

Elevar DataLayer

Last Updated / Reviewed: April 11th, 2023

Execution Time: ~15 minutes

Current Version: Version 3.1.10

Goal: To create dataLayer codes inside Shopify to track all ecommerce events, including purchase revenue.

Ideal Outcome: Get more than 95% data accuracy into GA4 reporting.

Configure the Checkout Thank You Page

If you are on a Shopify Plus store and would like to track the checkout pages, then you'll need to enable "checkout.liquid" first - [View Guide](#)

If you are on a Non-Plus Shopify Store, you will not be able to track the checkout pages - [View Guide](#)

1. Copy the code from the block below:

Please change all GTM container ID in yellow marked

```
<!--  
Elevar Data Layer V2  
  
This file is automatically updated and should not be edited directly.  
  
https://knowledge.getelevar.com/how-to-customize-data-layer-version-2  
  
Updated: 2023-04-11 09:11:34+00:00  
Version: 3.1.10  
-->  
<!-- Google Tag Manager -->  
<script>  
  window.dataLayer = window.dataLayer || [];  
</script>  
<script>
```

```
(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({"gtm.start":
  new Date().getTime(),event:"gtm.js"});var f=d.getElementsByTagName(s)[0],
  j=d.createElement(s),dl=l!="dataLayer"? "&l="+l:"";j.async=true;j.src=
  "https://www.googletagmanager.com/gtm.js?id="+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,"script","dataLayer","GTM-XXXXXXX");
</script>
<!-- End Google Tag Manager -->
<script id="elevar-gtm-suite-config" type="application/json">{"gtm_id":
"GTM-XXXXXXX", "event_config": {"cart_reconcile": true, "cart_view": true,
"checkout_complete": true, "checkout_step": true, "collection_view": true,
"defers_collection_loading": false, "defers_search_results_loading": false,
"product_add_to_cart": false, "product_add_to_cart_ajax": true,
"product_remove_from_cart": true, "product_select": true, "product_view": true,
"search_results_view": true, "user": true, "save_order_notes": true},
"gtm_suite_script":
"https://shopify-gtm-suite.getelevar.com/shops/d0b0038eba47fc05bb116d997e301563
dd42b6d5/3.1.10/gtm-suite.js", "consent_enabled": true, "apex_domain": null}</script>

{% if first_time_accessed %}
<script>
  (() => {
    const init = () => {
      window.__ElevarDataLayerQueue = [];
      window.__ElevarListenerLoadQueue = [];
      if (!window.dataLayer) window.dataLayer = [];
    }
    init();
    window.__ElevarTransformItem = event => {
      if (typeof window.ElevarTransformFn === "function") {
        try {
          const result = window.ElevarTransformFn(event);
          if (typeof result === "object" && !Array.isArray(result) && result !== null) {
            return result;
          } else {
            console.error("Elevar Data Layer: `window.ElevarTransformFn` returned a value
            " + "that wasn't an object, so we've treated things as if this " + "function wasn't
            defined.");
            return event;
          }
        } catch (error) {
          console.error("Elevar Data Layer: `window.ElevarTransformFn` threw an error, so
```

```
" + "we've treated things as if this function wasn't defined. The " + "exact error is
shown below.");
    console.error(error);
    return event;
  }
} else {
  return event;
}
}

window.ElevarPushToDataLayer = item => {
const enrichedItem = {
  event_id: window.crypto.randomUUID(),
  event_time: new Date().toISOString(),
  ...item
};
const transformedEnrichedItem = window.__ElevarTransformItem ?
window.__ElevarTransformItem(enrichedItem) : enrichedItem;
const listenerPayload = {
  raw: enrichedItem,
  transformed: transformedEnrichedItem
};
const getListenerNotifyEvent = () => {
  return new CustomEvent("elevar-listener-notify", {
    detail: listenerPayload
  });
};
if (transformedEnrichedItem._elevar_internal?.isElevarContextPush) {
  window.__ElevarIsContextSet = true;
  window.__ElevarDataLayerQueue.unshift(transformedEnrichedItem);
  if (window.__ElevarIsListenerListening) {
    window.dispatchEvent(getListenerNotifyEvent());
  } else {
    window.__ElevarListenerLoadQueue.unshift(listenerPayload);
  }
} else {
  window.__ElevarDataLayerQueue.push(transformedEnrichedItem);
  if (window.__ElevarIsListenerListening) {
    window.dispatchEvent(getListenerNotifyEvent());
  } else {
    window.__ElevarListenerLoadQueue.push(listenerPayload);
  }
}
```

```
}
if (window.__ElevaIsContextSet) {
  while (window.__ElevaDataLayerQueue.length > 0) {
    window.dataLayer.push(window.__ElevaDataLayerQueue.shift());
  }
}
}

const configElement = document.getElementById("elevar-gtm-suite-config");

if (!configElement) {
  console.error("Eleva: DL Config element not found");
  return;
}

const config = JSON.parse(configElement.textContent);

const script = document.createElement("script");
script.type = "text/javascript";
script.src = config.gtm_suite_script;
script.async = false;
script.defer = true;

script.onerror = () => {
  console.error("Eleva: DL JS script failed to load");
};
script.onload = async () => {
  if (!window.ElevaGtmSuite) {
    console.error("Eleva: `ElevaGtmSuite` is not defined");
    return;
  }

  const cartData = {
    attributes: {{- cart.attributes | json -}},
    cartTotal: "{{- cart.total_price | times: 0.01 | json -}}",
    currencyCode: {{- cart.currency.iso_code | json -}},
    items: [
      {%- for line_item in cart.items -%}
      {
        {%- if line_item.sku != blank -%}
        id: {{- line_item.sku | json -}},
```

```
{%- else -%}
  id: "{{- line_item.product_id | json -}}",
{%- endif -%}
name: {{- line_item.product.title | json -}},
brand: {{- line_item.vendor | json -}},
category: {{- line_item.product.type | json -}},
variant: {{- line_item.variant.title | json -}},
price: "{{- line_item.final_price | times: 0.01 | json -}}",
position: {{- forloop.index -}},
