Events

- What are the available events?
- How do I run a custom event?

What are the available events?

Products

Event	Description
`product.loaded`	Fired when the product component has finished mounting and is marked as loaded.
`product.images-updating`	Fired when the product images are about to be updated because the image set changed.
`product.images-updated`	Fired after the product images have been updated. Provides the updated `images`.
`product.variant-selected`	Fired when the selected product variant changes. Provides the variant, quantity, price, and total sale price.
`product.attribute-selected`	Fired when one or more product attributes are selected or changed. Provides the selected attributes.
`product.quantity-updated`	Fired when the product quantity changes. Provides the new quantity.
`product.subscription-plan-deselected`	Fired when the currently selected subscription plan is removed/deselected.
`product.subscription-plan-selected`	Fired when a subscription plan is selected. Provides the plan ID and selected plan.
`product.add-to-cart`	Fired before the product is sent to the cart API. This event can modify the cart payload or return `false`/`null` to prevent adding the item.
`product.added-to-cart`	Fired after the cart API successfully adds the product. Provides the returned cart/add-to-cart data.
`product.failed-adding-to-cart`	Fired when adding the product to the cart fails. Provides the returned errors.

Listings

Event	Description
`listings.loaded`	Fired after the listings component has mounted and its initial data has been loaded.

`listings.update`	Listener event that triggers a listings update/navigation to the supplied URL.
`listings.updating`	Fired after a listings request returns but before the response data is applied to the Vue component. Can modify the returned listings data.
`listings.updated`	Fired after the returned listings data has been applied to the component and Vue has completed the next tick.
`listings.navigated`	Fired when listings data is updated through a navigation that does not involve the component's normal history push/replace handling.

How do I run a custom event?

You can run your own custom event using the `runEvent()` method:

```
window.Aero.runEvent('acme', { foo: 'bar' });
```

And listen for it wherever needed:

```
window.AeroEvents.on('acme', (data) => {
    console.log(data);
});
```