DIGITAL PILLS

GOOGLE TAG MANAGER

GUIDA AVANZATA

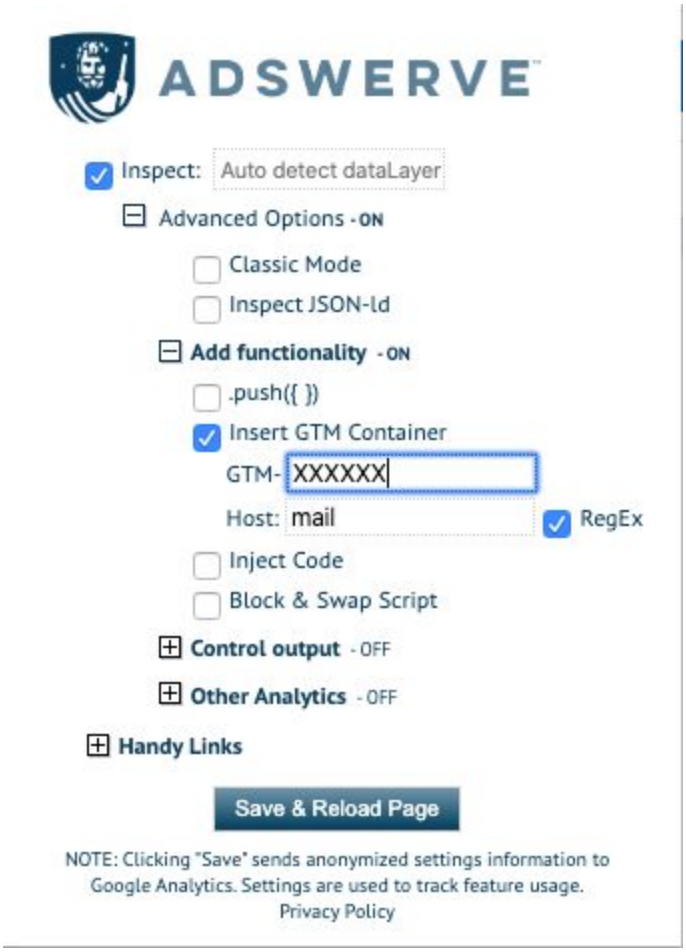
Sommario

Introduzione

Come tutti i Workshop Digital Pills, anche il Worshop su GTM avanzato è un appuntamento 100% pratico. Consigliamo di seguire il Workshop live, e riprovare successivamente a riprodurre le cose viste sfruttando la registrazione. Il presente ebook serve da supporto e riferimento per i codici condivisi durante la sessione live.

Durante questo Workshop il tutor Digital Pills lavorerà sui siti di Moncler (https://store.moncler.com/it-it/donna/spring-summer/vedi-tutti-capispalla) e Mailchimp (https://mailchimp.com/). Le due aziende prescelte non sono clienti Digital Pills, sono state selezionate in quanto utilizzano tecnologie congeniali al contenuto del Workshop, in particolare **GA enhanced ecommerce** per il sito Moncler e **One trust** per il sito Mailchimp.

A scopo didattico verrà inserito sui siti il contanier GTM sandbox dedicato al corso. Consigliamo di riprovare a casa sfruttando il plugin Chrome / Brave Adwserve scaricabile a questo link. Per sfruttare l'inject utilizzare questa configurazione, dove XXXXXX deve essere sostituito con l'id del vostro container di GTM, e nel campo host quello del sito target.



Come prima operazione, vediamo che usando il comando in console posso vedere il valore di un dato elemento di dataLayer, nel comando ricordarsi di sostituire il valore evidenziato con l'id del proprio container >>

google_tag_manager["GTM-NFNHG7N"].dataLayer.get('pageCategory')

```
google_tag_manager["GTM-NFNHG7N"].dataLayer.get('ecommerce.impressions.0.name')
"leave no trace"
```

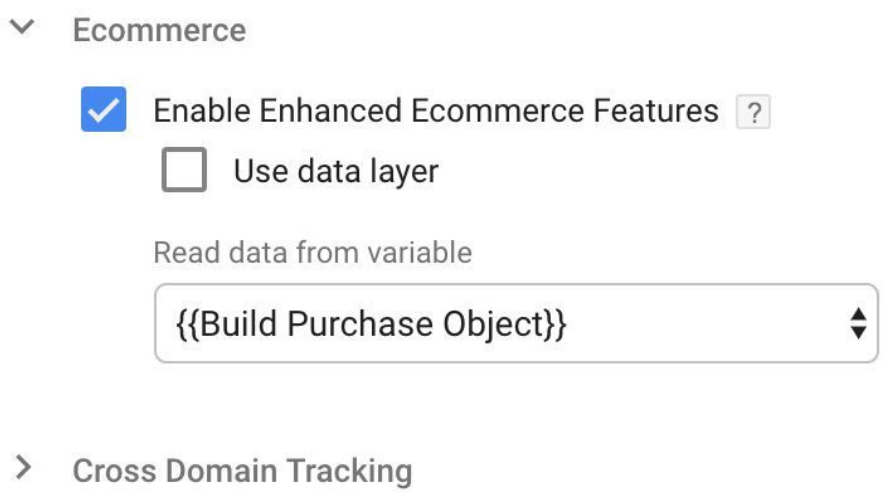
google_tag_manager["GTM-NFNHG7N"].dataLayer.get('ecommerce.impressions.0.name')

```
google_tag_manager["GTM-NFNHG7N"].dataLayer.get('ecommerce.impressions.0')
▸ {name: "leave no trace", id: "27411029964702147", list: "EU_Sneakers_Men_SS", position: 1, price: "500.00", …}
google_tag_manager["GTM-NFNHG7N"].dataLayer.get('ecommerce.impressions.0.name')
"leave no trace"
```