quantity: "{{- line_item.quantity | json -}}",
productId: "{{- line_item.product_id | json -}}",
variantId: "{{- line_item.variant_id -}}",
compareAtPrice: "{{- line_item.variant.compare_at_price | times: 0.01 | json -}}",
image: "{{- line_item.image | image_url -}}"
},
{%- endfor -%}
]
}
;

await window.ElevarGtmSuite.handlers.cartAttributesReconcile(
  cartData,
  config.event_config.save_order_notes,
  config.consent_enabled,
  config.apex_domain
);

if (config.event_config.user) {
  const data = {
{%- if checkout -%}
    cartTotal: "{{- checkout.total_price | times: 0.01 | json -}}",
    currencyCode: {{- checkout.currency | json -}},
{%- else -%}
    cartTotal: "{{- cart.total_price | times: 0.01 | json -}}",
    currencyCode: {{- cart.currency.iso_code | json -}},
{%- endif -%}
{%- if customer -%}
    customer: {
      id: "{{- customer.id | json -}}",
      email: {{- customer.email | json -}},
      firstName: {{- customer.first_name | json -}},
```

```
lastName: {{- customer.last_name | json -}},
phone: {{- customer.phone | json -}},
city: {{- customer.default_address.city | json -}},
zip: {{- customer.default_address.zip | json -}},
address1: {{- customer.default_address.address1 | json -}},
address2: {{- customer.default_address.address2 | json -}},
country: {{- customer.default_address.country | json -}},
province: {{- customer.default_address.province | json -}},
provinceCode: {{- customer.default_address.province_code | json -}},
orderCount: "{{- customer.orders_count | json -}}",
totalSpent: "{{- customer.total_spent | times: 0.01 | json -}}",
tags: {{- customer.tags | join: ' ' | json -}}
}
{%%- endif -%}
};

    window.ElevaGtmSuite.handlers.user(data);
}

{%%- if checkout -%}
{%%- if checkout.attributes["_eleva_visitor_info"] == blank -%}
if (config.event_config.save_order_notes) {
    const data = {%%- if checkout -%}
    {
        checkoutId: {{- checkout.id | json -}}
    }
    {%%- endif -%}
    ;
    window.ElevaGtmSuite.handlers.storeOrderNotesOffsite(data);
}
{%%- endif -%}

if (config.event_config.checkout_complete) {
    const data = {%%- if checkout -%}
    {
        currencyCode: {{- checkout.currency | json -}},
        actionField: {
            {%%- if checkout.order_id -%}
            id: {{- checkout.order_id | json -}},
            {%%- else -%}
            id: {{- checkout.id | json -}},
            {%%- endif -%}
        }
    }
    {%%- endif -%}
```

```
{%- if checkout.order_name -%}
  order_name: {{- checkout.order_name | json -}},
{%%- endif -%}
affiliation: {{- shop.name | json -}},
revenue: "{{- checkout.total_price | times: 0.01 | json -}}",
tax: "{{- checkout.tax_price | times: 0.01 | json -}}",
shipping: "{{- checkout.shipping_price | times: 0.01 | json -}}",
{% if checkout.discount_applications %}
  coupon: {{ checkout.discount_applications[0].title | json }},
{% endif %}
{%%- if order.subtotal_price -%}
  subTotal: "{{- order.subtotal_price | times: 0.01 | json -}}",
{%%- elsif checkout.subtotal_price -%}
  subTotal: "{{- checkout.subtotal_price | times: 0.01 | json -}}",
{%%- endif -%}
productSubTotal: "{{- checkout.line_items_subtotal_price | times: 0.01 | json -}}",
discountAmount: "{{- checkout.discounts_amount | times: 0.01 | json -}}",
},
{%%- if checkout.customer -%}
customer: {
  id: "{{- checkout.customer.id | json -}}",
  email: {{- checkout.email | json -}},
  firstName: {{- checkout.billing_address.first_name | json -}},
  lastName: {{- checkout.billing_address.last_name | json -}},
  {%%- if checkout.customer.phone -%}
    phone: {{- checkout.customer.phone | json -}},
  {%%- elsif checkout.billing_address.phone -%}
    phone: {{- checkout.billing_address.phone | json -}},
  {%%- else -%}
    phone: {{- checkout.shipping_address.phone | json -}},
  {%%- endif -%}
  city: {{- checkout.billing_address.city | json -}},
  zip: {{- checkout.billing_address.zip | json -}},
  address1: {{- checkout.billing_address.address1 | json -}},
  address2: {{- checkout.billing_address.address2 | json -}},
  country: {{- checkout.billing_address.country | json -}},
  province: {{- checkout.billing_address.province | json -}},
  provinceCode: {{- checkout.billing_address.province_code | json -}},
  orderCount: "{{- checkout.customer.orders_count | json -}}",
  totalSpent: "{{- checkout.customer.total_spent | times: 0.01 | json -}}",
  tags: {{- checkout.customer.tags | json -}}
```

```
    },
    {%- endif -%}
    items: [
      {%- for line_item in checkout.line_items -%}
      {
        id: {{- line_item.sku | json -}},
        name: {{- line_item.product.title | json -}},
        brand: {{- line_item.vendor | json -}},
        category: {{- line_item.product.type | json -}},
        variant: {{- line_item.variant.title | json -}},
        price: "{{- line_item.final_price | times: 0.01 | json -}}",
        quantity: "{{- line_item.quantity | json -}}",
        productId: "{{- line_item.product_id | json -}}",
        variantId: "{{- line_item.variant_id -}}",
        image: "{{- line_item.image | image_url -}}"
      },
      {%- endfor -%}
    ],
    landingSite: {{- checkout.landing_site | json -}}
  }
{%- endif -%}
;
    window.ElevaGtmSuite.handlers.checkoutComplete(data);
  }
  {%- endif -%}
};

    document.head.appendChild(script);
  })();
</script>
{% endif %}

<!-- Google Tag Manager (noscript) -->
<noscript>
  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XXXXXXX"
height="0" width="0" style="display:none;visibility:hidden"></iframe>
</noscript>
<!-- End Google Tag Manager (noscript) -->
```

2. Open your "Checkout Settings" admin page (**settings > Checkout**).

3. Scroll down to the **"Order status page"** section of the page.
4. Paste the code from above into the end of the **"Additional scripts"** input box:

Order status page ⓘ
Add tracking scripts and other customizations.

