



# PIXEL SETUP

Installation instructions

Version 4

Contents

- 1 Introduction .....2
- 2 Pixel setup.....2
  - 2.1 Using a Google Tag template .....2
  - 2.2 Setup with Google Analytics 4 Enhanced Ecommerce tracking .....2
  - 2.3 Manual configuration of Google Tag Manager.....3
  - 2.4 Page tracking of single page apps .....4

## 1 Introduction

This document describes how to configure the tracking of clickstream data to the Woony Talk service. The tracking setup presumes the use of Google Tag Manager on the web site.

Before starting the installation, please make sure that you have the GTM (Google Tag Manager) Container implemented on the webpage with ID available. Ensure that Google Tag Manager uses data layer object (optional at setup of Google Tag Manager).

Woony Pixel service uses custom variables in the data layer object.

Note that the eCommerce tracking is based on WooCommerce.

## 2 Pixel setup

Ensure that Google Tag Manager uses data layer object (optional at setup of Google Tag Manager). Woony Pixel service uses custom variables in the data layer object.

The Woony Pixel can be configured in two different ways

- Using a Google Tag template
- Adding a script to your web site and configuring Google Tag manager

### 2.1 Using a Google Tag template

How to install:

1. Download the template file 'Woony Pixel Template' from the customer zone at *woony.me*
2. Import the template file as 'Custom template' to your GTM container
3. Create a new tag based on the template. Name the tag "Woony Pixel".
4. Add the 'Woony Pixel' Tag to your GTM container by selecting the 'Woony Pixel' custom tag template
5. Mark the checkbox 'E-commerce tracking' in order to track cart content and purchase events
6. Set the trigger for the tag to 'All Pages'
7. Fill in the Woony License key (create an account at *woony.me* to get one)
8. In case of e-commerce tracking
  - a. Ensure that Google Analytics Enhanced Ecommerce is enabled for the actual account
  - b. A trigger of type "Click – All Elements" which fires when "Click classes" contains "single\_add\_to\_cart\_button" called "Woony - Add to cart"
9. In case of Single Page Application (SPA) site, add a 'History changed' trigger. See next chapter how to setup tracking of SPA sites.
10. Save and publish changes to complete the configuration

The tracking of clickstream data via Woony Pixel shall now work (after refresh, if page is already open).

### 2.2 Setup with Google Analytics 4 Enhanced Ecommerce tracking

Assure that Google Analytics 4 Enhanced Ecommerce tracking is enabled for your web site (see instructions from Google).

Add a trigger of type "Custom Event" to be used by the Woony Pixel tag. It can for example be called "GA4 Enhanced Ecommerce".

Copy the following to the field “event names”:

```
view_item|view_item_list|select_item|add_to_cart|remove_from_cart|view_cart|begin_checkout|add_payment_info|add_shipping_info|purchase
```

...and configure the trigger to fire on all custom events.

Add the trigger to your Wony Pixel tag (need to use version 1.0.0 and above of the Woony Tag template).

## 2.3 Manual configuration of Google Tag Manager

### 2.3.1 Google Tag Manager configuration

Log in to Google Tag Manager (GTM) and extend the configuration for the GTM container ID for your web site. A new separate Google Tag Manager container might be used for Woony Pixel.

### 2.3.2 Setup without e-commerce tracking

#### 2.3.2.1 Tags

A tag of type “Custom image” with the image URL set to

```
https://pixel.woony.se/?apiKey<license key>&pageUrl={{Page URL}}&vsID={{Woony - Visitor Id}}
```

where the <license key> is the Woony license key for the account. Set the name of the tag to “Woony Pixel”. Set the triggers for the tag to “All Pages”. Ensure that cache busting is enabled with default query parameter.

#### 2.3.2.2 Variables

Add a Custom JavaScript variable called “Woony – Visitor Id” with the following content:

```
function() {
  var visitorId = localStorage.getItem('woony-visitor-id');
  return visitorId;
}
```

### 2.3.3 Setup with e-commerce tracking

#### 2.3.3.1 Tags

A tag of type “Custom image” with the image URL set to

```
https://pixel.woony.se/?apiKey=<license key>&pageUrl={{Page URL}}&vsID={{Woony - Visitor Id}}&item={{Woony - Cart}}
```

where the <license key> is the Woony license key for the account. Set the name of the tag to “Woony Pixel”. Set the triggers for the tag to “All Pages” and “Woony - Add to cart”. Enable cache busting with default query parameter.

#### 2.3.3.2 Variables

Add a Custom JavaScript variable called “Woony – Visitor Id” with the following content:

```
function() {
  var visitorId = localStorage.getItem('woony-visitor-id');
  return visitorId;
}
```

Add a Data Layer variable “cartContent.items” called “GA EE – cartContent.items”. Assure that Google Analytics Enhanced Ecommerce is enabled for the actual account.

