gtag.js: Die geniale Zukunft von Google Analytics (und wie du dich darauf vorbereiten kannst)

Veröffentlicht am 6. Dezember 2017
Letztes Update: 17. November 2020

Ein gedrücktes Seufzen ging mir über die Lippen, als ich in der Digital- und Webanalyse-Helden Facebook Gruppe von Maik Bruns über gtag.js (Global Site Tag) gelesen habe - der früher oder später analytics.js (Universal Analytics) ersetzen wird: *Schon wieder migrieren?!!* Ja und nein: *Erfahre hier über die geniale Zukunft von Google Analytics und wie du dich perfekt auf sie vorbereiten kannst.*

> **Update 2020 + Info zu diesem Blogbeitrag: App+Web ist die *alte Bezeichnung* der neuen GA Generation,** die seit Oktober 2020 raus aus der Beta ist. Im Zuge dessen hat Google die App+Web auf „Google Analytics 4 bzw. GA4“ umbenannt. >> Alle Infos dazu findest du hier: **Hurra, hurra GA4 ist da!**
>
> **Hier findest du auch den ultimativen Google Analytics 4 Einsteiger- und Durchstarter Guide.**

Rückblick auf analytic.js

So lange ist es noch gar nicht her, als die Google Analytics Welt von ga.js auf analytics.js umgestiegen ist: 2013, also 7+ Jahre.

Für die digitale Welt und so mancher Internet Standards ein halbes Jahrhundert: So wurde XML abgelöst und durch JSON-Objekte ersetzt, was einen nachhaltigen Einfluss auf Tag Management Systeme hatte...

Der Aufwand umzusteigen war damals enorm: Gerade auf Agenturseite! Denn alle Kunden mussten nach und nach auf analytics.js umgestellt werden: Und je nach Unternehmensgröße und Google Analytics Implementierung war das extrem aufwendig und hat einige Ressourcen in Anspruch genommen.

Und jetzt schon wieder umsteigen? Auf gtag.js?

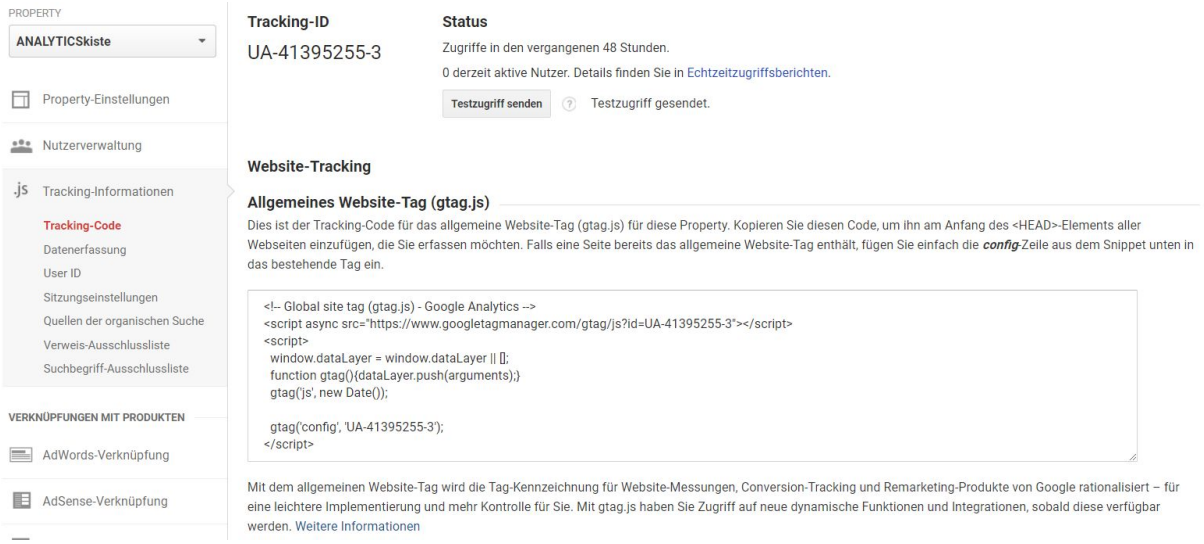
Wo es mit Sicherheit da draußen noch Websites gibt die ga.js verwenden...