Additional scripts

<!-- Google Tag Manager -->

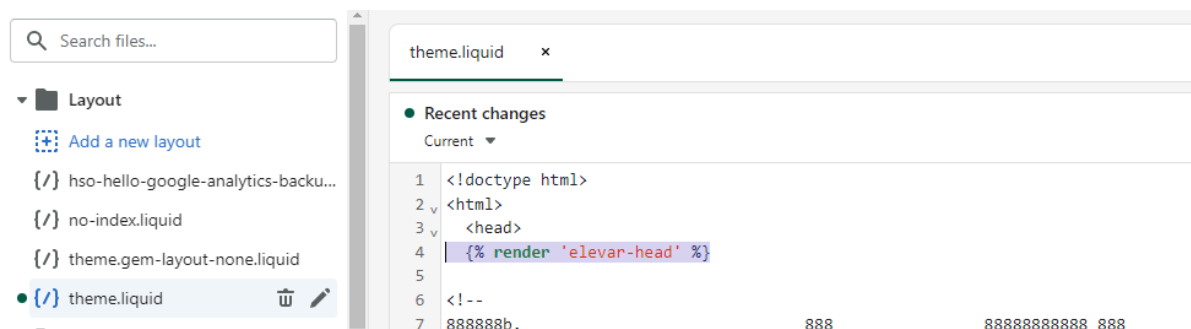
5. Click "Save".

Step-01: Install Data Layer 1st call code on Shopify

Login into Shopify admin panel:

Online Store----> Themes ----> Actions ----> Edit Code ----> Layout ----> theme.liquid (File) ---->
> Put the Data Layer call code directly below the opening <head> tag

```
{% render 'elevar-head' %}
```

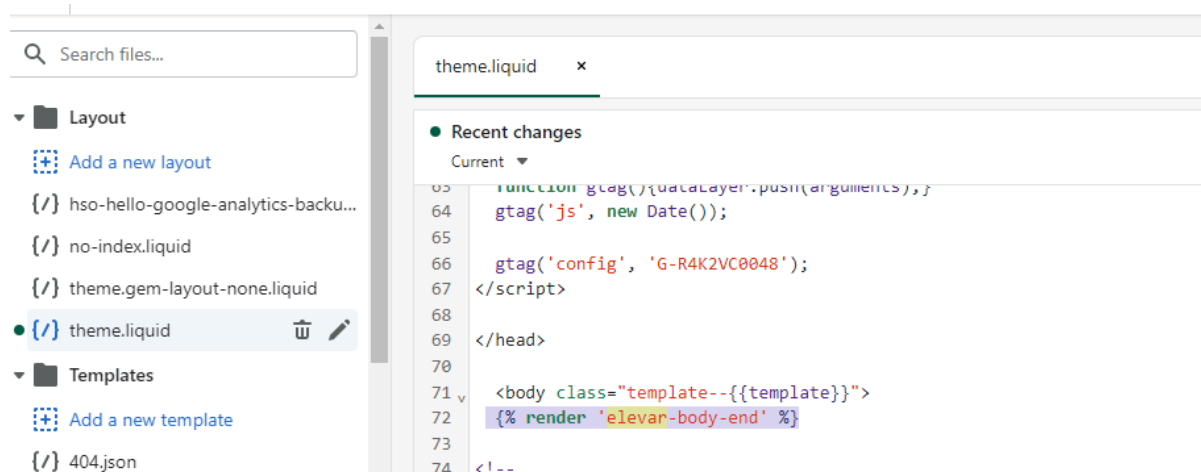


Step-02: Install Data Layer call 2nd code on Shopify

Login into Shopify admin panel:

Online Store----> Themes ----> Actions ----> Edit Code ----> Layout ----> theme.liquid (File) ---->
> Put the Data Layer call code directly below the opening fast <body> tag

`{% render 'elevar-body-end' %}`



Create Snippet Data Layer

▼ Snippets

✚ Add a new snippet

{/} elevar-body-end.liquid

{/} elevar-checkout-end.liquid

{/} elevar-head.liquid

● {/} fbTrack.liquid



Step-3: Create the dataLayer-allPages snippet (use exact naming and casing) & copy and paste the below code

Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevator-head**

```
<!--  
Elevator Data Layer V2  
  
This file is automatically updated and should not be edited directly.  
  
https://knowledge.getelevator.com/how-to-customize-data-layer-version-2  
  
Updated: 2023-04-11 09:36:56+00:00  
Version: 3.1.10  
-->  
<!-- Google Tag Manager -->  
<script>  
  window.dataLayer = window.dataLayer || [];  
</script>  
<script>  
(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({"gtm.start":  
  new Date().getTime(),event:"gtm.js"});var f=d.getElementsByTagName(s)[0],  
  j=d.createElement(s),dl=l!="dataLayer"?"&l="+l:"";j.async=true;j.src=  
  "https://www.googletagmanager.com/gtm.js?id="+i+dl;f.parentNode.insertBefore(j,f);  
})(window,document,"script","dataLayer","GTM-XXXXXX");  
</script>  
<!-- End Google Tag Manager -->  
<script id="elevator-gtm-suite-config" type="application/json">{"gtm_id": "GTM-XXXXX",  
  "event_config": {"cart_reconcile": true, "cart_view": true, "checkout_complete": true,  
  "checkout_step": true, "collection_view": true, "defers_collection_loading": false,  
  "defers_search_results_loading": false, "product_add_to_cart": false,  
  "product_add_to_cart_ajax": true, "product_remove_from_cart": true,  
  "product_select": true, "product_view": true, "search_results_view": true, "user": true,  
  "save_order_notes": true}, "gtm_suite_script":  
  "https://shopify-gtm-suite.getelevator.com/shops/d0b0038eba47fc05bb116d997e301563  
  dd42b6d5/3.1.10/gtm-suite.js", "consent_enabled": true, "apex_domain": null}</script>
```

