

Яков Осипенков

НАСТРОЙКА СТАНДАРТНОЙ ЭЛЕКТРОННОЙ ТОРГОВЛИ С ПОМОЩЬЮ GTM

ПОДРОБНОЕ РУКОВОДСТВО



Google Tag Manager

Содержание

Начало	3
Типы электронной торговли	4
Какой тип электронной торговли выбрать?	4
Как настроить электронную торговлю?	5
Кто настраивает электронную торговлю?	6
Почему в интернете нет конкретного руководства по настройке?	7
Электронная торговля для Яндекс.Метрики	7
Как настроить стандартную электронную торговлю?	8
Google Tag Manager и DOM Scraping	11
Использование расширения GTM Variable Builder	13
Использование специального тега dataLayer Builder + Standard Ecommerce	20
Что значит dataLayer.push () без события?	21
Способ №1. Код dataLayer.push () добавляется ДО контейнера GTM (на странице «Спасибо»)	23
Способ №2. Код dataLayer.push () добавляется ПОСЛЕ контейнера GTM (на странице «Спасибо»)	23
Способ №3. Код dataLayer активируется на странице с помощью события	24
Итоги	28

Начало

Оригинал статьи расположен по ссылке: <https://osipenkov.ru/standard-ecommerce-gtm/>

Когда речь заходит о настройке нестандартных событий, динамического ремаркетинга, User ID, электронной торговли или других не менее сложных процессов, у интернет-маркетолога возникает целый ряд вопросов. В этой публикации я постараюсь ответить на большую часть из них, чтобы у вас появилось окончательное понимание того, как все же настраивается электронная торговля для Google Analytics и Яндекс.Метрики.

Если вы никогда не слышали об электронной торговле, рекомендую прочитать эту статью: <https://osipenkov.ru/elektronnaya-torgovlya/> Чаще всего e-commerce настраивают для интернет-магазинов, чтобы отслеживать количество транзакций и доход от продажи товаров с привязкой к различным источникам трафика. В этой публикации (<https://osipenkov.ru/ecommerce-landing/>) я описал способ отслеживания и для обычного лендинга.

Типы электронной торговли

В Google Analytics существует два основных типа электронной торговли:

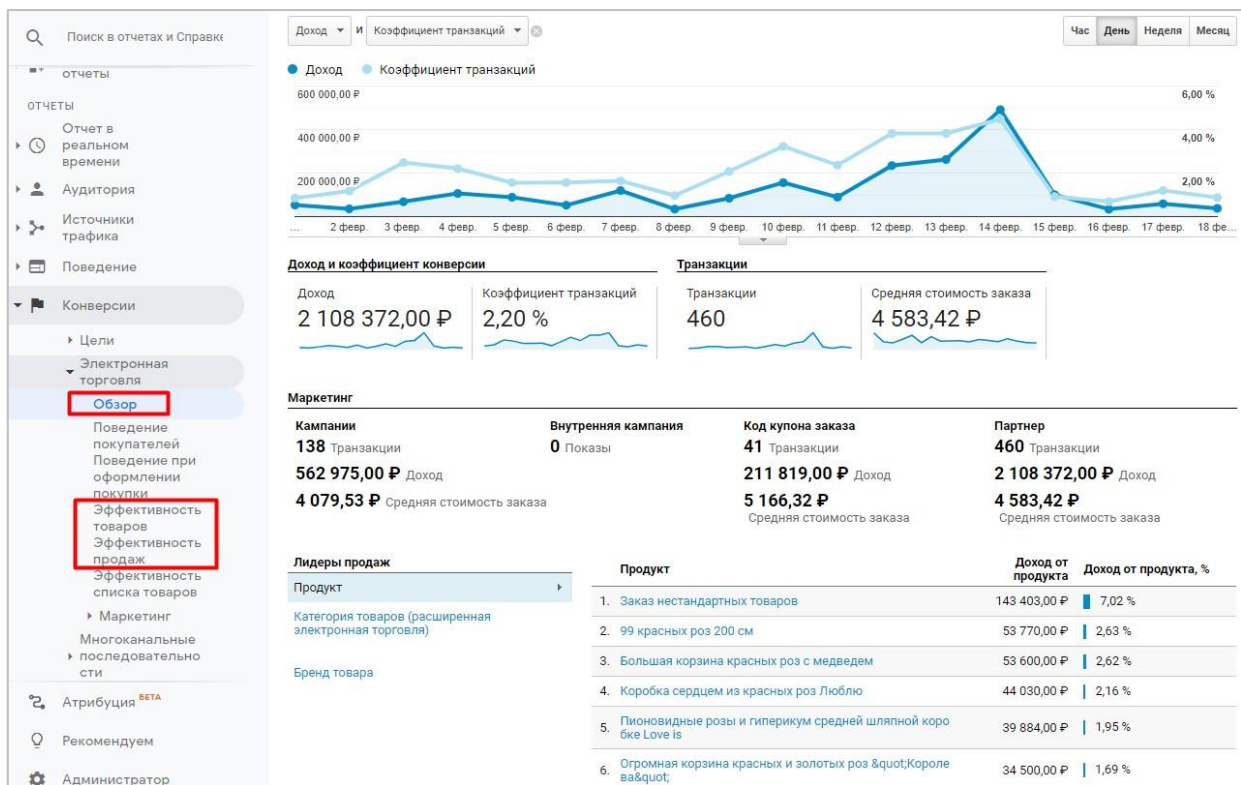
1. **Стандартная электронная торговля (Standard Ecommerce);**
2. **Расширенная электронная торговля (Enhanced Ecommerce, EE).**

Стандартная электронная торговля:

- появилась раньше (в библиотеке *ga.js*), чем расширенная (*analytics.js*);
- ограничена количеством предоставляемых отчетов;
- не такая гибкая по сравнению с EE, но легче в настройке.

Список отчетов, доступных в стандартной версии:

- **Обзор:** данные по основным показателям – доход, транзакции, средняя стоимость заказа, коэффициент транзакции и т.д.
- **Эффективность товаров:** данные по к каждому отдельному товару – доход, количество покупок, средняя цена, идентификатор продукта, категория продукта и т.д.
- **Эффективность продаж:** доход с разбивкой по ID заказа и дополнительным показателям – налоги, стоимость доставки, сумма возврата и т.д.

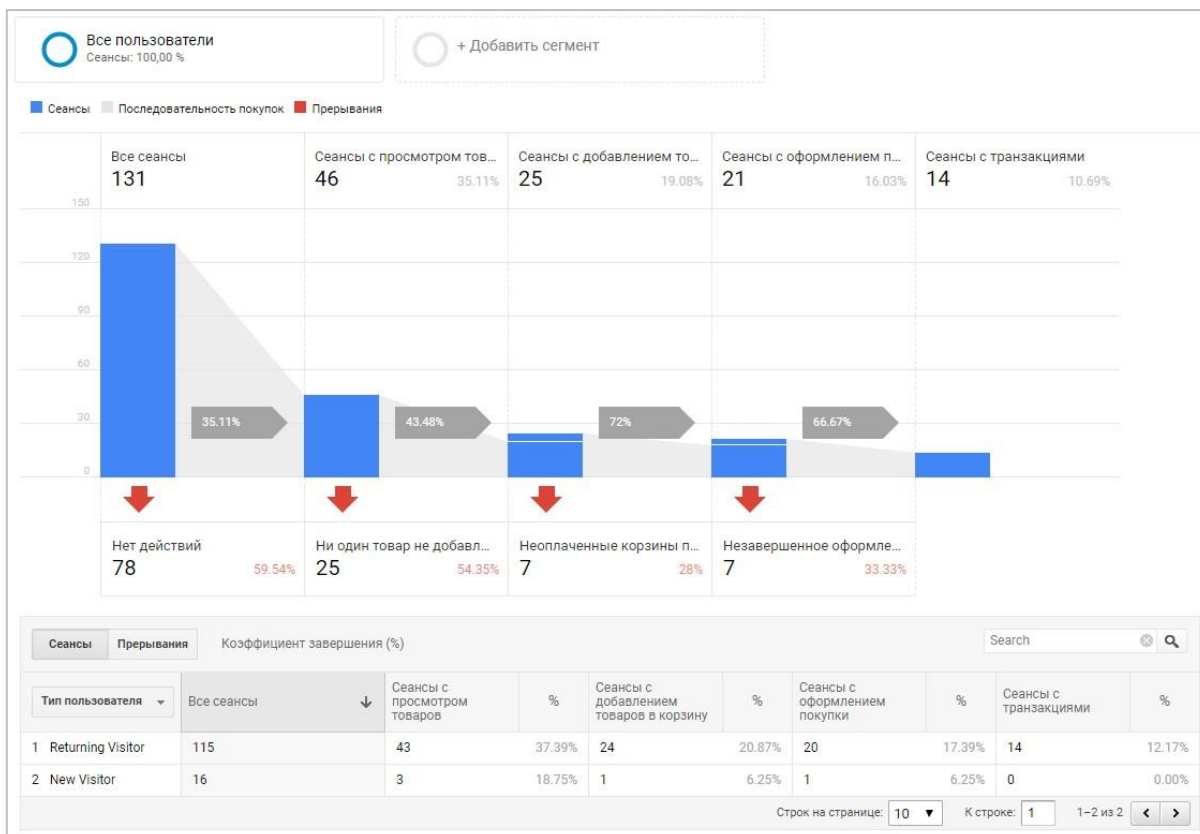


Список доступных отчетов в стандартной электронной торговле

Если для расширенной электронной торговли доступно до 9 уровней отслеживания действий (показы товаров, клики по товарам, показы сведений о товарах, показы промоакций, клики по промоакциям, оформление покупки, покупки, возвраты), то для работы стандартной электронной торговли достаточно настроить **ТОЛЬКО** покупки (транзакции).

Какой тип электронной торговли выбрать?

Если вы хотите видеть только данные о продажах, понимать, насколько эффективно продается каждый товар, какие источники трафика генерируют продажи, то достаточно настроить *стандартную электронную торговлю*. Если вы хотите видеть не только покупки, но и пути пользователей от этапа к этапу на вашем сайте (какое количество пользователей добавило товар в корзину, сколько из них перешло на шаг оформления заказа, а какая часть из этой дошедших совершила покупку), строить по ним сегменты, запускать на них ремаркетинговые кампании, то тогда вам следует выбрать *расширенную электронную торговлю*.



Отчет "Поведение покупателей" (Enhanced Ecommerce)

Внедрить функционал расширенной электронной торговли гораздо сложнее, для настройки требуется больше времени и действий разработчика. Поэтому при выборе типа электронной торговли не стоит забывать и об экономической составляющей.