Add a Custom JavaScript variable called “Woony – Cart” with the following content:

```
function() {
  var items = {{GA EE - cartContent.items}};
  var i;
  if (!items) return;
  var result = "[";
  for (var i = 0; i < items.length; i++) {
    if (i===0) {

result=result+"\\"i\\":"\""+items[i].id+"\\",\\n\\":"\""+items[i].name+"\\",\\c\\":"\""+items[i].category+"\\"
,\\q\\":"\""+items[i].quantity+"\\",\\p\\":"\""+items[i].price+"\\"";
    } else {

result=result+",{\\"i\\":"\""+items[i].id+"\\",\\n\\":"\""+items[i].name+"\\",\\c\\":"\""+items[i].category+"\\"
,\\q\\":"\""+items[i].quantity+"\\",\\p\\":"\""+items[i].price+"\\"";
    }
  }
  result = result+"];";
  return result;
}
```

### 2.3.3.3 Triggers

A trigger of type “Click – All Elements” which fires when “Click classes” contains “single\_add\_to\_cart\_button” called “Woony - Add to cart”.

## 2.4 Page tracking of single page apps

Samples of single page apps (SPA) are websites built with React.js, Vue.js and Gatsby.js.

There are several ways to add tracking to SPA sites

- Using history change listener with Google Tag Manager.
- Using the built-in page-view tracking of SPA sites offered by the Enhanced measurement in Google Analytics 4.
- Push pageview data into Data Layer when a visitor change page in the app.

The **first** alternative:

1. In Google Tag Manager, crate a trigger of type “History Change” called “History changed”.
2. Add the trigger “Hiistory changed” to the tag “Woony Pixel” created in chapter 3.5. For cache busting, use the string “gtmcb{{Event}}{{Page Path}}{{Page URL}}”

The **second** alternative:

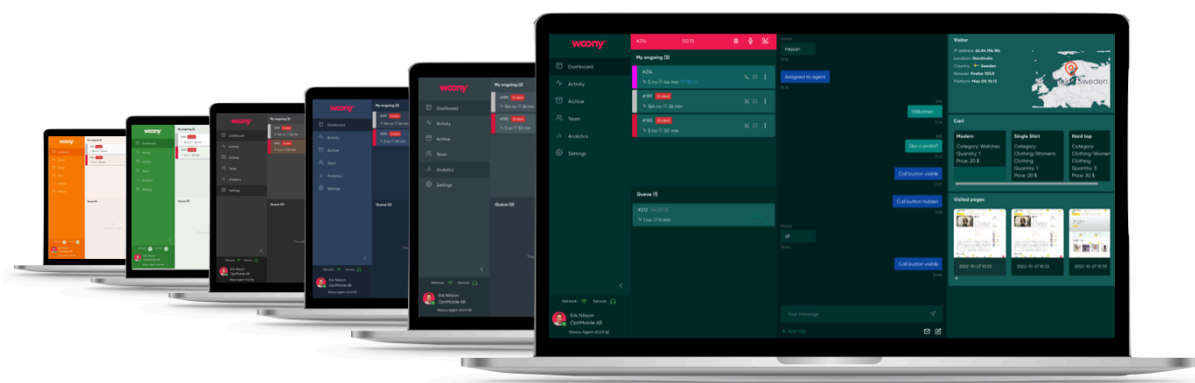
1. In Google Analytics 4 admin panel for the account, select Properties -> Data Streams and click on the web data stream. Make sure that enhanced measurement is enabled.
2. Select “Show advanced settings” and enable the “Page changes based on browse history events” checkbox.
3. Create a trigger in Google Tag Manager of type “Custom Event”. Set event name to “gtm.historyChange-v2” called “GA4 - history changed”.
4. Add the trigger “GA4 – history changed” to the tag “Woony Pixel” created in chapter 3.5. For cache busting, use the string “gtmcb{{Event}}{{Page Path}}{{Page URL}}”

If the URL of the websites contains URL fragments, the default *page\_location* need to be overridden in Google Analytics 4. Create a tag of type “Google Analytics – GA4 Event” for the actual Google Analytics configuration and set the value of parameter *page\_location* to `{{js-window.location.href}}` for the event *page\_view*.

The **third** alternative means adding pageview data to the Data Layer. Please contact us for further details.

---

For more information, please contact us at: [woony@optimobile.se](mailto:woony@optimobile.se)



OPTIMOBILE AB  
MALMSKILLNADSGATAN 32, SE-11151 STOCKHOLM, SWEDEN  
WOONY.SE / OPTIMOBILE.SE