Step-4: Create the dataLayer-allPages snippet (use exact naming and casing) & copy and paste the below code

Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevar-checkout-end**

```
<script>
  (() => {
    const init = () => {
      window.__ElevaDataLayerQueue = [];
      window.__ElevarListenerLoadQueue = [];
      if (!window.dataLayer) window.dataLayer = [];
    }
    init();
    window.__ElevarTransformItem = event => {
      if (typeof window.ElevarTransformFn === "function") {
        try {
          const result = window.ElevarTransformFn(event);
          if (typeof result === "object" && !Array.isArray(result) && result !== null) {
            return result;
          } else {
            console.error("Elevar Data Layer: `window.ElevarTransformFn` returned a value " + "that wasn't an object, so we've treated things as if this " + "function wasn't defined.");
            return event;
          }
        } catch (error) {
          console.error("Elevar Data Layer: `window.ElevarTransformFn` threw an error, so " + "we've treated things as if this function wasn't defined. The " + "exact error is shown below.");
          console.error(error);
          return event;
        }
      } else {
        return event;
      }
    }
    window.ElevarPushToDataLayer = item => {
      const enrichedItem = {
```

```
    event_id: window.crypto.randomUUID(),
    event_time: new Date().toISOString(),
    ...item
  };
  const transformedEnrichedItem = window.__ElevarTransformItem ?
window.__ElevarTransformItem(enrichedItem) : enrichedItem;
  const listenerPayload = {
    raw: enrichedItem,
    transformed: transformedEnrichedItem
  };
  const getListenerNotifyEvent = () => {
    return new CustomEvent("elevar-listener-notify", {
      detail: listenerPayload
    });
  };
  if (transformedEnrichedItem._elevar_internal?.isElevarContextPush) {
    window.__ElevarIsContextSet = true;
    window.__ElevarDataLayerQueue.unshift(transformedEnrichedItem);
    if (window.__ElevarIsListenerListening) {
      window.dispatchEvent(getListenerNotifyEvent());
    } else {
      window.__ElevarListenerLoadQueue.unshift(listenerPayload);
    }
  } else {
    window.__ElevarDataLayerQueue.push(transformedEnrichedItem);
    if (window.__ElevarIsListenerListening) {
      window.dispatchEvent(getListenerNotifyEvent());
    } else {
      window.__ElevarListenerLoadQueue.push(listenerPayload);
    }
  }
  if (window.__ElevarIsContextSet) {
    while (window.__ElevarDataLayerQueue.length > 0) {
      window.dataLayer.push(window.__ElevarDataLayerQueue.shift());
    }
  }
}

const configElement = document.getElementById("elevar-gtm-suite-config");

if (!configElement) {
```

```
console.error("Elevar: DL Config element not found");
return;
}

const config = JSON.parse(configElement.textContent);

const script = document.createElement("script");
script.type = "text/javascript";
script.src = config.gtm_suite_script;
script.async = false;
script.defer = true;

script.onerror = () => {
  console.error("Elevar: DL JS script failed to load");
};
script.onload = async () => {
  if (!window.ElevarGtmSuite) {
    console.error("Elevar: `ElevarGtmSuite` is not defined");
    return;
  }

  const cartData = {
    attributes: {{- cart.attributes | json -}},
    cartTotal: "{{- cart.total_price | times: 0.01 | json -}}",
    currencyCode: {{- cart.currency.iso_code | json -}},
    items: [
      {%- for line_item in cart.items -%}
      {
        {%- if line_item.sku != blank -%}
        id: {{- line_item.sku | json -}},
        {%- else -%}
        id: "{{- line_item.product_id | json -}}",
        {%- endif -%}
        name: {{- line_item.product.title | json -}},
        brand: {{- line_item.vendor | json -}},
        category: {{- line_item.product.type | json -}},
        variant: {{- line_item.variant.title | json -}},
        price: "{{- line_item.final_price | times: 0.01 | json -}}",
        position: {{- forloop.index -}},
        quantity: "{{- line_item.quantity | json -}}",
        productId: "{{- line_item.product_id | json -}}",
      }
    ]
  }
}
```

```
variantId: "{{- line_item.variant_id -}}",
compareAtPrice: "{{- line_item.variant.compare_at_price | times: 0.01 | json -}}",
image: "{{- line_item.image | image_url -}}"
},
{%%- endfor -%}
]
}
;
```

```
await window.ElevarGtmSuite.handlers.cartAttributesReconcile(
  cartData,
  config.event_config.save_order_notes,
  config.consent_enabled,
  config.apex_domain
);

if (config.event_config.user) {
  const data = {
{%%- if checkout -%}
    cartTotal: "{{- checkout.total_price | times: 0.01 | json -}}",
    currencyCode: {{- checkout.currency | json -}},
{%%- else -%}
    cartTotal: "{{- cart.total_price | times: 0.01 | json -}}",
    currencyCode: {{- cart.currency.iso_code | json -}},
{%%- endif -%}
{%%- if customer -%}
    customer: {
      id: "{{- customer.id | json -}}",
      email: {{- customer.email | json -}},
      firstName: {{- customer.first_name | json -}},
      lastName: {{- customer.last_name | json -}},
      phone: {{- customer.phone | json -}},
      city: {{- customer.default_address.city | json -}},
      zip: {{- customer.default_address.zip | json -}},
      address1: {{- customer.default_address.address1 | json -}},
      address2: {{- customer.default_address.address2 | json -}},
      country: {{- customer.default_address.country | json -}},
      province: {{- customer.default_address.province | json -}},
      provinceCode: {{- customer.default_address.province_code | json -}},
      orderCount: "{{- customer.orders_count | json -}}",
      totalSpent: "{{- customer.total_spent | times: 0.01 | json -}}",
```

```
tags: {{- customer.tags | join: ', ' | json -}}
}
{%%- endif -%}
};

window.ElevarGtmSuite.handlers.user(data);
}

{%%- if checkout -%}
const step = Shopify?.Checkout?.step;
const page = Shopify?.Checkout?.page;
const isOnBeginCheckout = step === "contact_information";
const isOnThankYouPage = page?.includes("thank_you");

{%%- if checkout.attributes["_elevar_visitor_info"] == blank -%}
if (config.event_config.save_order_notes && Shopify?.Checkout?.token) {
  if (isOnBeginCheckout) {
    const data = {%%- if checkout -%}
{
  token: Shopify?.Checkout?.token,
  currency: {{- checkout.currency | json -}},
  {%%- if customer -%}
    customer: { email: {{- customer.email | json -}} },