По сравнению с Enhanced Ecommerce, стандартная электронная торговля не требует такого большого количества времени, чтобы успешно внедрить отслеживание продаж. Поэтому научимся отслеживать именно ее. Пошаговый процесс, с примерами настроек и техническим заданием (ТЗ) для расширенной электронной торговли представлен в моем курсе по веб-аналитике.

Подробнее: <http://edu.osipenkov.ru/web-analytics.html>

Как настроить электронную торговлю?

Для внедрения на сайт стандартной электронной торговли можно использовать два метода настройки:

1. через код отслеживания;
2. с помощью Google Tag Manager.

Ниже будет рассмотрен второй способ (через Google Tag Manager), поскольку он является наиболее простым и **не всегда** требует сторонней помощи.

Кто настраивает электронную торговлю?

Обычно настройку электронной торговли руководство поручает интернет-маркетологу, который ведет и рекламные кампании, и имеет доступы к счетчикам веб-аналитики (Google Analytics и Яндекс.Метрики). Однако это не всегда так. Настройка любого типа электронной торговли подразумевает под собой работу с кодом, который необходимо разместить на странице оформленного заказа (в случае стандартной реализации) или на нескольких (при настройке расширенной электронной торговли). Такие коды вы увидите и в документации Google, и в любой статье в интернете, которая посвящена этой теме.

Вот один из примеров отслеживания транзакций для библиотеки *gtag.js*, событие **purchase** (покупка):

```
gtag('event', 'purchase', {
  "transaction_id": "24.031608523954162",
  "affiliation": "Google online store",
  "value": 23.07,
  "currency": "USD",
  "tax": 1.24,
  "shipping": 0,
  "items": [
    {
      "id": "P12345",
      "name": "Android Warhol T-Shirt",
      "list_name": "Search Results",
      "brand": "Google",
      "category": "Apparel/T-Shirts",
      "variant": "Black",
      "list_position": 1,
      "quantity": 2,
      "price": '2.0'
    },
    {
      "id": "P67890",
      "name": "Flame challenge TShirt",
      "list_name": "Search Results",
      "brand": "MyBrand",
      "category": "Apparel/T-Shirts",
      "variant": "Red",
      "list_position": 2,
      "quantity": 1,
      "price": '3.0'
    }
  ]
});
```

Этот код нельзя просто взять из документации и вставить к себе на сайт. Чтобы данные корректно передавались в Google Analytics, необходимо для каждой переменной (*id*, *name*, *list_name*, *brand*, *price* и т.д.) выводить собственное значение. Причем извлечение данных для интернет-магазинов на различных CMS-платформах может быть реализовано по-разному. Для 1С-Битрикс своя, для OpenCart своя, для WordPress своя и т.д.

Поэтому когда говорят о настройке электронной торговли, то подразумевается следующее - человек, который настраивает электронную торговлю, должен понимать как:

- сформировать массив с данными;
- сделать это на конкретном движке интернет-магазина;
- извлечь эти данные и передать в инструменты веб-аналитики.

Не всегда маркетолог умеет работать с кодом. Даже не каждый разработчик, хорошо разбирающийся в каком-то одном движке, согласится выполнять эту задачу для интернет-магазина на другой платформе. Поэтому настройку электронной торговли (стандартной или расширенной) необходимо выполнять как минимум в паре: *маркетологу* и *разработчику*. Первый подробно и понятно составляет техническое задание для второго, разработчик его внедряет, а в конце маркетолог проверяет корректность сделанной работы.

Но реальность такова, что заказчик не имеет в штате разработчика под эту задачу. Тогда снова есть несколько вариантов:

1. купить для своей CMS-платформы готовое решение (модуль/плагин);
2. нанять удаленного специалиста, который за \$ настроит электронную торговлю;
3. настроить маркетологу своими силами через Google Tag Manager;

Почему в интернете нет конкретного руководства по настройке?

Во-первых, все интернет-магазины индивидуальны. У кого-то сайт на конструкторе, к которому может не быть доступа к файлам на изменения, у кого-то коробочное решение, кто-то пишет движок с нуля и т.д. Во-вторых, даже в рамках одного интернет-магазина настройку электронной торговли можно реализовать по-разному (напрямую добавляя код отслеживания или через Google Tag Manager), формировать массив данных через уровень данных (dataLayer), или использовать собственный код JavaScript. В-третьих, описать все способы настройки довольно сложно. Получается очень много различных комбинаций.

Настроив электронную торговлю для Google Analytics, она будет работать и для Яндекс.Метрики?

Да, если настройка в обоих случаях реализована с помощью контейнера данных (dataLayer). В официальной документации Яндекса об этом написано следующее:

Имя контейнера данных и структура вкладываемых в него Ecommerce-объектов соответствует аналогичным сущностям в Google Analytics Enhanced Ecommerce. Это означает, что если вы уже настроили отправку данных в Google Analytics Enhanced Ecommerce, в том числе через Global Site Tag, и включили Ecommerce в Яндекс.Метрике, последняя также начнет собирать данные.

Хочу заметить, что в справке речь идет о расширенной электронной торговле. А мы в этом руководстве говорим о **Standard Ecommerce**.

Как настроить стандартную электронную торговлю?

Перейдем к пошаговому плану настройки стандартной электронной торговли с помощью Google Tag Manager.

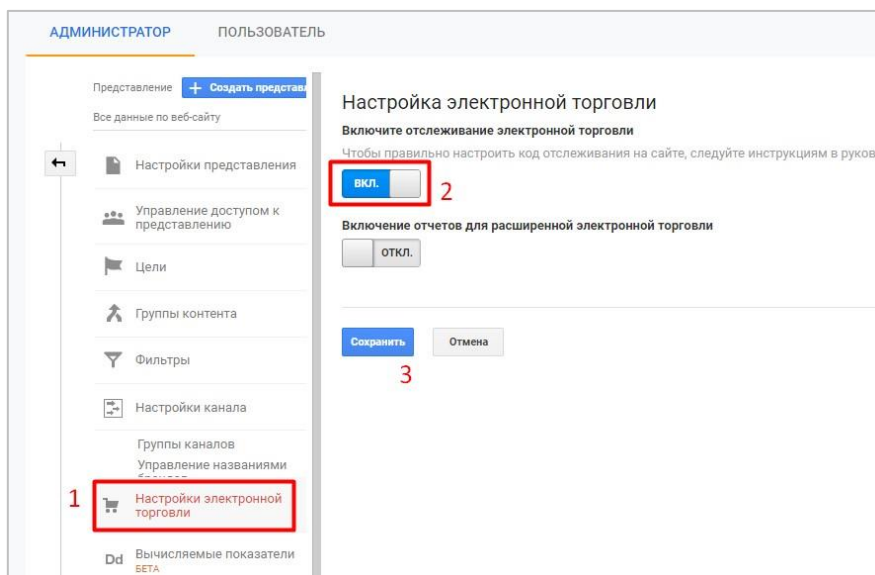
Алгоритм:

1. включить отчеты электронной торговли в Google Analytics;
2. отправить данные о транзакции на уровень данных;
3. отправить данные о совершенной транзакции в Google Analytics (через Google Tag Manager);
4. проверить корректность передачи данных.

Разберем каждый пункт более подробно.

Включить отчеты электронной торговли в Google Analytics

В настройках Google Analytics, на уровне представления, перейдите в раздел **Настройки электронной торговли** и **Включите отслеживание электронной торговли**.



Включение отслеживания электронной торговли

Сохраните настройки.

Отправить данные о транзакции на уровень данных

Уровень данных – это объект (массив объектов) или переменная JavaScript, которая хранит и передает информацию с вашего сайта в Google Tag Manager. Является ключевым компонентом GTM, используется в тегах, триггерах и переменных. Тот, кто понимает принцип формирования и работы dataLayer, имеет существенное преимущественно и может очень гибко настраивать отслеживание различных событий на своем сайте. Если вы еще не знакомы с уровнем данных, рекомендую прочитать эту статью: <https://osipenkov.ru/datalayer/>

Чтобы настроить отслеживание электронной торговли Google Analytics с помощью диспетчера тегов Google, необходимо передать данные о транзакции на уровень данных. Затем в Google Tag Manager мы создаем правила, при которых покупка считается успешной, а после с помощью тега Universal Analytics передаем сведения о транзакции из уровня данных в Google Analytics.

О том, как настроить отслеживание электронной торговли, написано в официальной справке Google (<https://support.google.com/tagmanager/answer/6107169>). Нет ничего лучше, чем следовать оригинальному источнику. Все сведения о транзакциях должны передаваться через уровень данных.

Пример кода выглядит следующим образом:

```
<script>
window.dataLayer = window.dataLayer || []
dataLayer.push({
  'transactionId': '1234',
  'transactionAffiliation': 'Acme Clothing',
  'transactionTotal': 38.26,
  'transactionTax': 1.29,
  'transactionShipping': 5,
  'transactionProducts': [{
    'sku': 'DD44',
    'name': 'T-Shirt',
    'category': 'Apparel',
    'price': 11.99,
    'quantity': 1
  }, {
    'sku': 'AA1243544',
    'name': 'Socks',
    'category': 'Apparel',
    'price': 9.99,
    'quantity': 2
  }
]});
</script>
```

Существует два типа данных со своими обязательными переменными и типом:

1. Данные о транзакции (**transactionId**, **transactionTotal**);

Данные о транзакции

Имя переменной	Описание	Тип
transactionId (обязательная переменная)	Уникальный идентификатор транзакции	Строка
transactionAffiliation (необязательная переменная)	Партнер или магазин	Строка
transactionTotal (обязательная переменная)	Общая сумма транзакции	Число
transactionShipping (необязательная переменная)	Стоимость доставки для данной транзакции	Число
transactionTax (необязательная переменная)	Сумма налога для данной транзакции	Число
transactionProducts (необязательная переменная)	Список приобретенных товарных единиц для данной транзакции	Массив объектов товаров

Обязательные переменные - transactionId, transactionTotal

2. Данные о товарах (**name, sku, price, quantity**);

Данные о товарах

Имя переменной	Описание	Тип
name (обязательная переменная)	Название товара	Строка
sku (обязательная переменная)	Идентификатор товара	Строка
category (необязательная переменная)	Категория товара	Строка
price (обязательная переменная)	Цена за единицу товара	Число
quantity (обязательная переменная)	Количество единиц товара	Число

Обязательные переменные - name, sku, price, quantity

Важно:

- имена переменных (**transactionId, transactionTotal, transactionShipping** и т.д.) должны называть точно также, как написано в документации Google;
- у каждой переменной - свой тип данных. Где-то строка, где-то число. Это важно, поскольку данные о транзакциях можно передавать не только в Google Analytics, но и в другие инструменты веб-аналитики, например, в Facebook. При некорректном типе переменной возникнет ошибка;
- вся структура кода должны быть идентична тому, который представлен в справке Google. Вы можете не использовать необязательные параметры, но вы не можете переименовать **transactionId** в **purchasedId**;
- пример выше - это всего лишь пример. Соответственно, значения всех параметров должны динамически заменяться функцией (-ями), которые напишет разработчик. Простое копирование этого кода ни к чему не приведет. Ведь транзакции 1234 (на примере выше) в вашем случае может вообще не быть, а сумма покупки изменяется от заказа к заказу;
- собственные переменные не поддерживаются в коде. Тот, кто формирует уровень данных, должен точно следовать руководству и использовать те переменные, которые есть в документации;
- не все переменные, представленные в коде выше, являются обязательными. Можно сократить их количество, оставив только обязательные.