GA enhanced ecommerce tracking

L'Enhanced Ecommerce di Google Analytics permette di registrare le product impression, promozioni e vendite in modo da raggiungere un livello di dettaglio molto più elevato nelle analisi. Per passare questi dati a Google Analytics abbiamo a disposizione essenzialmente 3 strade:

1. **DOM scraping**. Non raccomandato perché introduce un punto di debolezza nel nostro tracking: se qualcosa cambia nella pagina per test A/B o altre modifiche rischia di rompersi tutto. Oltretutto spesso non trovo tutti i valori che mi servono esposti lato utente, quindi rischio di perdere molte informazioni interessanti.
2. **Custom JS variable**. Questo metodo permette di mantenere flessibilità se si usano diversi sistemi di tracking allo stesso tempo, come ad esempio GA e Facebook Analytics. Il dataLayer che richiede GA come vedremo dopo è molto rigido: possiamo andare a scrivere una custom JS variable per andare a prendere i valori formattati in un altro modo e metterli a posto prima di passarli a GA.



3. **dataLayer (metodo raccomandato)**. Con questo modo possiamo sfruttare i push inviati direttamente dal codice e avere maggiore affidabilità. Consigliamo di partire con questo metodo, tutto quello che dovremo fare è spuntare la spunta all'interno di GTM e assicurarci che i dati arrivino nel modo corretto in dataLayer (cosa non scontata). Molti plugin per WP come DuracellTomi su questo punto sono deboli o non funzionano e richiedono quindi customizzazioni.

Per implementare correttamente dobbiamo inviare almeno questi **7 passaggi chiave** a GA:

1. Product impressions
2. Product clicks
3. Product detail views
4. Add to cart
5. Remove from cart
6. Checkout steps
7. Transactions

Dettagli dei dataLayers.push

1 - dataLayer.push eec.productImpression

Il dataLayer eec.productImpression viene inviato ogni volta che dei prodotti sono mostrati all'utente all'interno di una lista. Definire in modo intelligente la granularità desiderata delle liste è uno dei compiti dei web analyst.

```
event: "eec.productImpression",
      ecommerce: {
            currencyCode: "EUR",
            impressions: [
                  {
                        name: "Nome prodotto",
                        id: "54787545748",
                        brand: "The Brand",
                        price: "459",
                        variant: undefined,
                        category: "Product category",
                        position: 0,
                        list: "Home page"
                  }
```

2 - dataLayer.push eec.productClick

Il dataLayer `eec.productClick` viene inviato ogni volta che un prodotto mostrato in una lista viene cliccato dall'utente. Prestare particolare attenzione all'actionField che permette di attribuire correttamente le liste.

```
event: "eec.productClick",
      ecommerce: {
            click: {
                  actionField: {
                        list: "List name"
                  },
                  products: [
                        {
                              name: "Nome prodotto",
                              id: "5478578548540",
                              price: "459",
                              brand: "The Brand",
                              category: "Product category",
                              variant: undefined,
                              position: 1
                        }
                  ]
            }
      }
```

3 - dataLayer.push eec.productDetail

Il dataLayer `eec.productDetail` viene inviato ogni volta che un utente visualizza i dettagli di un determinato prodotto. Di seguito un esempio di key / value contenuti in questo dataLayer.

```
event: "eec.productDetail",
      ecommerce: {
            detail: {
                  products: [
                        {
                              name: "Nome prodotto",
                              id: "5478578548540",
                              price: "459",
                              brand: "The Brand",
                              category: "Product category",
                              variant: undefined
                        }
```

```
                ]
            }
        }
```

4/5 - dataLayer.push eec.addToCart & removeFromCart

Il dataLayer `eec.addToCart` e `eec.removeFromCart` viene mandato quando l'utente aggiunge a un prodotto al carrello oppure lo rimuove. is sent when the user adds or removes a product to /from the cart. Di seguito un esempio di key / value contenuti in questo dataLayer.

```
event: "eec.addToCart",
      ecommerce: {
            currencyCode: "EUR",
            add: {
                  actionField: {
                        list: "Product category"
                  },
                  products: [
                        {
                              name: "Nome prodotto",
                              id: "5478578548540",
                              price: "459",
                              brand: "The Brand",
                              category: "Product category",
                              variant: undefined,
                              quantity: 1
                        }
                  ]
            }
      }
```

6 - dataLayer.push eec.checkout

Il dataLayer `eec.checkout` viene mandato ogni volta che un utente vede una delle fasi di checkout: Spedizione, Fatturazione, Pagamento. Di seguito è a disposizione un esempio di key / value contenuti in questo dataLayer.

```
event: "eec.checkout",
     ecommerce: {
          checkout: {
               actionField: {
                    step: 1
               },
               products: [
                    {
                         name: "Nome prodotto",
                         id: "5478578548540",
                         price: "459.0",
                         brand: "The Brand",
                         category: "Product category",
                         variant: undefined,
                         quantity: "1"
                    }
               ]
          }
     }
```

```
event: "eec.checkout",
     ecommerce: {
          checkout: {
               actionField: {
                    step: 2
               }
          }
     }
```

```
event: "eec.checkout",
     ecommerce: {
          checkout: {
               actionField: {
                    step: 3
               }
          }
     }
```