  {%%- else -%}
    customer: {},
  {%%- endif -%}
  line_items: [
    {%%- for line_item in checkout.line_items -%}
    {
      id: {{- line_item.id | json -}},
      quantity: {{- line_item.quantity | json -}},
      title: {{- line_item.title | json -}},
      variant: {
        id: {{- line_item.variant.id | json -}},
        price: {{- line_item.variant.price | json -}},
        sku: {{- line_item.variant.sku | json -}},
        title: {{- line_item.variant.title | json -}},
        product: {
          id: {{- line_item.variant.product.id | json -}},
          title: {{- line_item.variant.product.title | json -}},
          vendor: {{- line_item.variant.product.vendor | json -}}
        }
      }
    }
  ]
}
    
```



```
    }
  },
  {%%- endfor -%}
],
{%%- if checkout.shipping_address -%}
  shipping_address: {
    city: {{- checkout.shipping_address.city | json -}},
    country: {{- checkout.shipping_address.country | json -}},
    country_code: {{- checkout.shipping_address.country_code | json -}},
    phone: {{- checkout.shipping_address.phone | json -}},
    province: {{- checkout.shipping_address.province | json -}},
    province_code: {{- checkout.shipping_address.province_code | json -}}
  },
{%%- else -%}
  shipping_address: {},
{%%- endif -%}
  shipping_price: {{- checkout.shipping_price | json -}},
  subtotal_price: {{- checkout.order.subtotal_price | json -}},
  total_price: {{- checkout.total_price | json -}},
  total_tax: {{- checkout.tax_price | json -}}
}
{%%- endif -%}
;

    window.ElevarGtmSuite.handlers.storeOrderNotesOffsite(data);
  } else {
    const data = {%%- if checkout -%}
    {
      checkoutId: {{- checkout.id | json -}}
    }
    {%%- endif -%}
    ;

    window.ElevarGtmSuite.handlers.storeOrderNotesOffsite(data);
  }
}
{%%- endif -%}
{% if request.path contains "thank_you" %}
if (config.event_config.checkout_complete && isOnThankYouPage) {
  const data = {%%- if checkout -%}
  {
    currencyCode: {{- checkout.currency | json -}},
    actionField: {
```

```
{%- if checkout.order_id -%}
  id: {{- checkout.order_id | json -}},
{%- else -%}
  id: {{- checkout.id | json -}},
{%- endif -%}
{%- if checkout.order_name -%}
  order_name: {{- checkout.order_name | json -}},
{%- endif -%}
affiliation: {{- shop.name | json -}},
revenue: "{{- checkout.total_price | times: 0.01 | json -}}",
tax: "{{- checkout.tax_price | times: 0.01 | json -}}",
shipping: "{{- checkout.shipping_price | times: 0.01 | json -}}",
{% if checkout.discount_applications %}
  coupon: {{ checkout.discount_applications[0].title | json }},
{% endif %}
{%- if order.subtotal_price -%}
  subTotal: "{{- order.subtotal_price | times: 0.01 | json -}}",
{%- elsif checkout.subtotal_price -%}
  subTotal: "{{- checkout.subtotal_price | times: 0.01 | json -}}",
{%- endif -%}
productSubTotal: "{{- checkout.line_items_subtotal_price | times: 0.01 | json -}}",
discountAmount: "{{- checkout.discounts_amount | times: 0.01 | json -}}",
},
{%- if checkout.customer -%}
  customer: {
    id: "{{- checkout.customer.id | json -}}",
    email: {{- checkout.email | json -}},
    firstName: {{- checkout.billing_address.first_name | json -}},
    lastName: {{- checkout.billing_address.last_name | json -}},
    {%- if checkout.customer.phone -%}
      phone: {{- checkout.customer.phone | json -}},
    {%- elsif checkout.billing_address.phone -%}
      phone: {{- checkout.billing_address.phone | json -}},
    {%- else -%}
      phone: {{- checkout.shipping_address.phone | json -}},
    {%- endif -%}
    city: {{- checkout.billing_address.city | json -}},
    zip: {{- checkout.billing_address.zip | json -}},
    address1: {{- checkout.billing_address.address1 | json -}},
    address2: {{- checkout.billing_address.address2 | json -}},
    country: {{- checkout.billing_address.country | json -}},
```

```
    province: {{- checkout.billing_address.province | json -}},
    provinceCode: {{- checkout.billing_address.province_code | json -}},
    orderCount: "{{- checkout.customer.orders_count | json -}}",
    totalSpent: "{{- checkout.customer.total_spent | times: 0.01 | json -}}",
    tags: {{- checkout.customer.tags | json -}}
  },
  {%- endif -%}
  items: [
    {%- for line_item in checkout.line_items -%}
    {
      id: {{- line_item.sku | json -}},
      name: {{- line_item.product.title | json -}},
      brand: {{- line_item.vendor | json -}},
      category: {{- line_item.product.type | json -}},
      variant: {{- line_item.variant.title | json -}},
      price: "{{- line_item.final_price | times: 0.01 | json -}}",
      quantity: "{{- line_item.quantity | json -}}",
      productId: "{{- line_item.product_id | json -}}",
      variantId: "{{- line_item.variant_id -}}",
      image: "{{- line_item.image | image_url -}}"
    },
    {%- endfor -%}
  ],
  landingSite: {{- checkout.landing_site | json -}}
}
{%- endif -%}
;

    window.ElevaGtmSuite.handlers.checkoutComplete(data);
  }
  {%- endif -%}

  if (Shopify?.Checkout && !isOnThankYouPage) {
    const data = {%- if checkout -%}
    {
      currencyCode: {{- checkout.currency | json -}},
      items: [
        {%- for line_item in checkout.line_items -%}
        {
          id: {{- line_item.sku | json -}},
          name: {{- line_item.product.title | json -}},
          brand: {{- line_item.vendor | json -}},
```

```
category: {{- line_item.product.type | json -}},
variant: {{- line_item.variant.title | json -}},
price: "{{- line_item.final_price | times: 0.01 | json -}}",
quantity: "{{- line_item.quantity | json -}}",
productId: "{{- line_item.product_id | json -}}",
variantId: "{{- line_item.variant_id -}}",
compareAtPrice: "{{- line_item.variant.compare_at_price | times: 0.01 | json -}}",
image: "{{- line_item.image | image_url -}}"
},
{%%- endfor -%}
]
}
{%%- endif -%}
;

if (config.event_config.cart_reconcile) {
  window.ElevarGtmSuite.handlers.cartItemsReconcile(data);
}
if (config.event_config.checkout_step) {
  window.ElevarGtmSuite.handlers.checkoutStep(data);
}
}
{%%- endif -%}
};

document.head.appendChild(script);
})();
</script>
<!-- Google Tag Manager (noscript) -->
<noscript>
  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XXXXXXX"
height="0" width="0" style="display:none;visibility:hidden"></iframe>
</noscript>
<!-- End Google Tag Manager (noscript) -->
```