Ausblick auf gtag.js

Ja, auf jeden Fall! Denn wir wollen ja nicht stehen bleiben, sondern einen riesengroßen Schritt in Richtung Zukunft machen: Und da gehört gtag.js dazu.

Im September 2017 hat Google die neue Javascript Bibliothek gtag.js (Global Site Tag) zum Tracking Standard für Google Analytics und Google Ads erklärt.

Seitdem sieht der **native Tracking Code** im Admin-Interface der beiden Tools so aus (hier am Beispiel in GA):



Screenshot: gtag.js Basis Tracking Code

Anders als analytics.js, ist **gtag.js keine reine Google Analytics Library**, sondern unterstützt und ersetzt gleichzeitig das AdWords Conversion Tracking (conversion.js).

Google setzt also auf die technische Vereinheitlichung seiner Google Produkte: Somit wird zukünftig nicht nur der Einbau der GA Tracking Codes erheblich erleichtert sondern es werden auch zahlreiche technische Fehlerquellen minimiert.

Jetzt NEU: Mit dem Global Site Tag macht Google das Website Tracking für die neue GA4 Property und somit für Firebase kompatibel.

gtag.js: Das ändert sich in naher Zukunft

Auf den ersten Blick wirkt der Umstieg auf gtag.js relativ simpel.

Wenn du Google Analytics bisher nativ implementiert hast *(also direkt über den Quellcode der Website eingebunden und nicht via Tag Management Tool),* entferne den alten analytics.js Code und ersetzt ihn gegen gtag.js.

Nativ implementiert: analytics.js raus

```
<!-- Google Analytics -->
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-XXXXXXXX-X', 'auto');
ga('send', 'pageview');
</script>
<!-- End Google Analytics -->
```

Nativ implementiert: gtag.js rein

```
<!-- Global site tag (gtag.js) -->
<script async
src="https://www.googletagmanager.com/gtag/js?id=UA-XXXXXXXX-X"></script>
<script>
 window.dataLayer = window.dataLayer || [];
 function gtag(){dataLayer.push(arguments);}
 gtag('js', new Date());

gtag('config', 'UA-XXXXXXXX-X');
</script>
```

Das wars!

Vorerst bleibt alles andere gleich, denn gtag.js ruft im Hintergrund analytics.js auf und sendet die Daten wie gewohnt und an Google Analytics.

analytics.js und alle anderen Google Product-Libraries sind nämlich integraler Bestandteil von gtag.js und das aus zwei guten Gründen:

1. Einerseits wird gtag.js dadurch nicht unnötig aufgeblasen.
2. Andererseits bleibt damit die Kompatibilität mit bestehenden Tracking Methoden bestehen.

Und wie sieht es mit dem GTM aus?

Wenn du bisher deinen GA Tracking Codes mit dem Google Tag Manager implementiert hat, muss sogar noch weniger tun: Hier wird Google vermutlich die Umstellung irgendwann ganz von alleine vornehmen…