7 - dataLayer.push eec.purchase

L'ultimo dataLayer è anche il più importante: serve per tracciare correttamente la transazione, quindi particolare attenzione ai vari campi. L'esempio viene direttamente dalla guida di Simo citata sotto.

```
window.dataLayer = window.dataLayer || [];
window.dataLayer.push({
  event: 'eec.purchase',
  ecommerce: {
    currencyCode: 'EUR',
    purchase: {
      actionField: {
        id: 'ORDER12345',
        affiliation: 'Simo\'s shop',
        revenue: '11.00',
        tax: '1.00',
        shipping: '2.00',
        coupon: 'SUMMER2019'
      },
      products: [{
        id: 'product_id',
        name: 'MY PRODUCT',
        category: 'guides/google-tag-manager/enhanced-ecommerce',
        variant: 'Text',
        brand: 'SIMO AHAVA',
        quantity: 2,
        price: '3.50',
        dimension3: 'Ecommerce',
        metric5: 12,
        metric6: 1002
      },{
        id: 'another_product_id',
        name: 'MY ADD-ON',
        price: '1.00',
        quantity: 1,
        coupon: 'ADD-ONS OFF'
      }]
    }
  }
});
```

Configurazione all'interno di GTM

Se il dataLayer è configurato correttamente in tutti gli step, possiamo procedere con l'attivazione all'interno di GTM. Ho a disposizione 2 possibilità:

1. Attivazione tramite la GA setting variable
2. Attivazione tramite gli eventi eec

Documentazione Google
https://developers.google.com/tag-manager/enhanced-ecommerce

Documentazione Simo Ahava
https://www.simoahava.com/analytics/enhanced-ecommerce-guide-for-google-tag-manager/

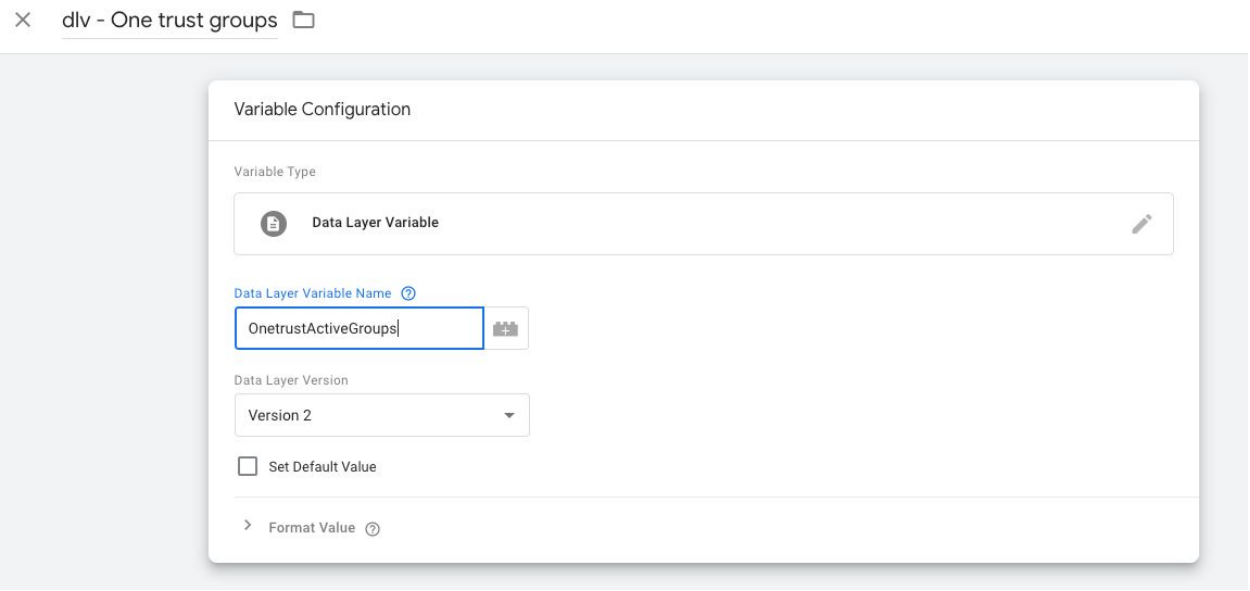
Triggers e blocking triggers

Per lavorare su questo case study cambiamo sito e prendiamo come sandbox il sito di Mailchimp, azienda americana di email marketing (https://mailchimp.com/).