Step-05: Create the dataLayer-allPages snippet (use exact naming and casing) & copy and paste the below code

Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevar-body-end**

```
<script>
(() => {
  const init = () => {
    window.__ElevaDataLayerQueue = [];
    window.__ElevarListenerLoadQueue = [];
    if (!window.dataLayer) window.dataLayer = [];
  }
  init();
  window.__ElevarTransformItem = event => {
    if (typeof window.ElevarTransformFn === "function") {
      try {
        const result = window.ElevarTransformFn(event);
        if (typeof result === "object" && !Array.isArray(result) && result !== null) {
          return result;
        } else {
          console.error("Elevar Data Layer: `window.ElevarTransformFn` returned a value " + "that wasn't an object, so we've treated things as if this " + "function wasn't defined.");
          return event;
        }
      } catch (error) {
        console.error("Elevar Data Layer: `window.ElevarTransformFn` threw an error, so " + "we've treated things as if this function wasn't defined. The " + "exact error is shown below.");
        console.error(error);
        return event;
      }
    } else {
      return event;
    }
  }
  window.ElevarPushToDataLayer = item => {
    const enrichedItem = {
```

```
    event_id: window.crypto.randomUUID(),
    event_time: new Date().toISOString(),
    ...item
  };
  const transformedEnrichedItem = window.__ElevarTransformItem ?
window.__ElevarTransformItem(enrichedItem) : enrichedItem;
  const listenerPayload = {
    raw: enrichedItem,
    transformed: transformedEnrichedItem
  };
  const getListenerNotifyEvent = () => {
    return new CustomEvent("elevar-listener-notify", {
      detail: listenerPayload
    });
  };
  if (transformedEnrichedItem._elevar_internal?.isElevarContextPush) {
    window.__ElevarIsContextSet = true;
    window.__ElevarDataLayerQueue.unshift(transformedEnrichedItem);
    if (window.__ElevarIsListenerListening) {
      window.dispatchEvent(getListenerNotifyEvent());
    } else {
      window.__ElevarListenerLoadQueue.unshift(listenerPayload);
    }
  } else {
    window.__ElevarDataLayerQueue.push(transformedEnrichedItem);
    if (window.__ElevarIsListenerListening) {
      window.dispatchEvent(getListenerNotifyEvent());
    } else {
      window.__ElevarListenerLoadQueue.push(listenerPayload);
    }
  }
  if (window.__ElevarIsContextSet) {
    while (window.__ElevarDataLayerQueue.length > 0) {
      window.dataLayer.push(window.__ElevarDataLayerQueue.shift());
    }
  }
}

const configElement = document.getElementById("elevar-gtm-suite-config");

if (!configElement) {
```

```
    console.error("Elevar: DL Config element not found");
    return;
  }

  let initialProductAnchorElements = null;

  const setElements = () => {
    initialProductAnchorElements = document.querySelectorAll(
`a[href*="/products/"]:not(a[href*="/collections/products/"]:not(a[href*="/collections/products/products/"]))`
    );
  };

  if (document.readyState === "loading") {
    document.addEventListener("readystatechange", setElements);
  } else {
    setElements();
  }

  const config = JSON.parse(configElement.textContent);

  const script = document.createElement("script");
  script.type = "text/javascript";
  script.src = config.gtm_suite_script;
  script.async = false;
  script.defer = true;

  script.onerror = () => {
    console.error("Elevar: DL JS script failed to load");
  };
  script.onload = async () => {
    if (!window.ElevarGtmSuite) {
      console.error("Elevar: `ElevarGtmSuite` is not defined");
      return;
    }

    const cartData = {
      attributes: {{- cart.attributes | json -}},
      cartTotal: "{{- cart.total_price | times: 0.01 | json -}}",
      currencyCode: {{- cart.currency.iso_code | json -}},
```

```
items: [
  {%- for line_item in cart.items -%}
  {
    {%- if line_item.sku != blank -%}
    id: {{- line_item.sku | json -}},
    {%- else -%}
    id: "{{- line_item.product_id | json -}}",
    {%- endif -%}
    name: {{- line_item.product.title | json -}},
    brand: {{- line_item.vendor | json -}},
    category: {{- line_item.product.type | json -}},
    variant: {{- line_item.variant.title | json -}},
    price: "{{- line_item.final_price | times: 0.01 | json -}}",
    position: {{- forloop.index -}},
    quantity: "{{- line_item.quantity | json -}}",
    productId: "{{- line_item.product_id | json -}}",
    variantId: "{{- line_item.variant_id -}}",
    compareAtPrice: "{{- line_item.variant.compare_at_price | times: 0.01 | json -}}",
    image: "{{- line_item.image | image_url -}}"
  },
  {%- endfor -%}
]
}
;
```

```
await window.ElevarGtmSuite.handlers.cartAttributesReconcile(
  cartData,
  config.event_config.save_order_notes,
  config.consent_enabled,
  config.apex_domain
);

if (config.event_config.user) {
  const data = {
    {%- if checkout -%}
    cartTotal: "{{- checkout.total_price | times: 0.01 | json -}}",
    currencyCode: {{- checkout.currency | json -}},
    {%- else -%}
    cartTotal: "{{- cart.total_price | times: 0.01 | json -}}",
    currencyCode: {{- cart.currency.iso_code | json -}},
    {%- endif -%}
  }
```



```
{%- if customer -%}
customer: {
  id: "{{- customer.id | json -}}",
  email: "{{- customer.email | json -}}",
  firstName: "{{- customer.first_name | json -}}",
  lastName: "{{- customer.last_name | json -}}",
  phone: "{{- customer.phone | json -}}",
  city: "{{- customer.default_address.city | json -}}",
  zip: "{{- customer.default_address.zip | json -}}",
  address1: "{{- customer.default_address.address1 | json -}}",
  address2: "{{- customer.default_address.address2 | json -}}",
  country: "{{- customer.default_address.country | json -}}",
  province: "{{- customer.default_address.province | json -}}",
  provinceCode: "{{- customer.default_address.province_code | json -}}",
  orderCount: "{{- customer.orders_count | json -}}",