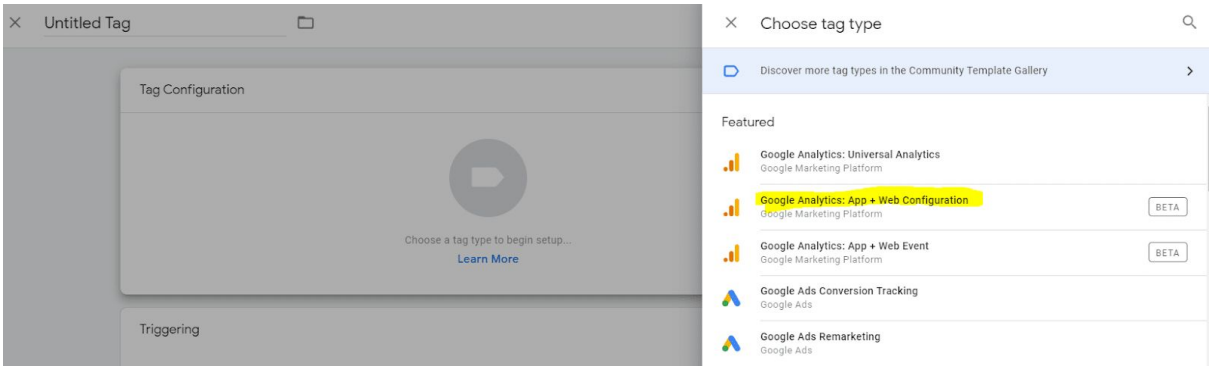
> **Update Juli 2019:** Nope, so einfach ist es (voerst) nicht! Der gtag kann direkt über den Google Tag Manager auf der Website implementiert werden. Dafür gibt es seit Juli 2019 und mit dem Announcement der App+Web Property einen eigenen gtag-Tag. Details zum Setup hier: **Hurra, hurra GA4 ist da!**

NEU: gtag via GTM auf der Website einbauen

Genauso wie analytics.js kann auch gtag.js über den GTM auf der Website eingebunden werden - allerdings nur im Zusammenhang mit der neuen App+Web Property bzw. GA4.

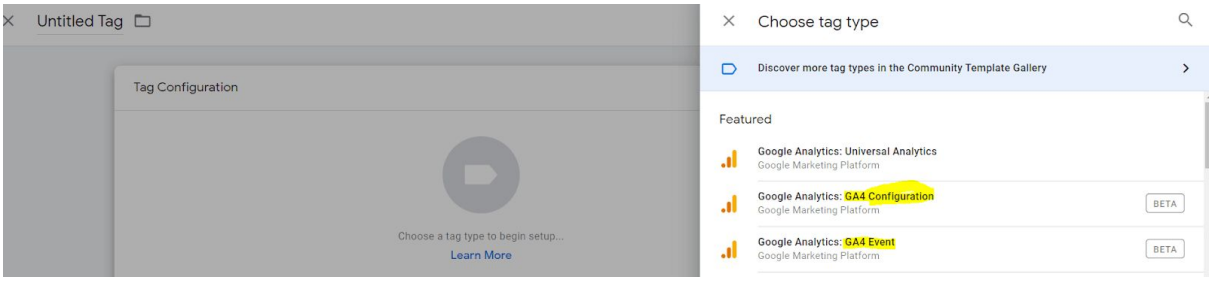
> **Super wichtig:** Die gtag-Implementierung via GTM sollte derzeit **NUR ZUSÄTZLICH** zum bestehenden analytics.js Setup vorgenommen werden, da der Tag speziell für das App+Web zw. GA4 Property Setup erstellt wurde. **Bitte entferne keine bestehenden Google Analytics Tags aus deinem GTM!**

Dafür ist ein neuer Tag Typ vorhanden: Google Analytics: **App+Web Configuration.**