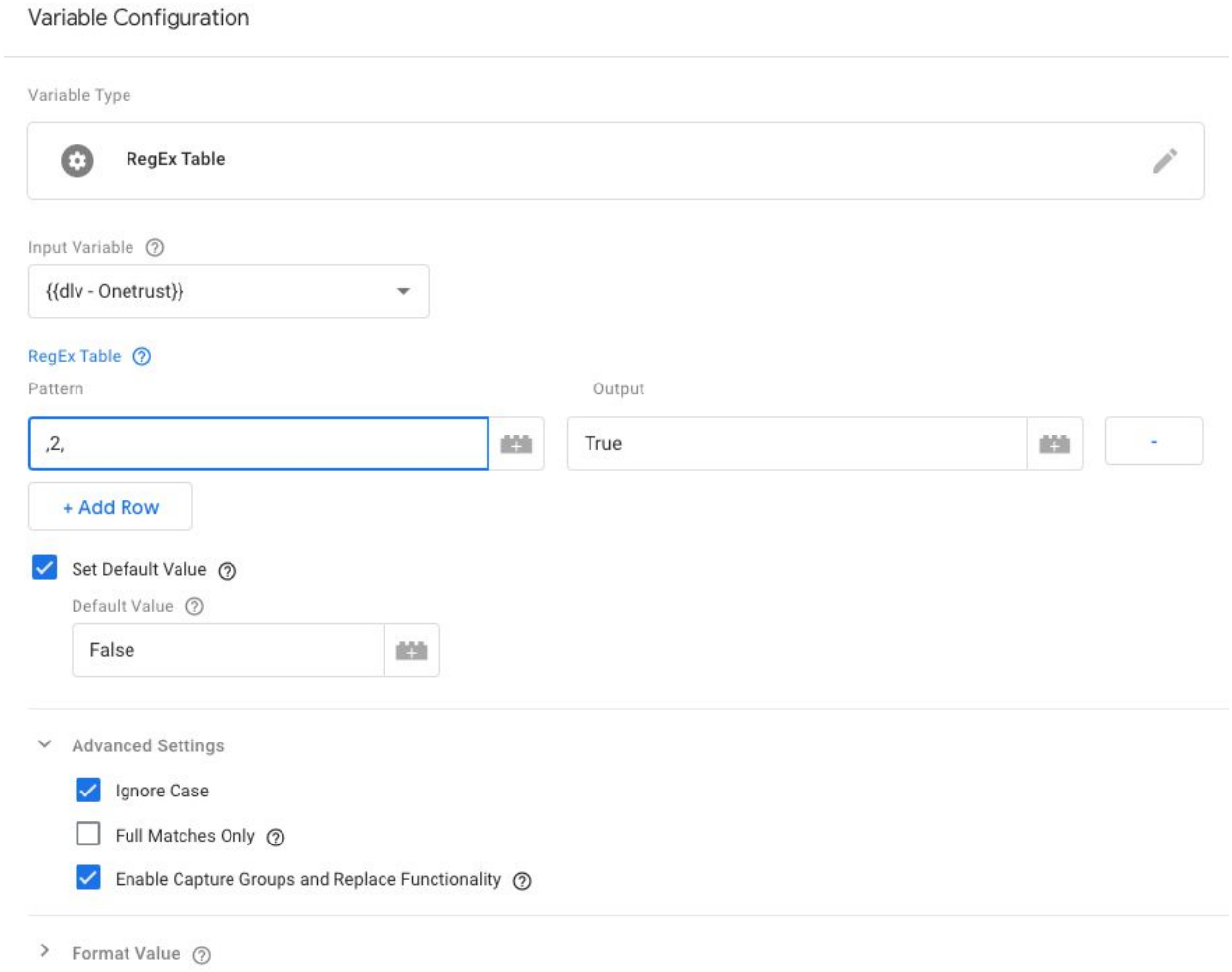
L'argomento che vogliamo affrontare è quello dei blocking trigger applicati alle normative GDPR. Al giorno d'oggi è infatti importante rispettare la volontà dell'utente se vuole o meno essere tracciato. Per gestire le preferenze esistono vari sistemi, tra cui i più famosi sono One Trust e CookieBot. Questi sistemi ci forniscono l'interfaccia per la gestione delle preferenze, ma noi dobbiamo fare tutto all'interno di GTM. Per fortuna non è così complicato (se sappiamo come fare).

Spesso vediamo un errore base, dove il blocking trigger viene applicato ad un evento GTM diverso da quello del firing trigger, ma in questo modo non funzionerà mai. Ricordiamoci sempre che i blocking trigger per funzionare hanno bisogno di essere impostati **sullo stesso evento** del firing trigger. Ecco come consigliamo di procedere.

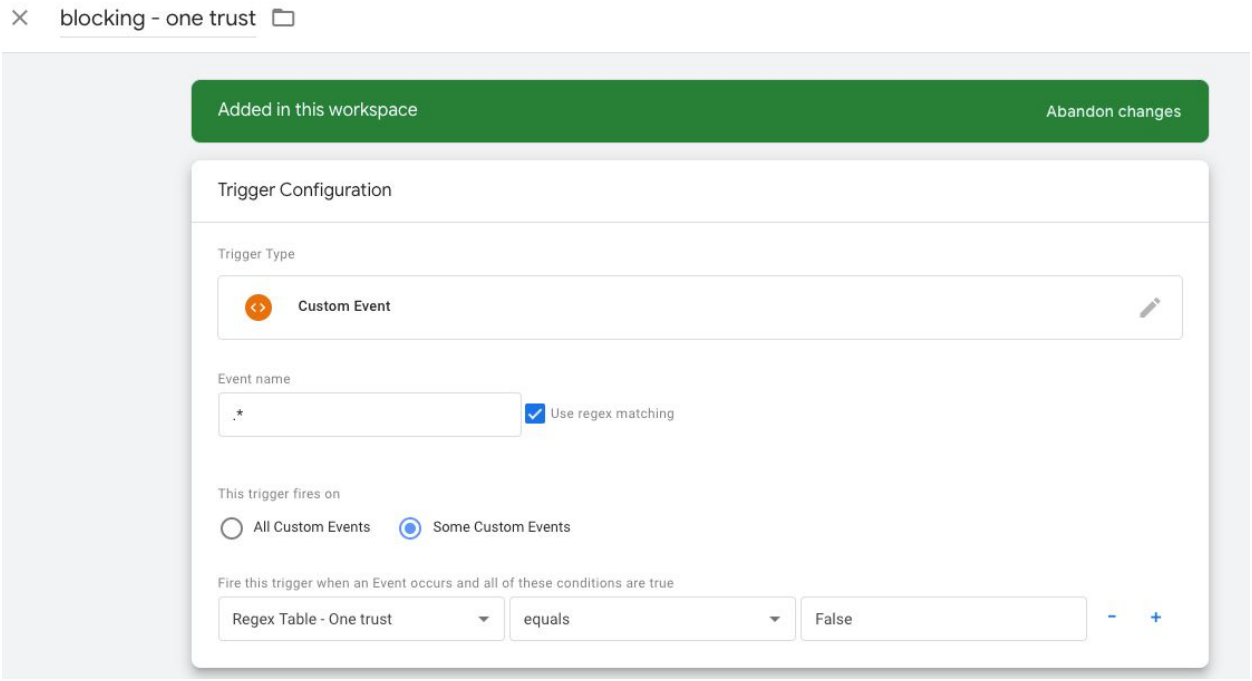
1. Creazione di una dlv che prenda il valore di One trust in dataLayer