  totalSpent: "{{- customer.total_spent | times: 0.01 | json -}}",
  tags: "{{- customer.tags | join: ' ' | json -}}"
}
{%- endif -%}
};

window.ElevarGtmSuite.handlers.user(data);
}

if (config.event_config.product_add_to_cart_ajax) {
  window.ElevarGtmSuite.handlers.productAddToCartAjax(
    config.event_config.save_order_notes,
    config.consent_enabled,
    config.apex_domain
  );
}

if (config.event_config.cart_reconcile) {
  window.ElevarGtmSuite.handlers.cartItemsReconcile(cartData);
}

{%- if request.page_type == "cart" -%}
if (config.event_config.cart_view) {
  window.ElevarGtmSuite.handlers.cartView(cartData);
}
if (config.event_config.product_remove_from_cart) {
  window.ElevarGtmSuite.handlers.productRemoveFromCart(cartData);
}
```

```
}
{%%- endif -%}

{%%- if collection and template.name == "collection" -%}
const data = {%%- if collection and template.name == "collection" -%}
{
  currencyCode: {{- cart.currency.iso_code | json -}},
  items: [
    {%%- assign itemIndex = 1 -%}
    {%%- for product in collection.products -%}
    {% assign default_variant = product.selected_or_first_available_variant %}
    {
      {%%- if default_variant.sku != blank -%}
      id: {{- default_variant.sku | json -}},
      {%%- else -%}
      id: "{{- default_variant.id | json -}}",
      {%%- endif -%}
      name: {{- product.title | json -}},
      brand: {{- product.vendor | json -}},
      category: {{- product.type | json -}},
      variant: {{- default_variant.title | json -}},
      price: "{{- product.price | times: 0.01 | json -}}",
      position: {{- itemIndex -}},
      productId: "{{- product.id | json -}}",
      variantId: "{{- default_variant.id | json -}}",
      handle: {{- product.handle | json -}}
    },
    {%%- assign itemIndex = itemIndex | plus: 1 -%}
  ]
  {%%- endfor -%}
}
{%%- endif -%};

await window.ElevarGtmSuite.handlers.collectionView(
  data,
  config.event_config.collection_view,
  config.event_config.defers_collection_loading,
  initialProductAnchorElements
);

if (config.event_config.product_select) {
```

```
    window.ElevarGtmSuite.handlers.productSelect(data);
  }
  {%%- endif -%}

  {%%- if search.performed -%}
  const data = {%%- if search.performed -%}
  {
    currencyCode: {%%- cart.currency.iso_code | json -%},
    items: [
      {%%- assign itemIndex = 1 -%}
      {%%- for product in search.results -%}
      {%%- if product.object_type == 'product' -%}
      {%%- for variant in product.variants -%}
      {
        {%%- if variant.sku != blank -%}
        id: {%%- variant.sku | json -%},
        {%%- else -%}
        id: "{%- variant.id | json -%}",
        {%%- endif -%}
        name: {%%- product.title | json -%},
        brand: {%%- product.vendor | json -%},
        category: {%%- product.type | json -%},
        variant: {%%- variant.title | json -%},
        price: "{%- product.price | times: 0.01 | json -%}",
        position: {%%- itemIndex -%},
        productId: "{%- product.id | json -%}",
        variantId: "{%- variant.id | json -%}",
        handle: {%%- product.handle | json -%}
      },
      {%%- endfor -%}
      {%%- assign itemIndex = itemIndex | plus: 1 -%}
    {%%- endif -%}
    {%%- endfor -%}
  ]
  }
  {%%- endif -%}
  ;

  await window.ElevarGtmSuite.handlers.searchResultsView(
    data,
    config.event_config.search_results_view,
```

```
config.event_config.defers_search_results_loading,
initialProductAnchorElements
);

if (config.event_config.product_select) {
  window.ElevartGtmSuite.handlers.productSelect(data);
}
{%endif-%}

{%if product-%}
const data = {%if product-%}
{
  attributes: {{- cart.attributes | json -}},
  currencyCode: {{- cart.currency.iso_code | json -}},
  {% assign default_variant = product.selected_or_first_available_variant %}
  {%if default_variant != blank-%}
  defaultVariant: {
    {%if default_variant.sku != blank-%}
    id: {{- default_variant.sku | json -}},
    {%else-%}
    id: "{{- default_variant.id | json -}}",
    {%endif-%}
    name: {{- product.title | json -}},
    brand: {{- product.vendor | json -}},
    category: {{- product.type | json -}},
    variant: {{- default_variant.title | json -}},
    price: "{{- default_variant.price | times: 0.01 | json -}}",
    productId: "{{- product.id | json -}}",
    variantId: "{{- default_variant.id | json -}}",
    compareAtPrice: "{{- default_variant.compare_at_price | times: 0.01 | json -}}",
    image: {{- product.featured_image | json -}},
    inventory: "{{- default_variant.inventory_quantity -}}"
  },
  {%else-%}
  defaultVariant: null,
  {%endif-%}
  items: [
    {%for variant in product.variants-%}
    {
      {%if variant.sku != blank-%}
      id: {{- variant.sku | json -}},
```

```
{%- else -%}
  id: "{{- variant.id | json -}}",
{%- endif -%}
name: {{- product.title | json -}},
brand: {{- product.vendor | json -}},
category: {{- product.type | json -}},
variant: {{- variant.title | json -}},
price: "{{- variant.price | times: 0.01 | json -}}",
productId: "{{- product.id | json -}}",
variantId: "{{- variant.id | json -}}",
compareAtPrice: "{{- variant.compare_at_price | times: 0.01 | json -}}",
image: {{- product.featured_image | json -}},
inventory: "{{- variant.inventory_quantity -}}"
},
{%- endfor -%}
]
}
{%- endif -%};

if (config.event_config.product_view) {
  window.ElevarGtmSuite.handlers.productView(data);
}
if (config.event_config.product_add_to_cart) {
  window.ElevarGtmSuite.handlers.productAddToCartForm(
    data,
    config.event_config.save_order_notes,
    config.consent_enabled,
    config.apex_domain
  );
}
{%- endif -%}
};

document.head.appendChild(script);
})();
</script>
<!-- Google Tag Manager (noscript) -->
<noscript>
  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-XXXXXXX"
height="0" width="0" style="display:none;visibility:hidden"></iframe>
</noscript>
```

```
<!-- End Google Tag Manager (noscript) -->
```