Таким образом, для формирования уровня данных для стандартной электронной торговли можно пойти 2 путями:

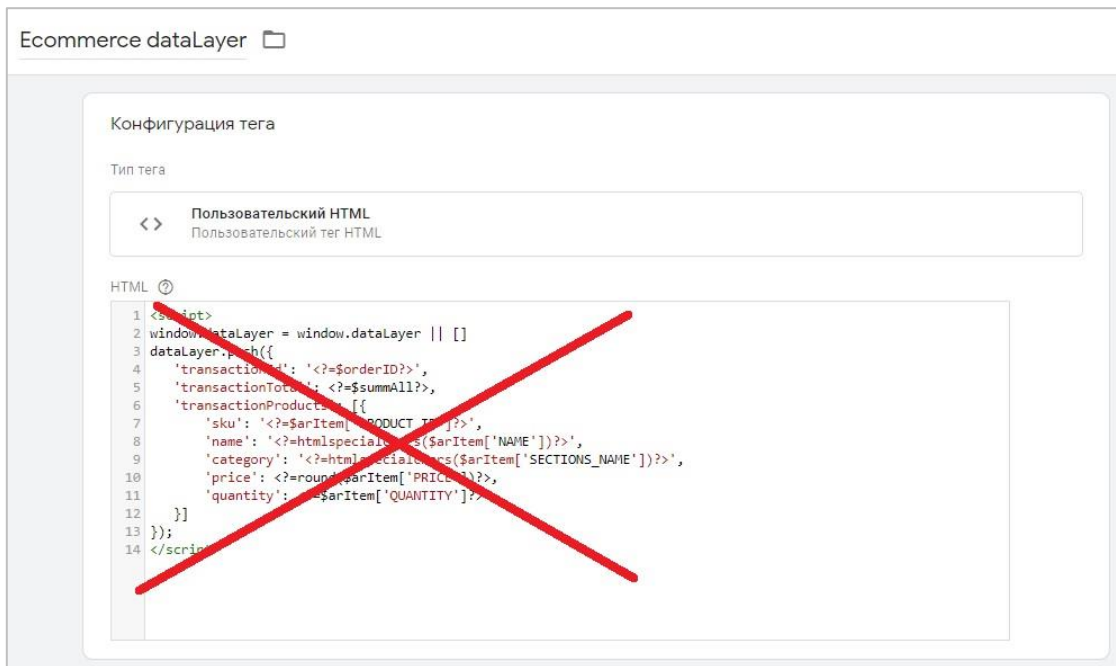
1. заполнить и передать в инструменты веб-аналитики только 2 обязательных переменных о самой транзакции (уникальный идентификатор транзакции **transactionId** и общую сумму транзакции **transactionTotal**);
2. заполнить и передать в инструменты веб-аналитики 2 обязательных переменных о транзакции и + 4 о товарах в заказе. В этом случае добавляются к **transactionId** и **transactionTotal** -> **name, sku, price, quantity**. Всего 6 переменных.

Сколько данных и какие следует передавать в Google Analytics решаете самостоятельно. Однако чем больше данных вы передаете, тем больше вы увидите информации в отчетах по электронной торговле.

Google Tag Manager не поддерживает PHP, поэтому если ваш разработчик пришлет вам сформированный уровень данных такого вида:

```
<script>
window.dataLayer = window.dataLayer || []
dataLayer.push({
  'transactionId': '<?=$orderID?>',
  'transactionTotal': <?=$summAll?>,
  'transactionProducts': [{
    'sku': '<?=$arItem['PRODUCT_ID']?>',
    'name': '<?=htmlspecialchars($arItem['NAME'])?>',
    'category': '<?=htmlspecialchars($arItem['SECTIONS_NAME'])?>',
    'price': <?=round($arItem['PRICE'])?>,
    'quantity': <?=$arItem['QUANTITY']?>
  }]
});
</script>
```

И вы захотите вставить его в **Пользовательский HTML тег**, то либо вы получите ошибку компилятора, либо вам просто выведется необработанный PHP-код в браузер, а не выполнится сам код. Если вы хотите увидеть результат работы скрипта на PHP, то вам нужно будет запустить этот скрипт на собственном сервере и получить результаты через JavaScript. Вот здесь как раз и пригодятся знания работы с Google Tag Manager.



Некорректное формирование уровня данных в GTM посредством функций PHP

Google Tag Manager и DOM Scraping

Но что, если мы не хотим или не можем обратиться за помощью к разработчику? Тогда нам нужно формировать уровень данных самостоятельно. Сделать это можно с помощью Google Tag

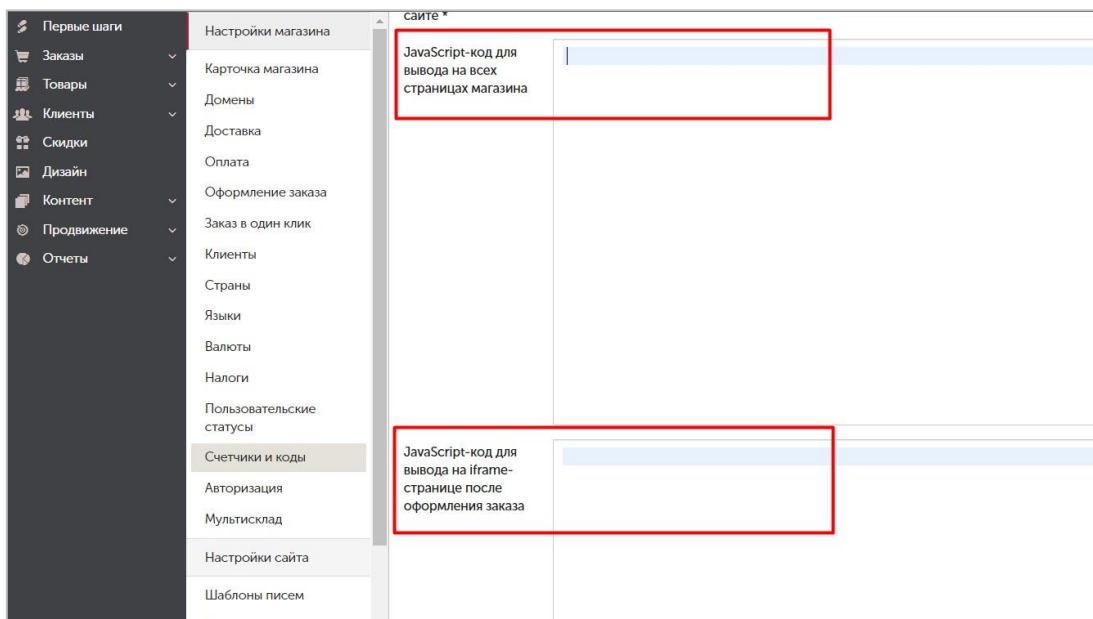
Manager. Технология получения веб-данных путем извлечения их со страниц сайтов по-английски называется **DOM Scraping (Web scraping)**. Этим мы и будем заниматься.

В таком способе извлечения данных со страницы сайт есть один существенный недостаток - все, что мы делаем, завязано на объектной модели документа (DOM). И при изменении каких-либо классов, идентификаторов отслеживаемых элементов на сайте есть большая вероятность потерять все настройки. Именно поэтому лучше выполнять отслеживание электронной торговли или любых других нестандартных событий на сайте с помощью функций и привлечения разработчика, а не извлекая данные с помощью скрапинга.

В качестве примера я буду использовать один из интернет-магазинов по продаже товаров для взрослых на платформе inSales и следующие инструменты:

- расширение GTM Variable Builder для браузера Google Chrome, которое позволяет создавать пользовательские переменные типа **Собственный код JavaScript** и извлекать значения из элементов сайта всего за пару кликов;
- специальный тег из Галереи шаблонов, который называется **dataLayer Builder + Standard Ecommerce**.

inSales не поддерживает PHP. Но в настройках есть возможность добавления собственного JavaScript-кода:



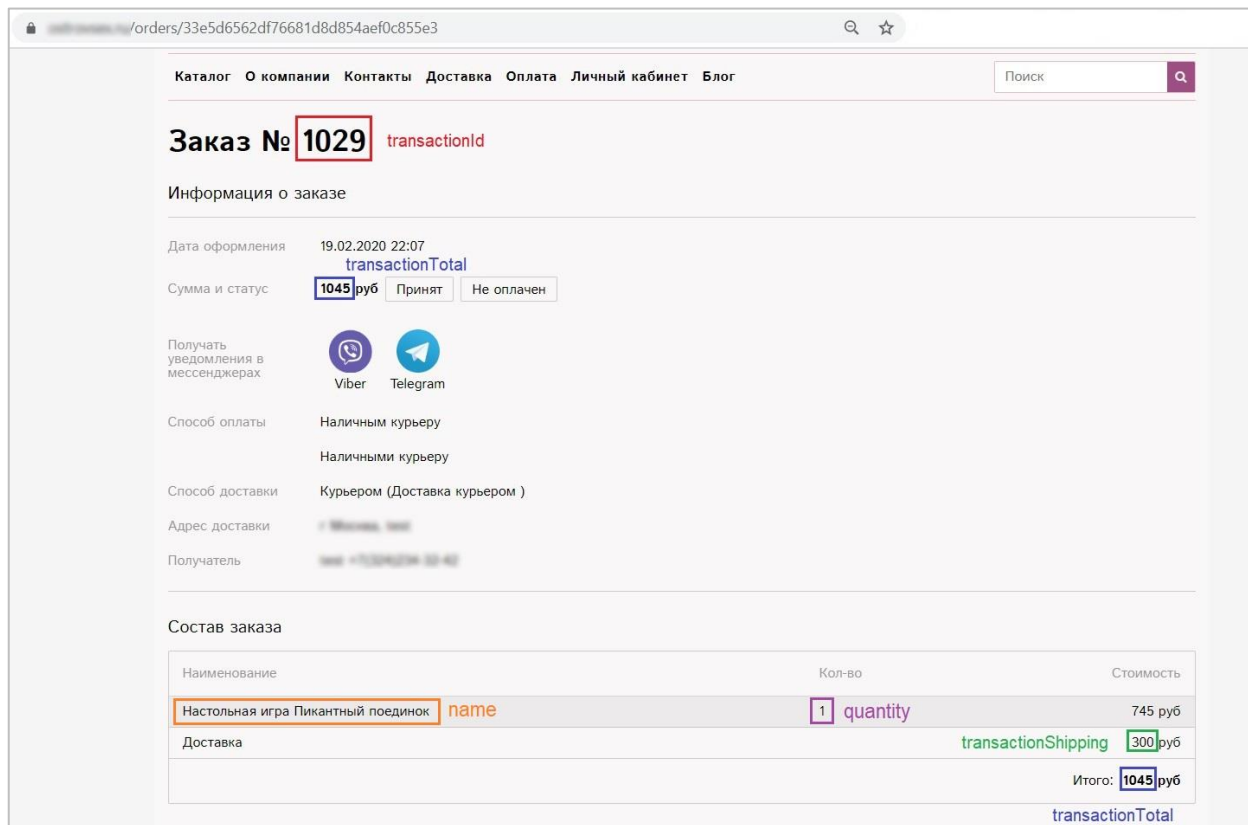
Возможность добавления собственного JavaScript-кода на страницах сайта