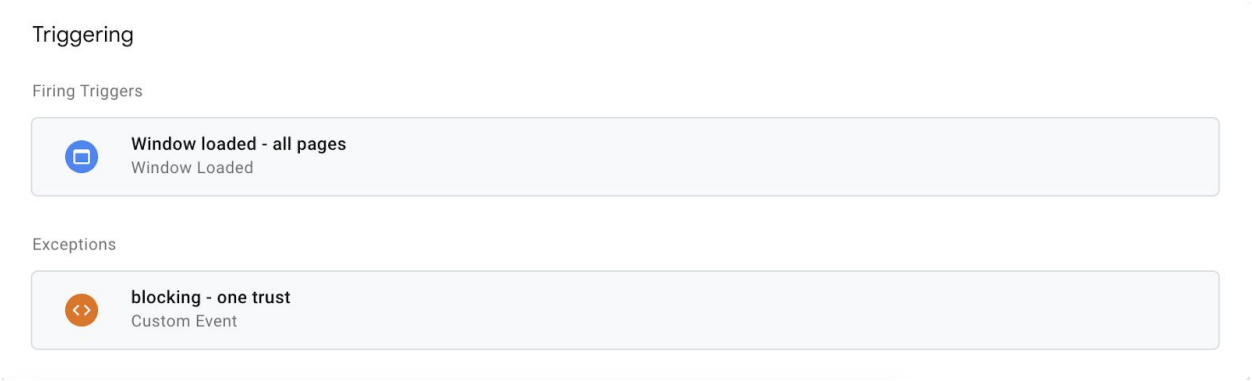
2. Creazione di una regextable che ritorni True quando la variabile contiene 2 (in questo caso i cookie analitici) e False altrimenti.



3. Creazione di custom trigger con regex match in modo che valga su tutti gli eventi.



4. Applicazione del blocking trigger a tutti i tag che mi interessano.



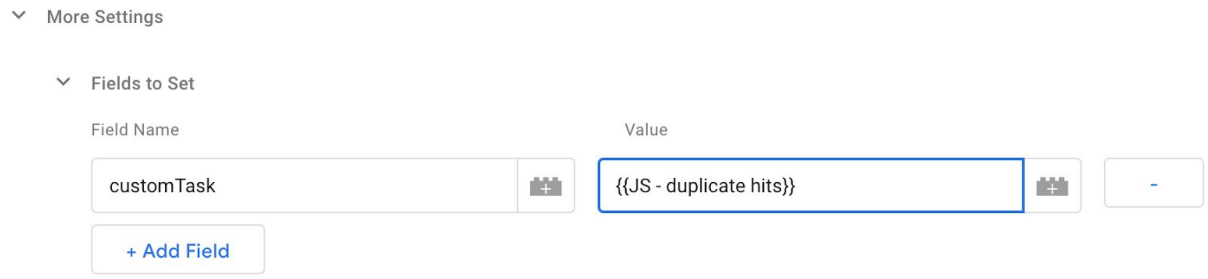
Custom JS variables

Le variabili JS custom offrono numerose possibilità di personalizzazione e potenziamento delle capacità di GTM. Abbiamo già visto un'applicazione pratica nel capitolo dedicato all'ecommerce, dove abbiamo dimostrato come utilizzare una variabile JS custom per popolare il dataLayer ecommerce.

Vediamo adesso due applicazioni pratiche di custom JS variables che potrebbero tornarvi utili nella vita di tutti i giorni.

1. **Duplicazione delle hit** di Google Analytics. Grazie all'unione del custom task e di una variabile JS custom, possiamo duplicare in pochi click tutte le hit che inviamo ad una property analytics su di una seconda. Questo può rivelarsi molto utile se per esempio vogliamo creare una property "Global" che raccoglie i dati di tutti i siti e varie property locali dove invece finiscono i dati - ad esempio - dei singoli paesi.

   Come visto nel workshop base andremo quindi a creare una Lookup table o una regex table per smistare le hit a GA dei singoli paesi, per esempio tramite l'hostname. Successivamente abiliteremo la funzione custom task all'interno dei tag che vogliamo duplicare o direttamente nella GA setting variable se vogliamo duplicarli tutti:



   La variabile che dobbiamo inserire ha questo codice:

```
function() {
  // Ricorda di modificare il valore evidenziato con quello della property verso cui si
vogliono duplicare i dati
  var newTrackingId = 'UA-XXXXX-Y';
  var globalSendTaskName = '_' + newTrackingId + '_originalSendTask';
  return function(customModel) {
```

```
    window[globalSendTaskName] = window[globalSendTaskName] ||
  customModel.get('sendHitTask');
    customModel.set('sendHitTask', function(sendModel) {
      var hitPayload = sendModel.get('hitPayload');
      var trackingId = new RegExp(sendModel.get('trackingId'), 'gi');
      window[globalSendTaskName](sendModel);
      sendModel.set('hitPayload', hitPayload.replace(trackingId, newTrackingId), true);
      window[globalSendTaskName](sendModel);
    });
  };
}
```

2. Creazione di un **content grouping misto**. A volte può capitare che non sia sufficiente creare il content groping all'interno di GA, magari perché la regola di raggruppamento non può fare affidamento solo sugli URL. Ad esempio potrebbe essere utile creare un content grouping che abbiamo un mix di regole URL e CSS selector. Anche in questo caso ci viene in soccorso la custom JS variable.