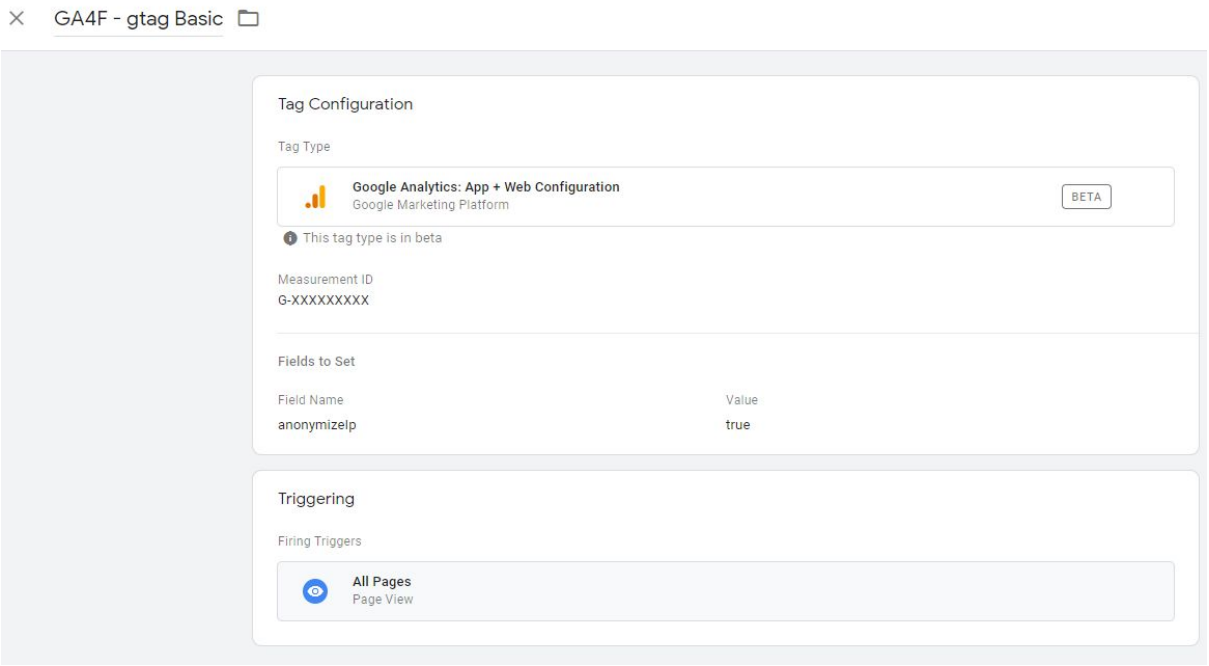
Screenshot: GTM App+ Web Configuration

Update 2020: Die App+Web Property ist jetzt raus aus der Beta und der neue Google Analytics Standard - dazu hier alle Details: Hurra, hurra GA4 ist da! Somit wurde auch der GTM Tag auf GA4 umbenannt:



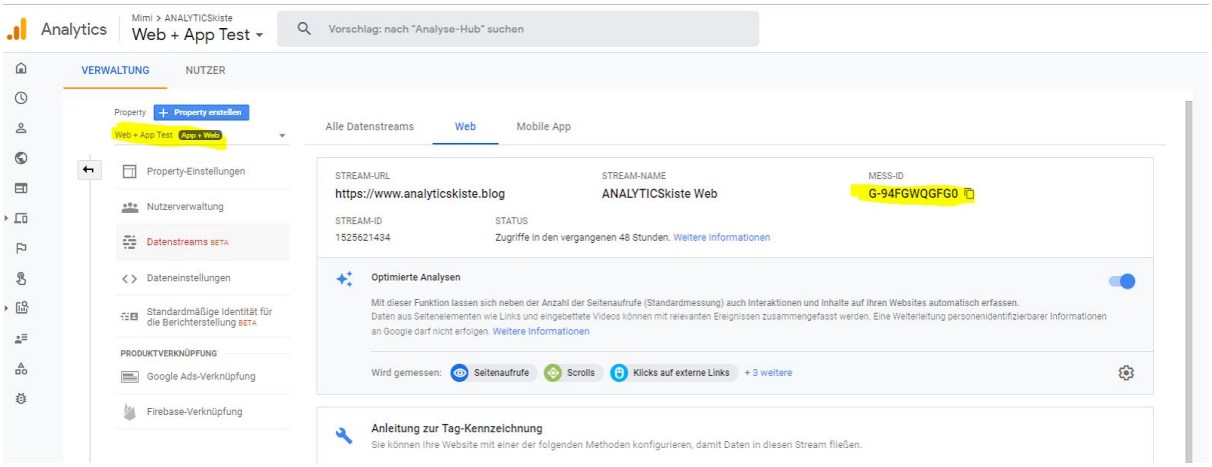
Screenshot: GA4 Tags im GTM

Egal ob App+Web oder GA4 wähle den neuen "Google Analytics Configuration Tag" aus:



Screenshot: GTM gtag Tag Configuration

Trage in das **Feld "Measurement ID"** deine neue GID ein, die du in der App+Web bzw. GA4 Property unter *Verwaltung --> Property --> Datenstreams --> Web* erhältst:



Screenshot: Mess-ID bzw. Measurement ID oder GID in der App+Web Property

> **Hinweis:** Dieser Tag-Typ ist NUR für die neue G-ID kompatibel und nicht mit der alten UA-ID.