Я не буду использовать эти поля, а сформирую уровень данных через Google Tag Manager. Давайте приступим к извлечению обязательных переменных для настройки стандартной электронной торговли.

Важно: значения всех извлекаемых со страницы переменных должны быть доступны на странице "Спасибо" во время отправки тега. Если на странице не выводится идентификатор транзакции, общая сумма покупки или какие-либо другие обязательные переменные, которые

участвуют в формировании уровня данных (данные о товаре - name, sku, price, quantity), то все равно необходимо будет прибегнуть к помощи разработчика. Для скрапинга важно, чтобы извлекаемые данные находились на нужной нам странице. В данном случае, это страница успешной покупки ("Спасибо"). Если этих данных нет, то и нечего извлекать.

В моем примере страница успешно оформленного заказа выглядит так:



Страница "Спасибо"

С этой страницы мы можем извлечь данные о переменных **transactionId** и **transactionTotal** (на примере выводится в двух местах), стоимости доставки **transactionShipping**, хоть она и не является обязательной, а также наименование товара **name** и количество **quantity**. К сожалению, для корректного формирования уровня данных 2 обязательных переменных из 4 для информации о товарах недостаточно. Поэтому есть два выбора:

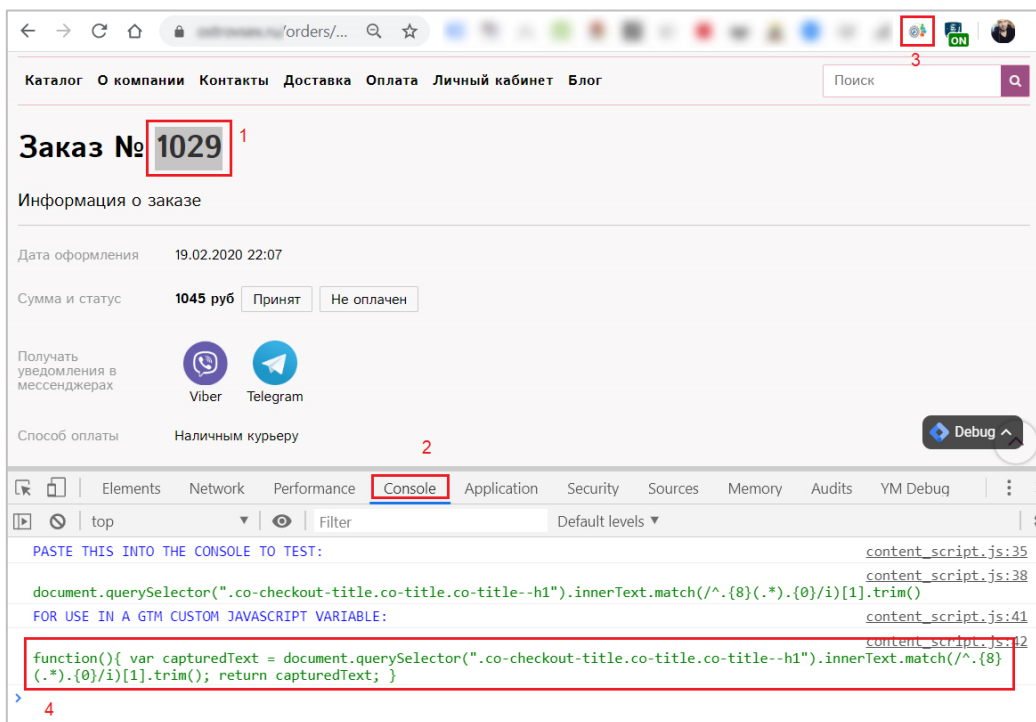
1. нужно просить разработчика изменить эту страницу и добавить недостающие **sku** и **price**;
2. передавать в уровне данных только **transactionId** и **transactionTotal** данные о транзакции без каких-либо других переменных.

Использование расширения GTM Variable Builder

Предположим, мы выбрали второй вариант. Чтобы извлечь данные по этим переменным, воспользуемся расширением **GTM Variable Builder**. Подробнее о том, как оно работает, описано в моем блоге (ссылка: <https://osipenkov.ru/gtm-variable-builder/>). Если кратко, то:

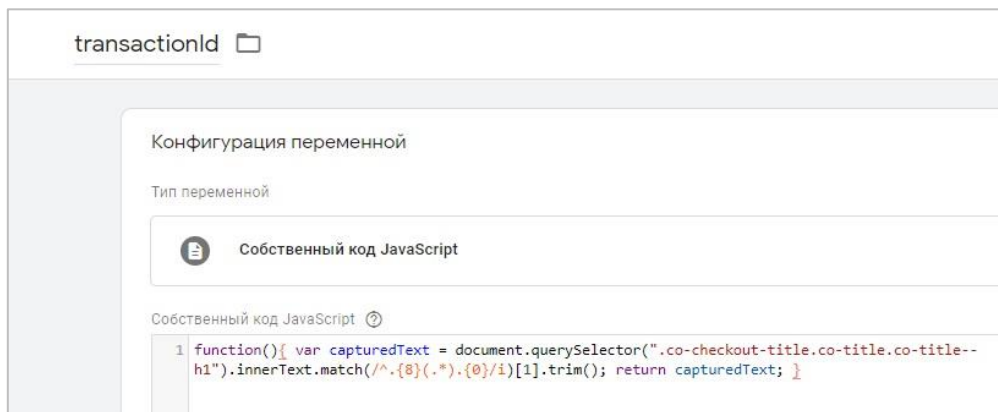
- выделяем нужный фрагмент на странице для извлечения данных;

- жмем на иконку расширения GTM Variable Builder в правом верхнем углу браузера;
- на вкладке **Console** получаем результат;



Извлечение данных для transactionId

- копируем код из строки **FOR USE IN A GTM CUSTOM JAVASCRIPT VARIABLE** и вставляем его в переменную типа **Собственный код JavaScript**.



transactionId

Сохраняем переменную. Аналогично проделываем со всеми остальными переменными. Не забудьте, что общая сумма транзакции (transactionTotal) должна быть числом. Если вы не хотите использовать расширение GTM Variable Builder, то тогда следует искать на странице "Спасибо" у конкретного элемента его CSS-селектор, копировать его, при необходимости проводить манипуляции с этим элементом (преобразование типа данных, обрезание части текста и т.д.), а после добавить в пользовательскую переменную типа **Собственный код JavaScript**.

В Google Tag Manager в режиме предварительного просмотра на вкладке **Variables** напротив конкретного события отображаются данные. Значения переменных с типом **number** (строка) выводятся без кавычек, а с типом **string** (строка) в кавычках. Как в случае с **transactionId**:

Event	Variable Name	Source	Type	Value
5 Click	name	Собственный код JavaScript	string	'Настольная игра Пикантный поединок'
4 Window Loaded	price	Собственный код JavaScript	number	745
3 purchase	transactionId	Собственный код JavaScript	string	'1029'
2 DOM Ready	transactionShipping	Собственный код JavaScript	number	300
	transactionTotal	Собственный код JavaScript	number	1045

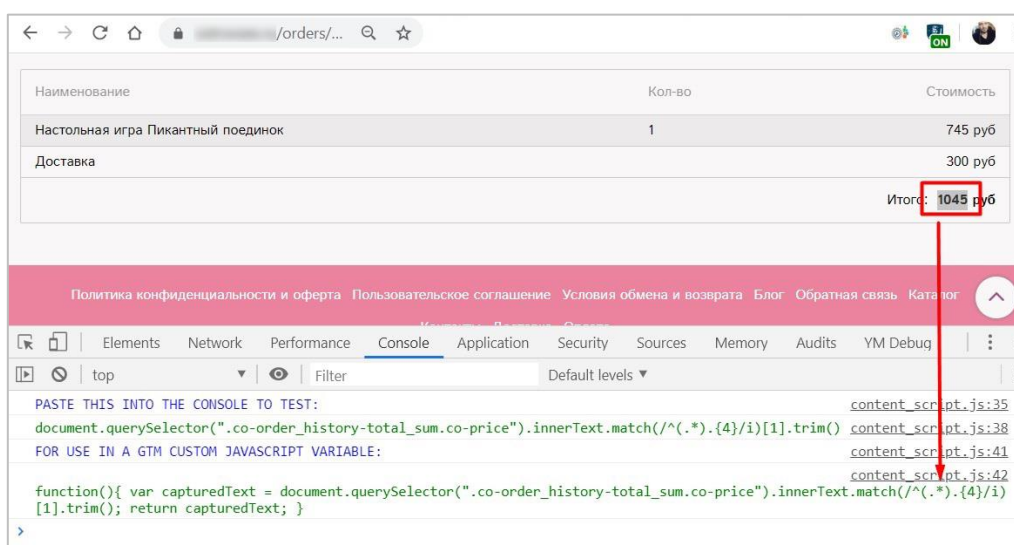
Типы переменных в GTM

На вкладке **Data Layer** переменные с числовым значением выделяются зеленым цветом, а со строковым - красным.

```
Tags Variables Data Layer Errors (0)
purchase
purchase:
1 {
2   event: 'purchase',
3   name: 'Настольная игра Пикантный поединок',
4   sku: '68210', строка
5   price: 745, число
6   quantity: 1,
7   transactionId: '1029',
8   transactionTotal: 1045,
9   gtm.uniqueEventId: 3
10 }
```

Цвет типа данных в Google Tag Manager

Примечание: извлекать числовые переменные (например, стоимость доставки или общую сумму транзакции) с помощью GTM Variable Builder можно не целиком, а лишь часть (без руб.). Просто выделите фрагмент и расширение само допишет нужное *регулярное выражение*.