   Ecco un esempio di codice che potremmo utilizzare.

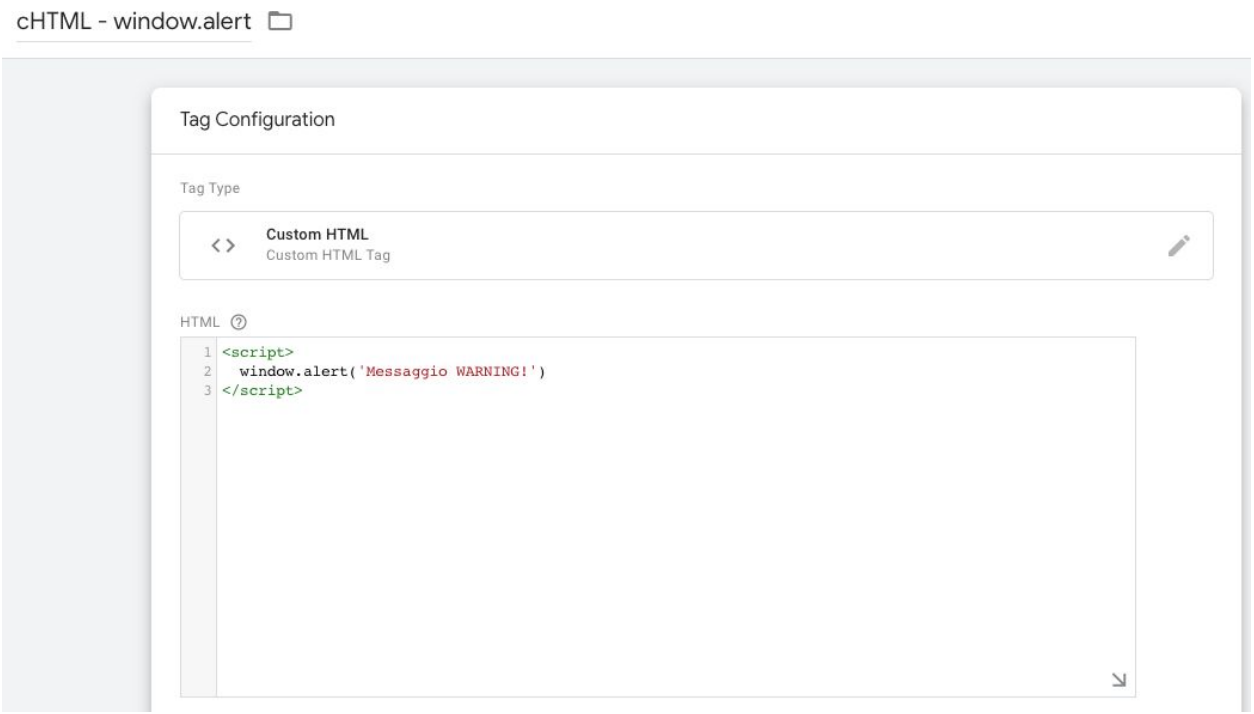
```
function () {
    // Se URL contiene shop allora metto Shop
   if (window.location.href.indexOf("/shop") > 0) {
     return "Shop";
  }
     else if (window.location.href.indexOf("/landing") > 0) {
     return "Landing page";
  }
  else if (document.querySelectorAll(".my-class").length > 0) {
     return "Another page";
  }
  else {
     return "(not set)";
  }
}
```

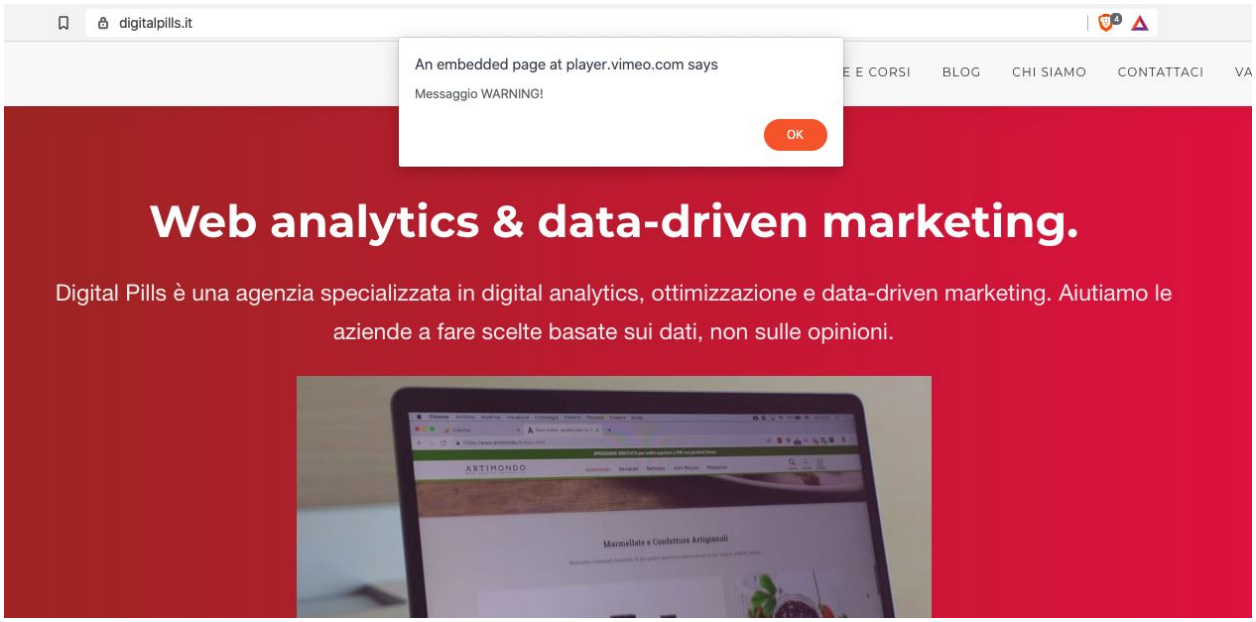
   Per "chiudere il cerchio" mi basterà andare nella GA setting variable ed impostare il content grouping relativo all'indice che mi interessa mettendo come valore quello della variabile appena creata.