Weiters kannst du auch wieder Felder wie anonymizeIP *(dringend empfohlen!)* einstellen.

> **Tipp:** Alle technischen Details findest du im Blog von Simo Ahava.

Füge deinen All Pages Trigger hinzu und veröffentliche den Tag.

gtag ist jetzt über den GTM auf deiner Website eingebunden und feuert zusätzlich zum analytics.js Pageview in dein Universal Analytics einen zusätzlichen Pageview an die App+Web bzw. GA4 Property.

> **Hinweis:** Du kannst das Versenden eines zweiten Pageviews an die App+Web bzw. GA4 Property unterbinden, indem du das Häkchen *"Send a page view event when this configuration loads"* nicht setzt! **Wichtig: Dadurch werden KEINE Pageviews in deiner App+Web bzw. GA4 Property erfasst!**

gtag.js: Das ändert sich in entfernter Zukunft

In entfernter Zukunft sieht die Welt jedoch ganz anders aus: Denn natürlich ändert sich mit der neuen JS Library auch die Art der zukünftigen Datenerfassung.

So werden statt bisher Funktions-Parameter, JSON-Objekte an GA übergebe: Eigentlich so, wie wir es bereits von dataLayer-push gewöhnt sind.

Der große Vorteil: Statt einer beschränkten Parameter-Anzahl, können im Prinzip beliebig viele Objekt-Properties übergeben werden. *Das macht flexibel!*

Ein Beispiel aus der Google Analytics Doku:

Example:

analytic.js:

```
// Creates the default tracker.
ga('create', 'GA_TRACKING_ID', 'auto');

// Uses the default tracker to send the event to the
// Google Analytics property with tracking ID GA_TRACKING_ID.
ga('send', 'event', 'Videos', 'play', 'Fall Campaign');
```

gtag.js:

```
// Sends the event to the Google Analytics property with
// tracking ID GA_TRACKING_ID set by the config command in
// the global tracking snippet.
gtag('event', 'play', {
  'event_category': 'Videos',
  'event_label': 'Fall Campaign'
});
```

Screenshot: Beispiel Natives Tracking Alt vs Neu für Google Analytics

Außerdem gibt es noch zwei weitere bahnbrechende Veränderungen:

> **Zum einen ist dass, das grundlegende Datenmodell von Google Analytics:** Statt auf Sitzungen und Seitenaufrufe fokussiert sich gtag.js genauso wie Firebase auf User und Events. Statt Hit-, Session- und User-basierten Dimensionen gibt es User Properties und Event Parameter.

Damit wird das bisherige **Ereignis Tracking** völlig über den Haufen geworfen wird: >> Lies dazu hier mehr!

Und auch hier nimmt Google den User wieder an die Hand und verhindert wild zusammengewürfelte und unterschiedliche Schreibweisen durch Vereinheitlichung: Denn Google stellt nun eine Liste mit empfohlenen Events und dazu passenden Event-Parametern zur Verfügung.

> **Die zweite Neuerung ist ein riesengroßer, mega Vorteil:** Mit dem nativen gtag.js lassen sich **Google Analytics Properties in Gruppen zusammenfassen**.

Das bedeutet, dass ein Hit (Pageview, Event, Transaktion) *automatisch* an mehrere GA Properties gesendet werden kann - ohne dass dafür alles doppelt und dreifach implementiert werden muss. Und nicht nur zu mehreren GA Properties, sondern sogar automatisch zu verschiedenen Google Produkten z.B. GA und Firebase. *Und das ist wirklich awesome…* Technische Details dazu erfährst du hier.