Извлечение суммы транзакции

Данные о товарах извлекать чуть сложнее, поскольку в одной транзакции может быть несколько товаров. Здесь нам также необходим CSS-селектор товара, который мы извлечем со страницы

"Спасибо". Если вы используете расширение GTM Variable Builder, то он сам сформирует код для переменной, и вместо **document.querySelector** будет использовать **document.querySelectorAll**, потому что нужно захватить все значения продукта, а не только первого.

Если вы будете извлекать данные вручную, воспользуйтесь кодом ниже:

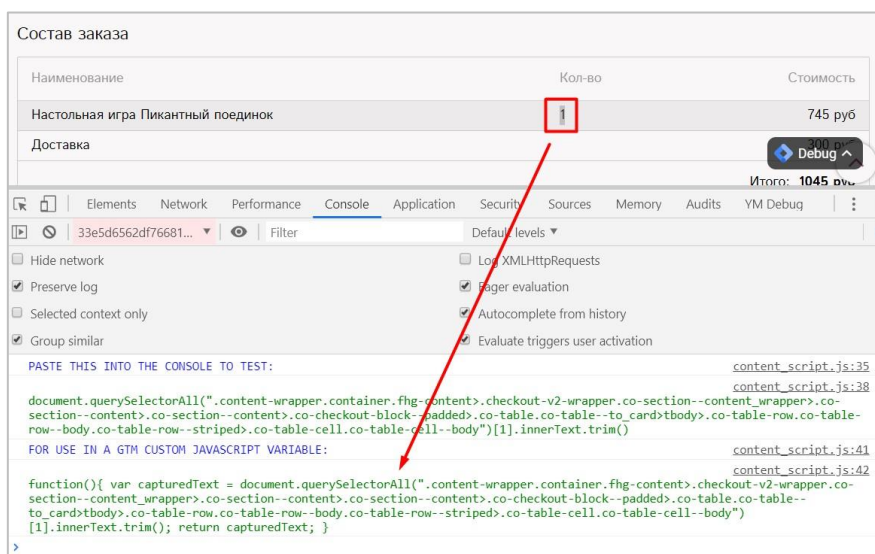
```
function() {  
var items = document.querySelectorAll("CSS-селектор");  
var itemsLength = items.length;  
var myProducts = []  
for (i = 0; i < itemsLength; i++) {  
    myProducts.push(items[i].innerText);  
}  
return myProducts;  
}
```

, где в поле **"CSS-селектор"** вставляете то значение, которое присутствует на вашем сайте.

- **document.querySelectorAll** - возвращает статический (не динамический) NodeList, содержащий все найденные элементы документа, которые соответствуют указанному селектору.
- **document.querySelector** - возвращает первый элемент (Element) документа, который соответствует указанному селектору или группе селекторов.

Поскольку в заказе может быть несколько товаров (2, 3, 5, 10 и т.д.), то **document.querySelector** (как было в примерах ранее) нам не подойдет, поскольку он добавит в массив только первый элемент. Чтобы в dataLayer поместились все товары, которые купил пользователь, используем **document.querySelectorAll**.

Для наглядности приведу пример извлечение **quantity** с помощью GTM Variable Builder на странице "Спасибо":



Извлечение данных по переменной quantity со страница "Спасибо"

Как видим, GTM Variable Builder вернул нам значение с `document.querySelectorAll`. А это значит, что в dataLayer передается корректная информация.

Примечание: используйте точку в качестве десятичного разделителя в числе, а не запятую. При передаче числового значения запятая может быть неправильно определена.

Итак, что же у нас получилось? На примере моего интернет-магазина на странице "Спасибо" я могу извлечь 2 обязательные переменные для данных о транзакции (**transactionId** и **transactionTotal**), 1 необязательную (**transactionShipping**) и 2 из 4 для формирования данных о товаре (**name** и **quantity**). На странице успешного оформления покупки нет только ID товара (**sku**) и цены за единицу товара (**price**). В этом случае я могу отправить только 2 переменные в уровень данных, а затем эту информацию в Google Analytics.

Заказ № 1034

Информация о заказе

Дата оформления 20.02.2020 23:38

Сумма и статус 5568 руб Принят Не оплачен

Tag Manager Tags Variables Data Layer Errors (0) Version: QUICK_PREVIEW GTM-MVBZ5K

Summary purchase

```
purchase:
1 {event: 'purchase', transactionId: '1034', transactionTotal: 5568, gtm.uniqueEventId: 6}
```

Data Layer values after this Message:

```
1 {
2   gtm: {start: 1582231253643, uniqueEventId: 6},
3   event: 'purchase',
4   transactionId: '1034',
5   transactionTotal: 5568
6 }
```

Отправка 2 переменных

В результате в отчете "Транзакции" в Google Analytics получим данные:

ОТЧЕТЫ

Отчет в реальном времени

Аудитория

Источники трафика

Поведение

Конверсии

Цели

Электронная торговля

Обзор

Эффективность товаров

Эффективность продаж

Транзакции

Время до покупки

Многоканальные последовательности

Атрибуция БЕТА

+ Добавить сегмент

Статистика

Доход и Выбор показателя

День Неделя Месяц

Доход

6 000,00 Р

4 000,00 Р

2 000,00 Р

четверг, 20 февраля 2020 г.

Основной параметр: Идентификатор транзакции

Дополнительный параметр

Идентификатор транзакции	Доход	Налог	Доставка	Количество
	5 568,00 Р % от общего количества: 100,00 % (5 568,00 Р)	0,00 Р % от общего количества: 0,00 % (0,00 Р)	0,00 Р % от общего количества: 0,00 % (0,00 Р)	0 % от общего количества: 0,00 % (0)
1. 1034	5 568,00 Р (100,00 %)	0,00 Р (0,00 %)	0,00 Р (0,00 %)	0 (0,00 %)

Строк на странице: 10 К строке: 1 1-1 из 1

Этот отчет создан 20.02.2020 в 23:43:40 - Обновить отчет

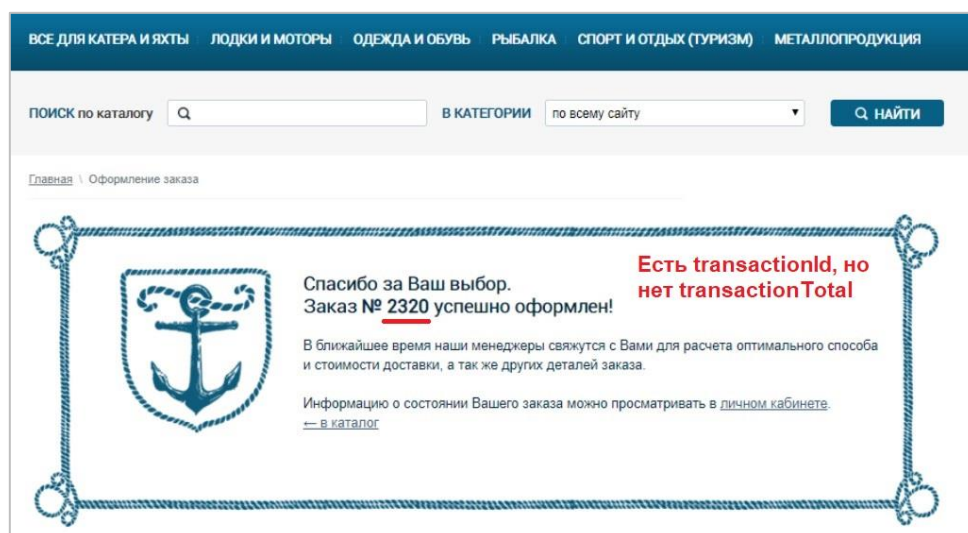
Отчет "Транзакции"

А сам уровень данных в Google Tag Manager будет иметь вид:

```
HTML ⓘ
1 <script>
2 window.dataLayer = window.dataLayer || [];
3 dataLayer.push({
4   'event': 'purchase',
5   'transactionId': '{{transactionId}}',
6   'transactionTotal': {{transactionTotal}}
7   });
8 </script>
```

Уровень данных с 2 обязательными переменными

Таким образом, настроить самую простую реализацию уровня данных для стандартной электронной торговли и передачей этой информации в Google Analytics можно всего лишь с помощью 2 обязательных переменных. Если вам необходимо передавать и данные о товаре, то сформируйте другой уровень данных, содержащий +4 обязательных переменных (**name, sku, price, quantity**). Но для начала извлеките с помощью GTM значения всех обязательных переменных.



Пример страницы "Спасибо", на которой нет transactionTotal

Если у вас также, как и у меня в этом примере, нет каких-то данных, вы можете пропустить эти переменные и сформировать уровень данных без них:

```
<script>
window.dataLayer = window.dataLayer || [];
dataLayer.push({
  'event': 'purchase',
  'transactionId': '{{transactionId}}',
  'transactionTotal': {{transactionTotal}},
  'transactionShipping': 0,
  'transactionProducts': [{
    'sku': '',
    'name': '{{name}}',
    'price': '',
    'quantity': {{quantity}}
  ]
});
</script>
```

Просто в Google Analytics будет попадать не вся информация о товарах, а в недостающих значения в отчетах будет написано (**not set**).

Сформировать такой уровень данных можно несколькими способами:

1. создать тег типа **Пользовательский HTML**;
2. использовать специальный тег из Галереи шаблонов **dataLayer Builder + Standard Ecommerce**.

