Vue.config.devtools = true;
        },
    });
</script>
@endpush
```

Extending Vue without a Custom Component

You can extend Vue without using a custom Vue component by using some of the `window.AeroAdmin` methods. Methods are provided for you to add **data**, **methods**, **computed methods**, **watchers**, and **listeners**.

Adding Data

You can make data available in Vue through the `window.AeroAdmin.addData` method. This method expects two parameters, the variable name and the variable value. You can pass JSON as the values for complex data types such as arrays.

```
@push('scripts')
  <script>
    window.AeroAdmin.addData('myVariable', '{{ old('myVariable', 'value') }}');
    window.AeroAdmin.addData('myOtherVariable', 'Example');
    window.AeroAdmin.addData('myArrayVariable', {!! json_encode(['one', 'two', 'three',
'four']) !!});
  </script>
@endpush
```

Adding a Method

You can make Vue methods using the `window.AeroAdmin.addMethod` method. This method expects two parameters, the method name and a closure defining the method. Inside of the closure you can use **this** to reference the Vue instance (this allows you to call other methods or interact with the data variables (anything you can do from within Vue)).

```
@push('scripts')
  <script>
    window.AeroAdmin.addMethod('test', function (value) {
      console.log(value);
    });

    window.AeroAdmin.addMethod('testing', function (value) {
      this.test(value);
    })
  </script>
@endpush
```

Adding a Watcher

You can make a Vue watcher using the `window.AeroAdmin.addWatch` method. This method expects two parameters, the watcher name and a closure defining the watcher. Inside of the closure you can use **this** to reference the Vue instance (this allows you to call other methods or interact with the data variables (anything you can do from within Vue)).

```
@push('scripts')
<script>
  window.AeroAdmin.addData('testing', 'value');

  window.AeroAdmin.addMethod('test', function (value) {
    console.log(value);
  });

  window.AeroAdmin.addWatch('testing', function (value) {
    this.test(value);
  })
</script>
@endpush
```

Adding a Computed Method

You can make Vue computed methods using the `window.AeroAdmin.addComputed` method. This method expects two parameters, the computed method name and a closure defining the computed method. Inside of the closure you can use **this** to reference the Vue instance (this allows you to call other methods or interact with the data variables (anything you can do from within Vue)).

```
@push('scripts')
<script>
  window.AeroAdmin.addComputed('computedMethod', function () {
    return ['one', 'two', 'three', 'four'];
  });
</script>
@endpush
```

Adding a Listener

You can add a listener to Vue using the `window.AeroAdmin.addListener` method. This method expects two parameters, the event name to listener for and a closure defining what to do when the event is emitted. Inside of the closure you can use **this** to reference the Vue instance (this allows you to call other methods or interact with the data variables (anything you can do from within Vue)).

```
@push('scripts')
<script>
  window.AeroAdmin.addListener('loaded', function () {
```

```
        console.log('Vue has been loaded');
    });
</script>
@endpush
```

Revision #1

Created 2026-09-02 13:27:34 UTC by Michael Price

Updated 2026-09-02 13:27:34 UTC by Michael Price