Aber ich denke, da kommt noch mehr:

Wird Google Analytics technisch gesehen zu Firebase?

> **Update 2019:** Ich hatte recht!

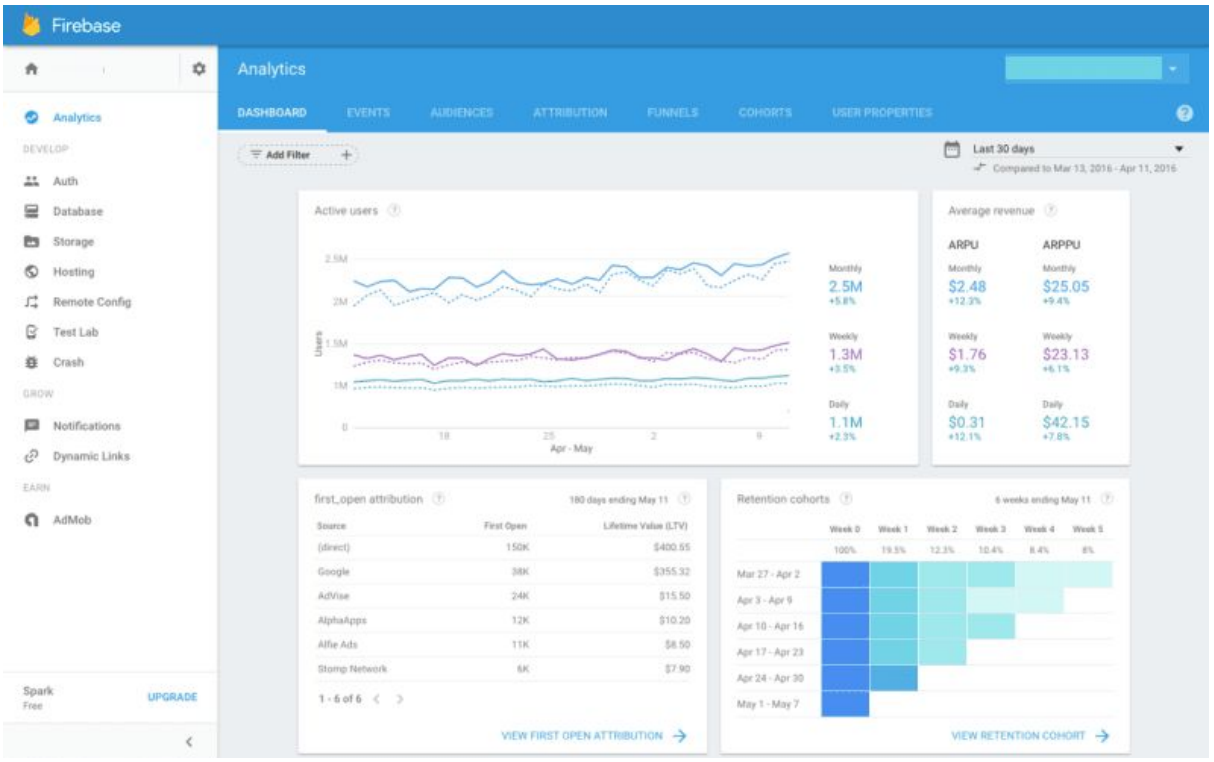
Das alles erinnert mich stark an den Launch von Firebase Analytics 2016: Firebase hat das App Tracking in Google Analytics vollkommen revolutioniert.

Hart genommen hat Firebase Google Analytics als Webanalyse Tool für Apps ersetzt: Neues Tool, neue GTM Version, neues Service SDK.

Alles neu und weg von Google Analytics!

Der **Schwerpunkt von Firebase Analytics sind Ereignisse**: Google stellt dazu zahlreiche Standard Events zur Verfügung *(oha! so wie bei gtag.js)* - und es können zusätzlich bis zu 500 benutzerdefinierte Events implementiert werden *(äh - warum nur 500?)*.