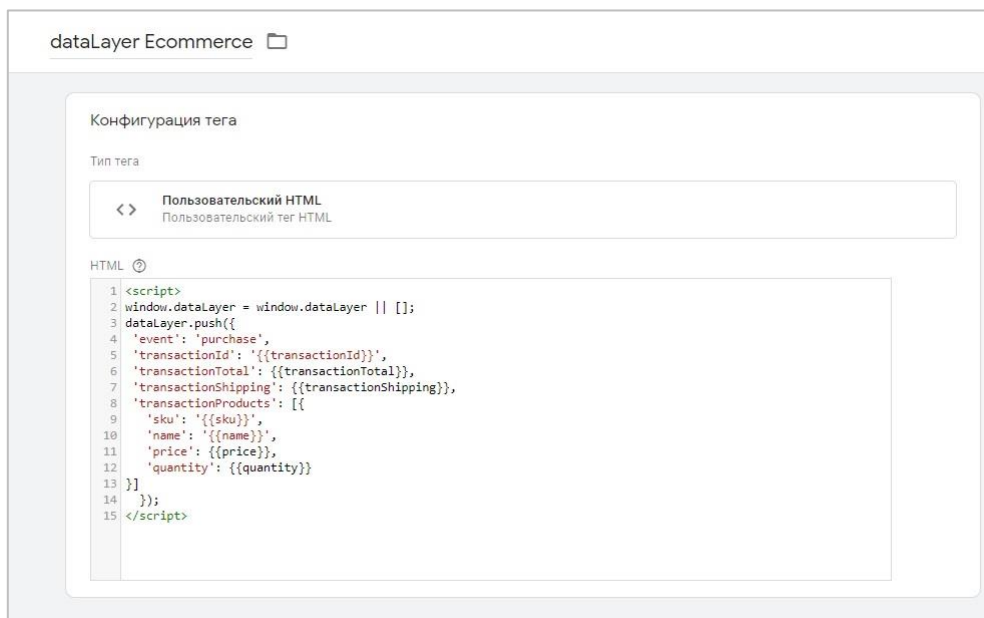
Разберем два способа настройки.

Создание пользовательского HTML тега

Перейдите в раздел **Теги** и создайте тег типа **Пользовательский HTML**. Вставьте в него следующую конструкцию:

```
<script>
window.dataLayer = window.dataLayer || [];
dataLayer.push({
  'event': 'purchase',
  'transactionId': '{{transactionId}}',
  'transactionTotal': {{transactionTotal}},
  'transactionShipping': {{transactionShipping}},
  'transactionProducts': [{
    'sku': '{{sku}}',
    'name': '{{name}}',
    'price': {{price}},
    'quantity': {{quantity}}
  ]
});
</script>
```

В Google Tag Manager это будет выглядеть так:



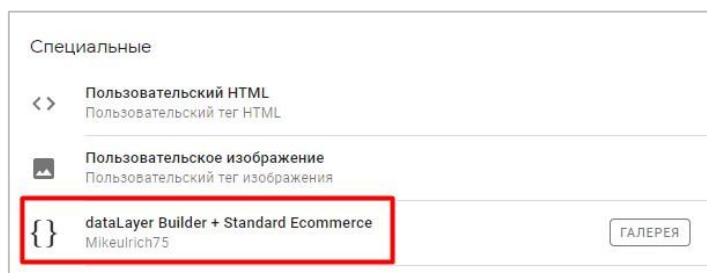
Сформированный уровень данных для электронной торговли

Каждой обязательной переменной присваивается значение переменной, которое мы создали на предыдущих шагах. С помощью двойных фигурных скобок **{{ }}** мы можем обратиться к любой переменной GTM.

Использование специального тега из Галереи шаблонов

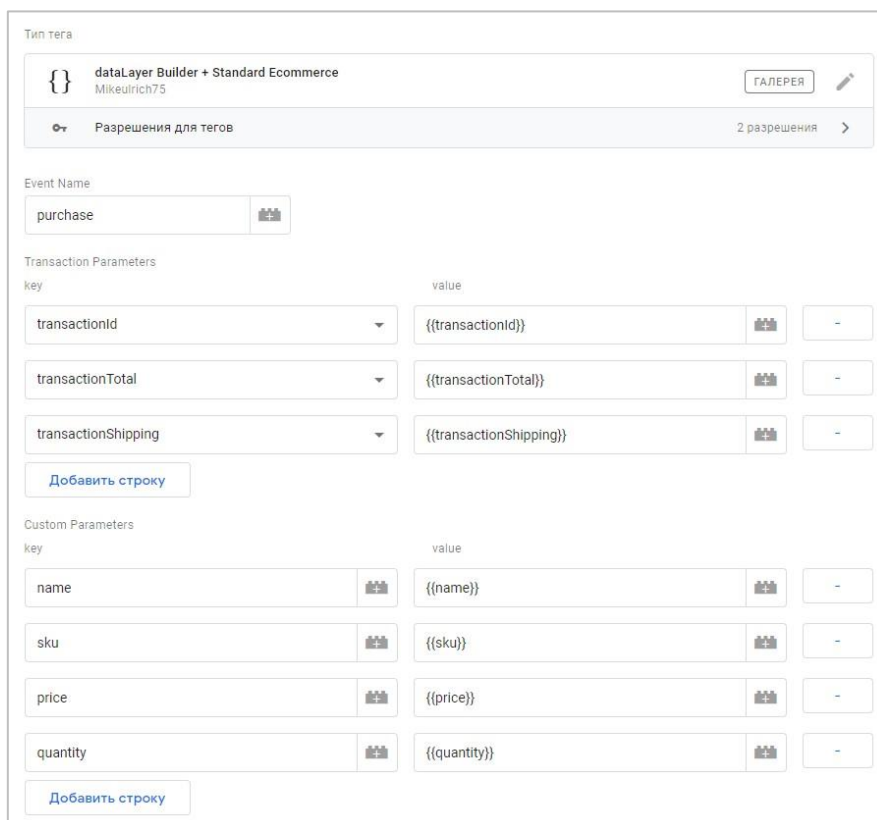
В октябре 2019 года команда Google представила **Галерею шаблонов**, которая призвана помочь рекламодателям в работе с тегами партнеров. Независимые разработчики могут делиться своим решениями, чтобы другие быстро внедряли и управляли их тегами, а также снижали вероятность неправильной настройки.

Некоторое время назад в Галерее шаблонов появился **dataLayer Builder + Standard Ecommerce** и **dataLayer Builder + Enhanced Ecommerce** от **Mikeulrich75**. Поскольку мы настраиваем стандартную электронную торговлю, то устанавливаем его.



Тег dataLayer Builder + Standard Ecommerce

Тег позволяет упростить формирование уровня данных. Больше не нужно создавать пользовательский HTML тег, вручную прописывать переменные, следить за синтаксисом кода (исправлять кавычки, запятые, скобки и т.д.). Все это можно сделать в простом и удобном интерфейсе тега. Он чем-то напоминает пользовательскую переменную **Таблица поиска** в Google Tag Manager.



Настройки тега

В поле **Event Name** введите название пользовательского события. Это **purchase**. В параметрах транзакции всего два обязательных ключа со своими значениями - это **transactionId** и **transactionTotal**. Но у меня еще был **transactionShipping**. В поле **value** я добавляю созданные переменные (см. выше). Аналогично проделываем и с параметрами товара - **name, sku, price, quantity**.

Таким образом, с появлением Галереи шаблонов GTM и готового тега от Mikeulrich75 стало проще формировать уровень данных для электронной торговли. Но можно использовать и первый вариант настройки, как вам будет удобнее.

В приведенном выше коде мы помещаем событие GTM с именем **purchase** на уровень данных. Для того, чтобы событие срабатывало и данные отправлялись в Google Analytics, необходимо:

1. создать пользовательское событие **purchase**;
2. создать тег Universal Analytics с типом отслеживания **Транзакция**.

Но перед тем, как это сделать, давайте разберем различные способы размещения уровня данных на странице сайта. Чтобы узнать, где метод **dataLayer.push ()** должен быть добавлен, нужно подумать о том, что происходит после успешной покупки?

Покупателя перенаправляет на отдельную страницу "Спасибо" с уникальным URL? Или страница остается прежней, URL не изменяется, а пользователю на экране отображается только сообщение об успешной транзакции?

Если клиент перенаправляется на страницу "Спасибо", и вы используете уровень данных без события, то код электронной торговли должны быть размещен над кодом контейнера Google Tag Manager.

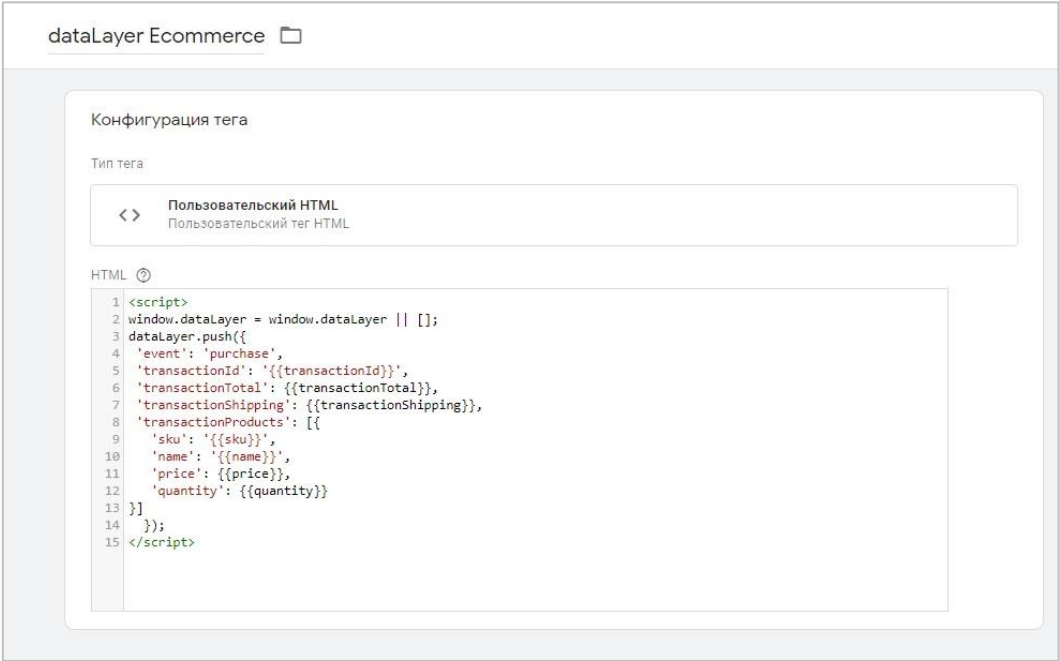
Что значит **dataLayer.push ()** без события?