Custom HTML e custom template

- Perche non usare i tag custom HTML, riassunto articolo Simo https://www.simoahava.com/analytics/custom-html-tag-guide-google-tag-manager/
- Come creare un custom template base

I tag custom HTML sono una delle funzioni più potenti e abusate di GTM. Cerchiamo di fare un po' di chiarezza. GTM ci permette di iniettare all'interno della pagina del codice HTML custom per fare praticamente quello che vogliamo. Ad esempio potremmo creare un tag di questo tipo per mostrare all'utente un messaggio di avvertimento.





Ma cosa succede “dietro le quinte” quando inserisco il codice dentro GTM? Il codice viene inserito al termine della pagina dopo che questa ha terminato di caricare. Ogni volta che un tag viene aggiunto il browser è costretto a ricalcolare tutte le dimensioni, proporzioni, etc rispetto a quell’elemento, con grande impatto negativo sulla performance della pagina (per approfondimenti consigliamo questo articolo di Simo).

Per questo è importante ricordare che **i custom HTML tag sono da evitare** a meno che strettamente necessari. In particolare non si deve evitare di utilizzare GTM come un sistema CMS, quindi per gestire operazioni come:

- Resizing e posizionamento di elementi in pagina;
- Inserimento del codice di sistemi di gestione delle preferenze cookie (Cookiebot, One Trust, etc);
- Modificare la UX/UI;
- Inserire tag di venditori terzi di pubblicità come FB e simili.

Per questo esistono i **custom template**, di cui adesso vedremo un’applicazione pratica.

Creazione di un custom template di base

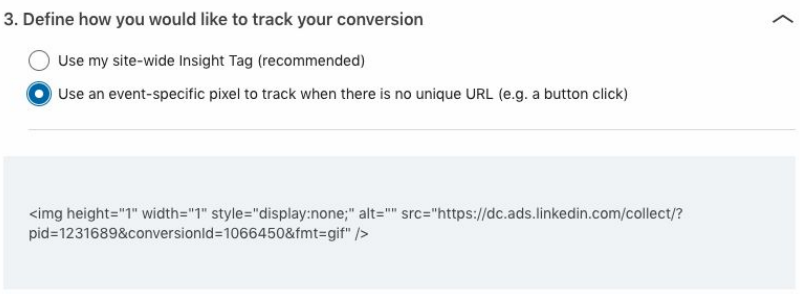
I custom template si compongono di 4 componenti fondamentali, che devono essere configurate quando si crea un nuovo template:

1. Impostazioni di base (details), come il nome, logo, descrizione.

2. L'interfaccia utente, che permette di impostare come il nostro utente finale vedrà il tag all'interno di GTM.
3. L'editor di codice, per inserire il vero e proprio codice del tag.
4. I permessi, dove si imposta cosa può fare il nostro tag.

Nel tutorial di oggi andremo a creare un custom template per Linkedin ads, ma se siete degli sviluppatori sicuramente avrete già in mente decine di applicazioni pratiche!

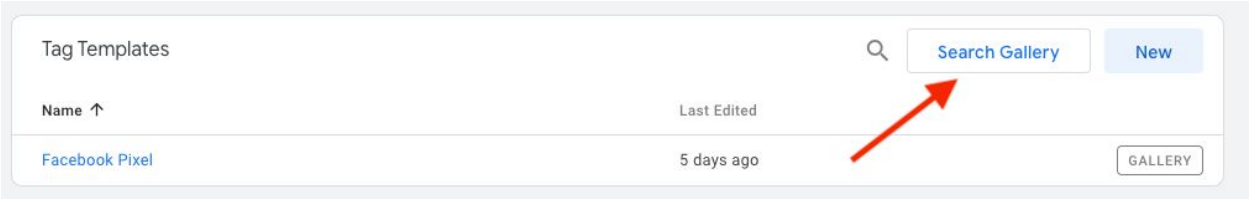
Potrebbe tornarci utile questo cutom template perché tra i tag nativi di GTM troviamo soltanto il pixel di base di Linkedin (assimilabile alla pageview di GA), ma per vari motivi potrebbe esserci utile inserire delle conversioni evento (ad esempio sparate al click di un bottone, al termine di un video, etc)...



Per il tutorial passo-passo rimandiamo alla registrazione del webinar che abbiamo inviato, dove vengono mostrati tutti i passaggi per crearlo. A questo link può essere scaricato il template sviluppato da Kevin Haag.

Utilizzo di custom template della gallery

Se non siamo sviluppatori o semplicemente vogliamo utilizzare i tag messi a disposizione da altri, possiamo importare all'interno del nostro container i template che ci interessano, cliccando sul tasto "Search Gallery" nell'area dedicata ai custom template.



