

Google Analytics (GA4) Integration Instructions for the Email Course Landing Page

To measure user interactions on the **Bloating Myth-Busting Email Course** page and within the emails, use GA4 custom events triggered via data attributes. Data attributes allow you to add event information directly to HTML elements without relying on CSS classes or IDs. Analytics expert David Vallejo describes the approach as follows:

- Use a `data-ga4-event` attribute to identify the event name (e.g., `data-ga4-event="cta_click"`) ¹.
- Use `data-ga4-param-*` attributes to specify event parameters such as link name or product category ².
- When a user clicks an element with `data-ga4-event`, a script should read the attributes and push a `dataLayer` event with the event name and parameters to GA4 ³.

1. Adding Data Attributes

Add the following data attributes to interactive elements on the course landing page and in email templates:

Element	Example HTML	Purpose
CTA button to join the course	<code>Join Now</code>	Tracks when users click to opt into the course.
Affiliate product link	<code>Digestive Enzyme Supplement</code>	Captures which product categories are clicked and from which email day.
Consultation booking link	<code>Book a Consultation</code>	Tracks conversions to consultation scheduling.

2. JavaScript Snippet

Include a JavaScript snippet on the course landing page (and in your email template header if supported) to listen for clicks on elements with the `data-ga4-event` attribute and push them to the `dataLayer`. Vallejo's example shows how to loop through all tagged elements and convert parameter names to `snake_case` ³:

```

<script>
(function() {
  var events = document.querySelectorAll('[data-ga4-event]');
  var unCamelCase = function(str, separator) {
    separator = typeof separator === 'undefined' ? '_' : separator;
    return str.replace(/[a-z\d]([A-Z])/g, '$1' + separator + '$2')
      .replace(/([A-Z]+)([A-Z][a-z\d]+)/g, '$1' + separator + '$2')
      .toLowerCase();
  };
  for (var i = 0; i < events.length; i++) {
    events[i].addEventListener('click', function(event) {
      var target = event.currentTarget;
      if (target) {
        var dl = {};
        dl['event'] = target.dataset['ga4Event'];
        Object.entries(target.dataset).forEach(function(e) {
          var key = e[0];
          var value = e[1];
          var m = key.match('ga4Param(.+)');
          if (m && m[1]) {
            dl[unCamelCase(m[1], '_')] = value;
          }
        });
        window.dataLayer = window.dataLayer || [];
        window.dataLayer.push(dl);
      }
    });
  }
})();
</script>

```

3. Configuring GA4 or GTM

- If using **Google Tag Manager (GTM)**, create a trigger that fires on clicks matching the CSS selector `[data-ga4-event]`. Set up a custom event tag that sends the event name (`{{Click Element}} dataset.ga4Event`) and parameters (`gtm.element.dataset.paramName`) to GA4.
- If using **gtag.js** directly, modify the click handler to send events via `gtag('event', eventName, parameters)` instead of pushing to the dataLayer.

4. Testing and Validation

- Use the GA4 DebugView or the Tag Assistant Chrome extension to verify that events fire correctly when you click buttons or links.
- Check that parameters such as `product_category` and `email_day` are recorded as custom dimensions for analysis.

Following these guidelines will help you understand how users engage with your course and which calls-to-action drive the most conversions.

1 2 3 Tracking Google Analytics 4 Events using Data Attributes - David Vallejo

<https://www.thyngster.com/tracking-google-analytics-4-events-using-data-attributes>