А теперь хотелось бы показать вам одну тонкую деталь, которые многие могли бы не заметить, но которая играет важную роль при отслеживании электронной торговли. И это строчка кода с **'event': 'purchase'** и без нее. В официальной документации и в самом начале этой статьи я приводил такой код:

```
<script>
window.dataLayer = window.dataLayer || []
dataLayer.push({
  'transactionId': '1234',
  'transactionAffiliation': 'Acme Clothing',
  'transactionTotal': 38.26,
  'transactionTax': 1.29,
  'transactionShipping': 5,
  'transactionProducts': [{
    'sku': 'DD44',
    'name': 'T-Shirt',
    'category': 'Apparel',
    'price': 11.99,
    'quantity': 1
  },{
    'sku': 'AA1243544',
    'name': 'Socks',
    'category': 'Apparel',
    'price': 9.99,
    'quantity': 2
  }]
});
</script>
```

Код из документации Google

А затем в Google Tag Manager настраивал такой:



Сформированный уровень данных для электронной торговли

Казалось бы, разница в одной строчке **'event': 'purchase'** после **dataLayer.push**, но она все меняет. В первом случае (без event) важен порядок размещения кода (ДО или ПОСЛЕ контейнера GTM), а во втором случае уже нет, поскольку это классический формат события Google Analytics, где есть **Категория, Действие, Ярлык, Ценность**. Просто выглядит чуть сложнее. Но если присмотреться и записать код в одну строчку, то становится очень похоже:

```
dataLayer.push('event', 'purchase', {'transactionId': '{{transactionId}}', ....});
```

на конструкцию обычного отслеживаемого события, например, для библиотеки gtag.js:

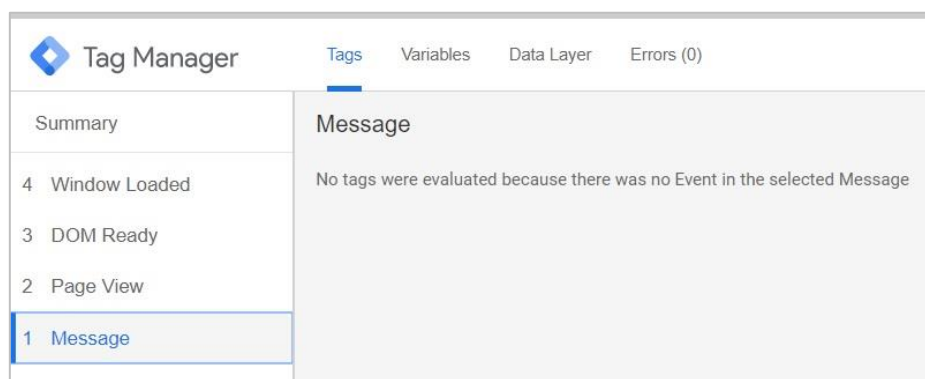
```
gtag('event', <action>, { 'event_category': <category>, 'event_label': <label>, 'value': <value>});
```

И что это значит? Если в `dataLayer` отсутствует строка с **event**, то мы не сможем создать триггер типа **Пользовательское событие** и активировать его в тот момент, когда нам это необходимо.

Способ №1. Код `dataLayer.push ()` добавляется ДО контейнера GTM (на странице «Спасибо»)

Возвращаемся к размещению кода **`dataLayer.push ()`** над кодом контейнера GTM в случае, если пользователь перенаправляется на страницу "Спасибо" с отдельным URL. Почему так? Потому что тогда данные о транзакции электронной торговли уже будут доступны для Google Tag Manager, когда он начнет загружаться. А чем раньше эти данные будут переданы на уровень данных, тем быстрее мы сможем отправить эти данные в Google Analytics.

Поскольку этот `dataLayer.push` не имеет параметра **event** (как я написал выше), в режиме отладки он будет отображаться словом **Message**.



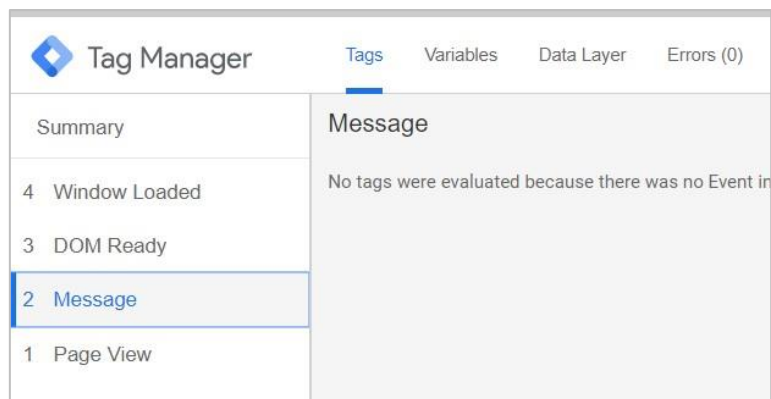
Код расположен ДО контейнера GTM

Message мы не сможем использовать в качестве триггера в GTM. Поэтому самый ранний момент, когда мы можем отправить данные транзакции в Google Analytics - это событие просмотра страницы (Page View).

Вы прекрасно знаете, что каждый раз, когда на сайте загружается страница, и вы включаете режим предварительного просмотра GTM, там отображаются 3 события: **Просмотр страницы (Page View)**, **Модель DOM готова (DOM Ready)** и **Окно загружено (Window Loaded)**. Если разработчик на сайте разместит код электронной торговли с `dataLayer.push` до контейнера Google Tag Manager, то вы сможете отправить данные о транзакции в Google Analytics с событием Page View.

Способ №2. Код `dataLayer.push ()` добавляется ПОСЛЕ контейнера GTM (на странице «Спасибо»)

Поскольку этот `dataLayer.push` также не имеет параметра **event**, в режиме отладки он также будет отображаться как **Message**.

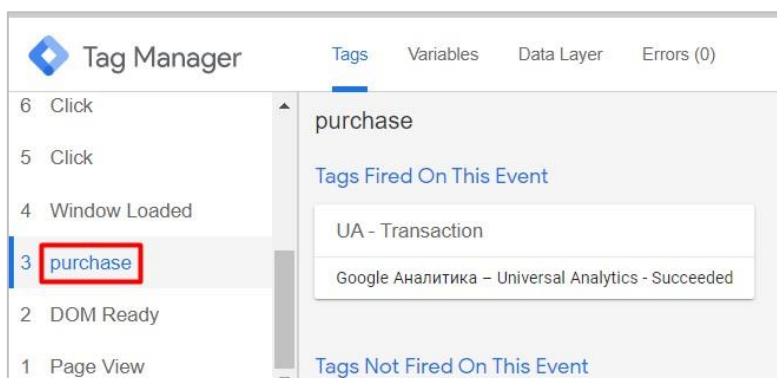


Код расположен ПОСЛЕ контейнера GTM

Однако на этот раз `dataLayer.push` был завершён чуть позже, потому что он расположен ниже контейнера Google Tag Manager. Теперь с событием `Page View` мы не сможем получить доступ к данным. Остается использовать только два других события: **Модель DOM готова (DOM Ready)** или **Окно загружено (Window Loaded)**. Чаще всего при настройке используют именно второе событие **DOM Ready**.

Способ №3. Код `dataLayer` активируется на странице с помощью события

В этом способе как раз и используется дополнительный параметр события (**event**), значением которого является покупка (**purchase**).



Событие `purchase`

Такой способ чаще всего используется веб-аналитиками, поскольку с помощью пользовательского события мы можем настроить активацию триггера и тега как угодно.

И вот теперь мы снова возвращаемся к нашему примеру и части, когда для того, чтобы событие срабатывало и данные отправлялись в Google Analytics, необходимо:

1. создать пользовательское событие **purchase**;
2. создать тег Universal Analytics с типом отслеживания **Транзакция**.

Чтобы создать пользовательское событие, перейдите на вкладку **Триггеры** и создайте триггер типа **Пользовательское событие** с именем события **purchase**:

Тип триггера

Пользовательское событие

Имя события

Использовать регулярные выражения

Условия активации триггера

Все специальные события Некоторые специальные события


Пользовательское событие purchase

Далее перейдите на вкладку **Теги** и создайте тег **Universal Analytics** с типом отслеживания **Транзакция**.

UA - Transaction

Конфигурация тега

Тип тега

 Google Аналитика – Universal Analytics
Google Платформа для маркетинга

Тип отслеживания

Транзакция

Настройки Google Analytics

Включить переопределение настроек в этом теге

Идентификатор отслеживания

{{UA-}}

Триггеры

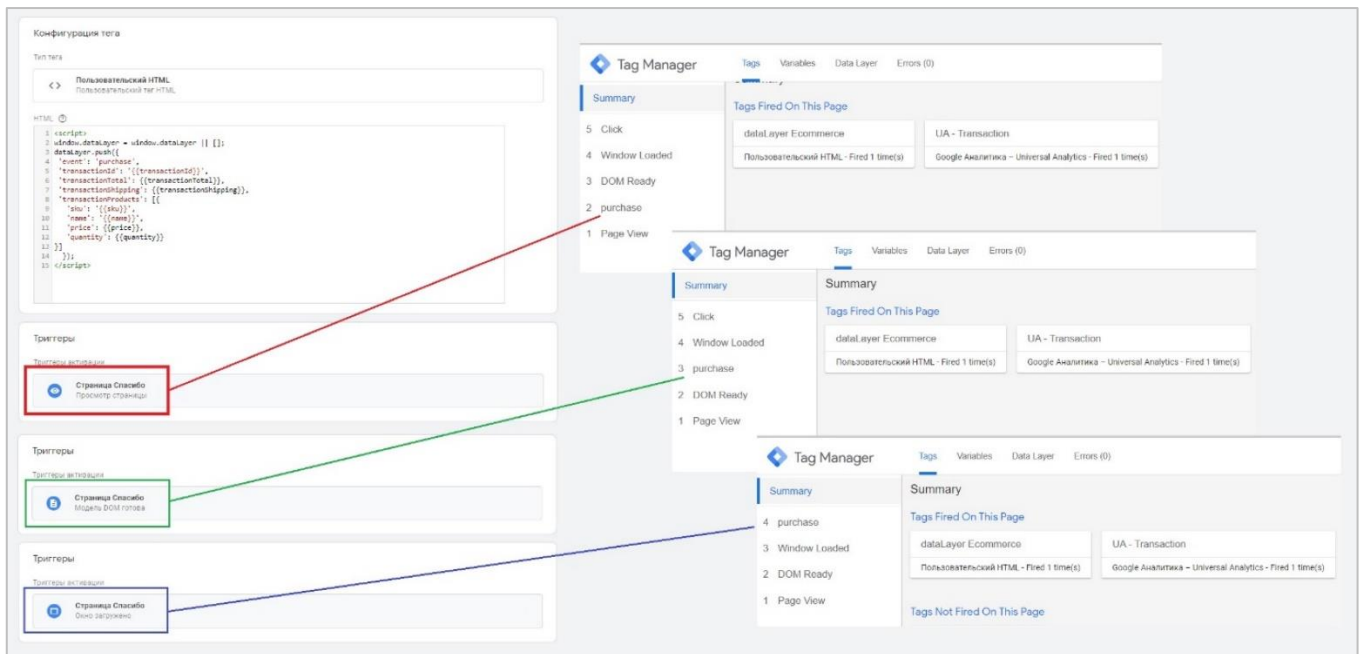
Триггеры активации

purchase
Пользовательское событие

Universal Analytics с типом отслеживания "Транзакция"

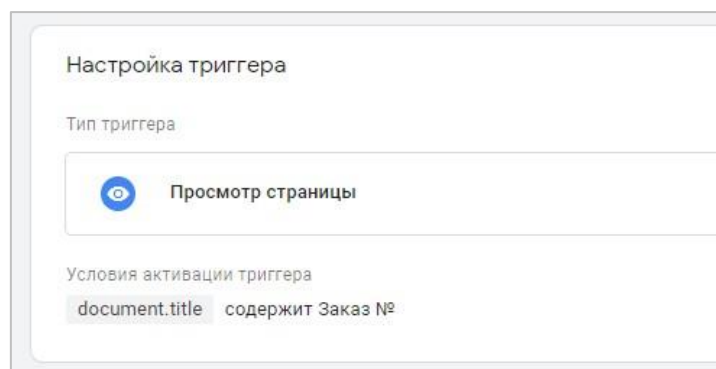
В качестве триггера активации укажите пользовательское событие **purchase**. Сохраните настройки.