Auch die grafische Oberfläche von Firebase spiegelt sich mittlerweile in Google Analytics wieder: Der Homescreen, die Navigation, die Schreibweise beim Event Tracking…



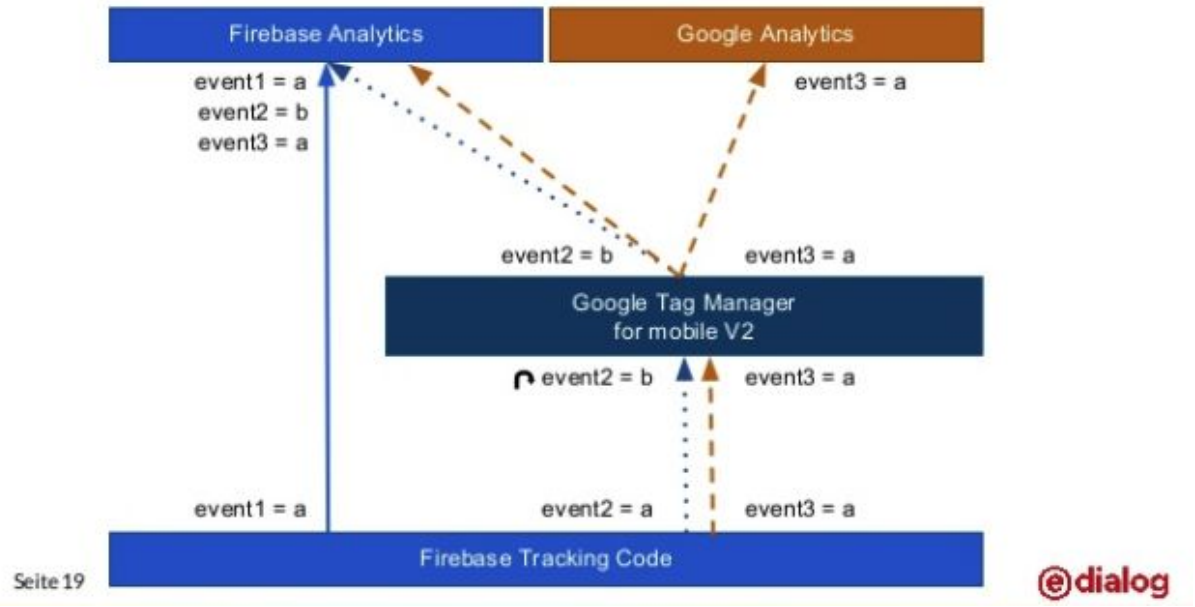
Screenshot: Firebase Analytics Dashboard

Da schaut sich Google Analytics wohl einiges von Firebase ab…

Die wohl größte Veränderung durch Firebase Analytics, ist die Verwendung des Google Tag Managers: Denn der GTM for Mobile ist kein Stand-alone Produkt mehr sondern baut auf der Syntax von Firebase auf. D.h. er mit mit der Firebase SDK ganz einfach mit installiert…

Und genau diese Veränderung könnte es *(ist es derzeit aber nicht!)* mit gtag.js und dem GTM in Zukunft ebenfalls geben: **Der GTM wird direkt über den gtag.js Basistracking Code mit implementiert.** Ob dataLayer-push oder gtag() im Code verwendet wird, ist dann egal - es ist das selbe!

Das vereinfacht nicht nur die Integration ganz enorm, sondern bietet auch die Möglichkeit Daten sowohl an Firebase als auch an Google Analytics *(und später an jedes andere Google Produkt)* zu senden - wie ich 2016 schon für meinen Firebase Vortrag am MeasureCamp #9 sehr schön dargestellt habe:



Screenshot: Google Tracking Code in Zukunft, Quelle: Slideshare, e-dialog

> **Update 2019: Die Grafik ist obsolet! App+Web bzw. GA4 Daten können jetzt automatisch über die neue App+Web bzw. GA4 Property in GA erfasst werden!**
>
> **Hinweis:** Aufgrund meiner Grafik könnte man fälschlicherweise annehmen, dass gtag.js den GTM ersetzen könnte. Das ist nicht der Fall! Denn gtag.js verfügt und wird vermutlich nie, über eine grafische Oberfläche verfügen. Außerdem können mit gtag.js auch nur Google Produkte angesteuert werden. Mit dem GTM können hingegen auch third-party Tags implementiert werden.

gtag.js vs. analytics.js: Umsteigen oder nicht?