Source – Data Layer Listener

To ensure transaction data is available to GTM, you'll need to manually add an additional script to the "Thank You" page via your Shopify admin. This is in addition to the code copied from the data layer settings.

1. Copy the code from the block below:
2. Open your "Checkout Settings" admin page (by clicking [here](#)).
3. Scroll down to the "Order status page" section of the page.
4. Paste the code from above into the end of the "Additional scripts" input box (right below where you pasted the Data Layer code):

Order status page ⓘ
Add tracking scripts and other customizations.

Additional scripts

<!-- Google Tag Manager -->

5. Click "Save".

Step-06: Create the dataLayer-allPages snippet (use exact naming and casing) & copy and paste the below code

Go to Online Store ----> Themes ----> Snippets ----> Add a new snippet

Name: **elevar-head-listener**

```
<script id="elevar-dl-listener-config"
type="application/json">{"data_layer_listener_script":
"https://shopify-gtm-suite.getelevar.com/shops/d0b0038eba47fc05bb116d997e301563
dd42b6d5/3.1.10/events.js", "ss_url": null, "signing_key": null, "myshopify_domain":
null}</script>
<script>
(() => {
  const configElement = document.getElementById("elevar-dl-listener-config");

  if (!configElement) {
    console.error("Elevar: DLL Config element not found");
    return;
  }

  const config = JSON.parse(configElement.textContent);

  const script = document.createElement("script");
  script.type = "text/javascript";
  script.src = config.data_layer_listener_script;
  script.async = false;
  script.defer = true;

  script.onerror = function () {
    console.error("Elevar: DLL JS script failed to load");
  };
  script.onload = function () {
    if (!window.ElevarGtmSuiteListener) {
      console.error("Elevar: `ElevarGtmSuiteListener` is not defined");
      return;
    }

    window.ElevarGtmSuiteListener.handlers.listen({
      ssUrl: config.ss_url,
```

```
    signingKey: config.signing_key,  
    myshopifyDomain: config.myshopify_domain  
  });  
};  
  
document.head.appendChild(script);  
})();  
</script>
```

Now You need to create GTM tags, triggers & variables.

Please [click here](#) to download the full GTM container for GA4.

The screenshot displays the Google Tag Manager (GTM) interface. At the top, the header shows 'Tag Manager' and 'All accounts > The Analytics Team > Elever Test'. A search bar is present. Below the header, the 'Workspace' tab is active, showing 'GTM-WJNSTTB' and 'Workspace Changes: 73'. The interface is divided into several sections:

- Left Sidebar:** Contains navigation links for 'Overview', 'Tags', 'Triggers', 'Variables', 'Folders', and 'Templates'.
- Top Right:** Includes 'Preview' and 'Submit' buttons.
- Main Content Area:**
 - New Tag:** A section with a 'Choose from over 50 tag types' prompt and an 'Add a new tag' button.
 - Description:** A section with a 'Description' label and an 'Edit description' button.
 - Workspace Changes:** A summary table showing 0 Modified, 73 Added, and 0 Deleted changes. A 'Manage workspaces' button is also present.
 - Changes Not Published:** A green box with the text 'Add tags and publish to make your changes live.'

Below these sections is a detailed 'Workspace Changes' table:

Name ↑	Type	Change	Last Edited	User
UPDATE AFTER IMPORTING	Folder	Added	18 hours ago	Imran.morshed@gmail.com
Update After Importing Event - subscribe	Trigger	Added	18 hours ago	Imran.morshed@gmail.com
All Events	Trigger	Added	18 hours ago	Imran.morshed@gmail.com
Debug Mode	Built-in Variable	Enabled	18 hours ago	Imran.morshed@gmail.com

Please [click here](#) to download the GTM container for Facebook conversion.