И последнее, что осталось выяснить - это триггер активации для уровня данных, который формируется в теге типа **Пользовательский HTML** тег. Какой он? При условии, что вы используете третий способ отслеживания (код с событием **'event': 'purchase'**), не имеет значения, что выбрать. Отличие будет в том, когда будут передаваться данные - с **Page View**, с **DOM Ready** или с **Window Loaded**. Они все равно передадутся.



Различные способы активации тега

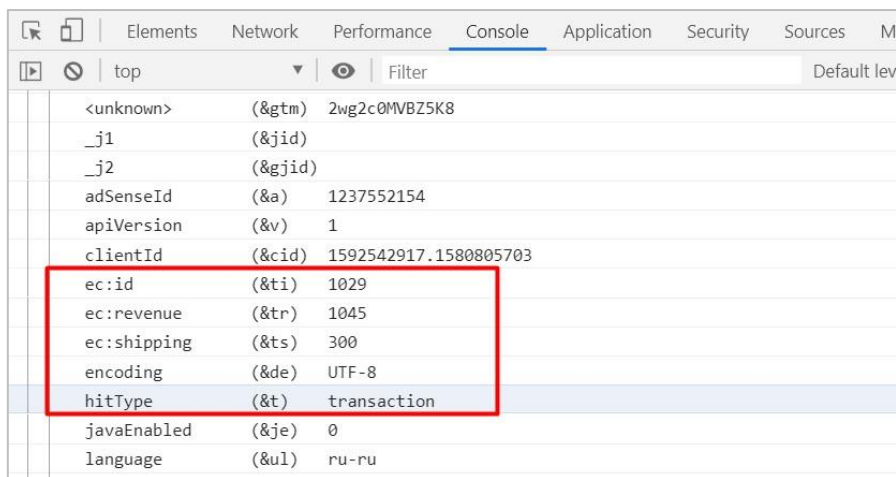
В качестве условия активации триггера можно указать конкретно страницу "Спасибо". У меня в примере это реализовано через заголовок страницы, который на странице успешного заказа имеет title **Заказ №:**



Условие активации триггера - на странице "Спасибо"

Все, что осталось сделать, это проверить корректность формирования уровня данных и отправки информации о транзакции в Google Analytics. Для этого можно воспользоваться расширением **GA Debugger**.

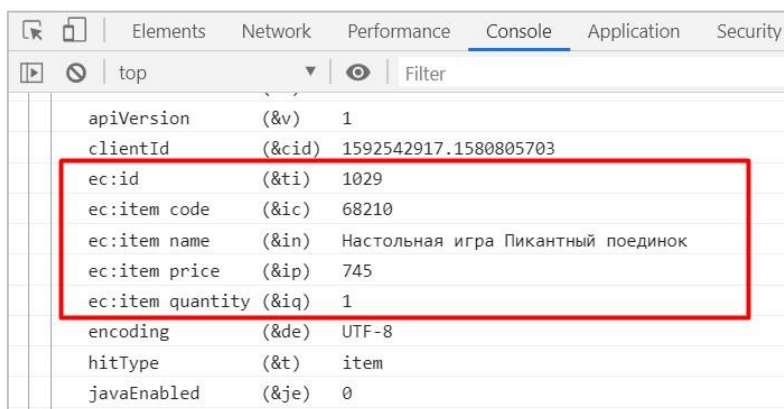
Включите расширение, откройте вкладку **Console** в браузере и совершите транзакцию. Вы увидите много информации, появляющейся в консоли. GA Debugger проверяет веб-сайт и выводит всю информацию, связанную с Google Analytics. Нас интересует **hitType - transaction**:



<unknown>	(>m)	2wg2c0MVBZ5K8
_j1	(&jid)	
_j2	(&gid)	
adSenseId	(&a)	1237552154
apiVersion	(&v)	1
clientId	(&cid)	1592542917.1580805703
ec:id	(&ti)	1029
ec:revenue	(&tr)	1045
ec:shipping	(&ts)	300
encoding	(&de)	UTF-8
hitType	(&t)	transaction
javaEnabled	(&je)	0
language	(&ul)	ru-ru

hitType - transaction

Это покупка, которая была отправлена в Google Analytics. **ec:id** - идентификатор транзакции, **ec:revenue** - общая сумма покупки (в Google Analytics - Доход), а также стоимость доставки **ec:shipping**. Если вы передаете информацию о купленных товарах, то прокрутите консоль вниз. Вы должны увидеть такие данные:

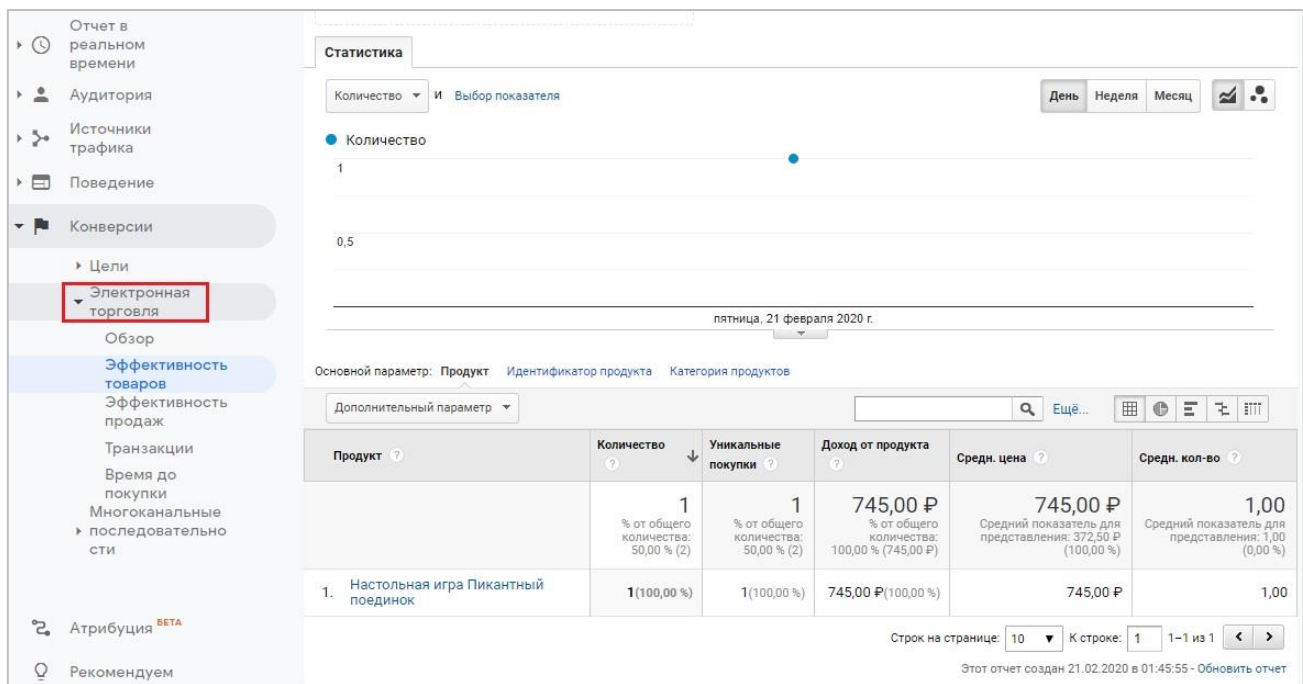


apiVersion	(&v)	1
clientId	(&cid)	1592542917.1580805703
ec:id	(&ti)	1029
ec:item code	(&ic)	68210
ec:item name	(&in)	Настольная игра Пикантный поединок
ec:item price	(&ip)	745
ec:item quantity	(&iq)	1
encoding	(&de)	UTF-8
hitType	(&t)	item
javaEnabled	(&je)	0

Данные о товаре в GA Debugger

Посмотрите всю консоль на предмет ошибок. Если они есть, исправьте самостоятельно или проконсультируйтесь с разработчиком. Если ошибок нет, то на этом настройка стандартной электронной торговли с помощью Google Tag Manager завершена.

Данные по отслеживанию доступны в отчетах Google Analytics в разделе **Конверсии - Электронная торговля**:



Отчеты по электронной торговле

Итоги

Несмотря на то, что стандартная электронная торговля настраивается проще, чем расширенная, по причине меньшего количества отслеживаемых событий, она все равно может быть сложна для начинающих.

В рамках этого руководства мы с вами:

- досконально разобрали процесс настройки без привлечения разработчика и код стандартной электронной торговли;
- узнали про обязательные и необязательные переменные для уровня данных;
- поработали с технологией DOM Scraping;
- научились извлекать данные со страницы напрямую и с помощью GTM Variable Builder;
- познакомились с инструментами, которые упрощают работу - GTM Variable Builder, специальный тег dataLayer Builder + Standard Ecommerce, GA Debugger;
- освоили 3 способа размещения уровня данных на странице сайта;
- передали всю информацию в Google Analytics.

Как я упомянул ранее, не рекомендуется использовать извлечение информации со страницы сайта для формирования уровня данных, поскольку любое изменение CSS-селекторов элементов со стороны верстальщика приведет к сбою в отслеживании. Но в то же время, настройка стандартной электронной торговли с помощью DOM Scraping без привлечения разработчика демонстрирует всю мощь диспетчера тегов Google.

При подготовке данного руководства частично использовались материалы с сайтов mixedanalytics.com и analyticsmania.com