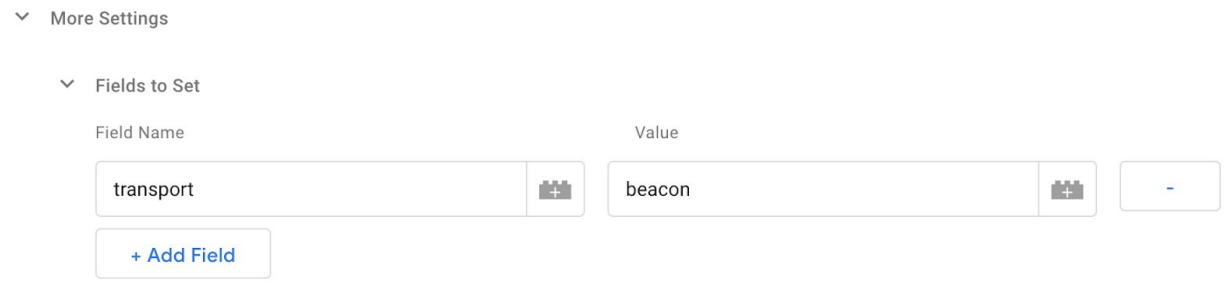
4 trucchetti finali

1. Ricorda di **restringere lo scope** delle tue variabili JS inserendo una funzione anonima nei tuoi tag HTML. Per evitare di inquinare il *global namespace* ricorda di inserire i tuoi tag HTML all'interno di una funzione anonima, in modo da non doverti preoccupare di causare conflitti!

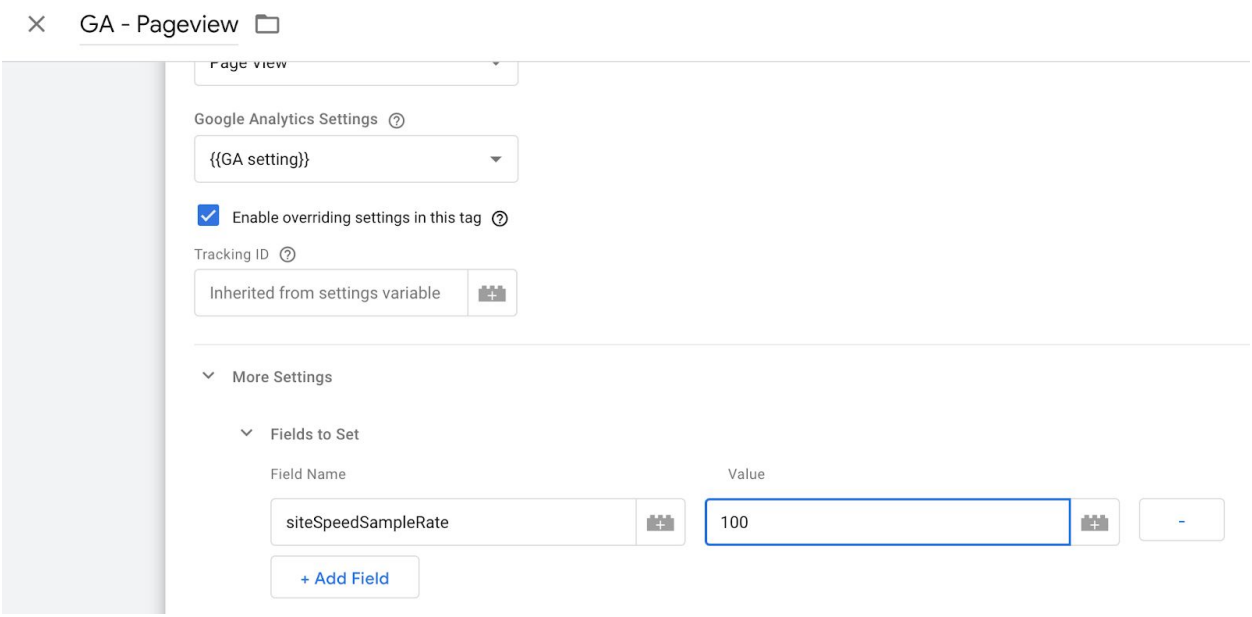
2. **Sfrutta il customTask**. Il customTask ci permette di accedere alla hit PRIMA che questa venga costruita, ed e la prima azione che viene processata nella coda di analytics. Link articolo completo di Simo. Questo ci permette di fare cose interessanti come:
    a. Duplicare le hit di GA su un'altra property;
    b. Rimuovere le PII dalle hit di GA;
    c. Impostare il clientId come custom dimension in GA.

   ATTENZIONE: soltanto un customTask viene ammesso per ogni GA setting / task. Se si vuole fare più di una cosa bisogna unirli. Check: https://www.simoahava.com/tools/customtask-builder/

3. Usa il **transport beacon**. Questa impostazione all'interno di GA ci permette di modificare il metodo di trasporto delle hit in modo da non perdere informazioni quando un utente sta lasciando una pagina, ad esempio un evento click su un link. Per farlo basta impostare in questo modo la variabile di GA:



4. Imposta lo **speed sampling**. Non molti sanno che i dati sulla velocità delle pagine presenti nei report site speed all'interno di GA di default prendono in considerazione soltanto l'1% del traffico, troppo poco se hai un sito con poco traffico. Il limite che possiamo raggiungere è di 10k/giorno, e possiamo modificarlo direttamente all'interno di GTM, in questo modo:



Questo è uno dei pochi casi in cui conviene modificare direttamente il tag di pageview di GA invece che la GA setting variable, in modo da evitare duplicati in caso di virtual pageviews.