Wenn du eine neue Website erstellst und Google Analytics nativ einbinden möchtest, kommst du um den gtag ohnehin nicht herum, da er seit 2017 der neue Tracking Standard für GA ist.

Wenn du analytics.js nativ implementiert hast, brauchst du heute nicht auf gtag umsteigen - außer du möchtest von der neuen App+Web bzw. GA4 Property profitieren: Dann ist der Umstieg zwingend erforderlich. In dem Fall empfehle ich dir aber gleich auch den Umstieg auf den Google Tag Manager.

Wenn du analytics.js via GTM eingebunden hast und von der neuen App+Web bzw. GA4 Property profitieren möchtest, kannst du den gtag direkt über den GTM und den neuen App+Web bzw. GA4 Tag-Typ auf deiner Website einbinden. Wichtig ist, dass du diesen ZUSÄTZLICH zu deiner bestehenden Implementierung nutzt!

> **Ansonsten ist derzeit noch keine komplette Umstellung erforderlich!**

Ähnlich wie bei Firebase, empfiehlt Google natürlich früher oder später auf gtag.js umzusteigen, da neue Features nur noch für gtag.js bereitgestellt werden.

Lass dich davon aber nicht stressen!

Denn nachdem sowohl gtag.js als auch die App+Web Property derzeit noch Beta sind, rät selbst Google von einer kompletten Umstellung von analytics.js auf gtag.js ab!

> **Update 2020: App+Web ist die *alte Bezeichnung* der neuen GA Generation,** die seit Oktober 2020 raus aus der Beta ist. Im Zuge dessen hat Google die App+Web auf „Google Analytics 4 bzw. GA4“ umbenannt. >> Alle Infos dazu findest du hier: **Hurra, hurra GA4 ist da!**

Warten wir erst mal in Ruhe ab was in den kommenden Jahren alles auf uns zukommt, bevor wir unser gesamtes Website Tracking erneut überarbeiten.

> **Hinweis:** Und keine Sorge, Google wird *analytics.js* noch sehr, sehr lange unterstützen. Denn obwohl *urchin.js* gefühlt schon 100 Jahre alt ist, unterstützt Google auch diese Bibliothek immer noch. Genauso wird es also vermutlich für *ga.js, analytics.js* und die Bibliotheken aller anderen Google Produkte auch sein.

Auf in die Zukunft mit gtag.js

Obwohl ich am Anfang dieses Blogartikels noch sehr skeptisch auf *gtag.js* geblickt habe und voller Verzweiflung an die anstrengende Migrations-Arbeit dachte: *Nach diesem Blogartikel ist die Freude gigantisch.*

> **gtag.js und Firebase fürs Web ist die Zukunft des Website Trackings!**

Mit gtag.js wird die Code-Implementierung vereinheitlicht und vereinfacht: Was heute noch mit mühsamen Workarounds “schirch” implementiert wird, wird zukünftig “schön” lösbar sein *(und das erfreut Tekki-Herzen)*.

Mit Firebase fürs Web bzw. GAv2 werden sich auch die Analyse Möglichkeiten verändern: Ereignisse sind ein erster Schritt in diese Richtung. Die neuen Reports in GAv2 der Zweite.

Ich denke aber, da kommt noch viel mehr...

Call to Action

Du hast eine Frage zum Artikel?

Schreibe sie einfach in die Kommentare. Ich helfe, wo ich kann.

Happy Analyzing, Deine Michaela