

gatsby-plugin-google-analytics

Easily add Google Analytics to your Gatsby site.

Upgrade note

This plugin uses Google's `analytics.js` file under the hood. Google has a guide recommending users upgrade to `gtag.js` instead. There is another plugin `gatsby-plugin-gtag` which uses `gtag.js` and we recommend it.

Install

```
npm install gatsby-plugin-google-analytics
```

How to use

```
// In your gatsby-config.js
module.exports = {
  plugins: [
    {
      resolve: `gatsby-plugin-google-analytics`,
      options: {
        // The property ID; the tracking code won't be generated without it
        trackingId: "YOUR_GOOGLE_ANALYTICS_TRACKING_ID",
        // Defines where to place the tracking script - `true` in the head and `false` in the body
        head: false,
        // Setting this parameter is optional
        anonymize: true,
        // Setting this parameter is also optional
        respectDNT: true,
        // Avoids sending pageview hits from custom paths
        exclude: ["/preview/**", "/do-not-track/me/too/"],
        // Delays sending pageview hits on route update (in milliseconds)
        pageTransitionDelay: 0,
        // Enables Google Optimize using your container Id
        optimizeId: "YOUR_GOOGLE_OPTIMIZE_TRACKING_ID",
        // Enables Google Optimize Experiment ID
        experimentId: "YOUR_GOOGLE_EXPERIMENT_ID",
        // Set Variation ID. 0 for original 1,2,3...
        variationId: "YOUR_GOOGLE_OPTIMIZE_VARIATION_ID",
        // Defers execution of google analytics script after page load
        defer: false,
        // Any additional optional fields
        sampleRate: 5,
        siteSpeedSampleRate: 10,
        cookieDomain: "example.com",
      },
    },
  ],
}
```

```

        // defaults to false
        enableWebVitalsTracking: true,
      },
    ],
  }
}

```

See below for the complete list of optional fields.

Note that this plugin is disabled while running `gatsby develop`. This way, actions are not tracked while you are still developing your project. Once you run `gatsby build` the plugin is enabled. Test it with `gatsby serve`.

<OutboundLink> component

To make it easy to track clicks on outbound links in Google Analytics, the plugin provides a component.

To use it, simply import it and use it like you would the `<a>` element e.g.

```

import React from "react"
import { OutboundLink } from "gatsby-plugin-google-analytics"

const Component = () => (
  <div>
    <OutboundLink href="https://www.gatsbyjs.com/plugins/gatsby-plugin-google-analytics/">
      Visit the Google Analytics plugin page!
    </OutboundLink>
  </div>
)

export default Component

```

Options

trackingId

Here you place your Google Analytics tracking id.

head

Where do you want to place the GA script? By putting `head` to `true`, it will be placed in the “`<head>`” of your website. By setting it to `false`, it will be placed in the “`<body>`”. The default value resolves to `false`.

anonymize

Some countries (such as Germany) require you to use the `__anonymizeIP` function for Google Analytics. Otherwise you are not allowed to use it. The option adds

two blocks to the code:

```
function gaOptout(){document.cookie=disableStr+'=true; expires=Thu, 31 Dec 2099 23:59:59 UTC; path=/';
...

ga('set', 'anonymizeIp', 1);
```

If your visitors should be able to set an Opt-Out-Cookie (No future tracking) you can set a link e.g. in your imprint as follows:

```
<a href="javascript:gaOptout();">Deactivate Google Analytics</a>
```

respectDNT

If you enable this optional option, Google Analytics will not be loaded at all for visitors that have “Do Not Track” enabled. While using Google Analytics does not necessarily constitute Tracking, you might still want to do this to cater to more privacy oriented users.

If you are testing this, make sure to disable Do Not Track settings in your own browser. For Chrome, Settings > Privacy and security > More Then disable Send a "Do Not Track" request with your browsing traffic

exclude

If you need to exclude any path from the tracking system, you can add it (one or more) to this optional array as glob expressions.

pageTransitionDelay

If your site uses any custom transitions on route update (e.g. `gatsby-plugin-transition-link`), then you can delay processing the page view event until the new page is mounted.

optimizeId

If you need to use Google Optimize for A/B testing, you can add this optional Optimize container id to allow Google Optimize to load the correct test parameters for your site.

experimentId

If you need to set up SERVER_SIDE Google Optimize experiment, you can add the experiment ID. The experiment ID is shown on the right-hand panel on the experiment details page. Server-side Experiments

variationId

Besides the experiment ID you also need the variation ID for SERVER_SIDE experiments in Google Optimize. Set 0 for original version.

enableWebVitalsTracking

Optimizing for the quality of user experience is key to the long-term success of any site on the web. Capturing Real user metrics (RUM) helps you understand the experience of your user/customer. By setting **enableWebVitalsTracking** to **true**, Google Analytics will get “core-web-vitals” events with their values.

We send three metrics:

- **Largest Contentful Paint (LCP)**: measures loading performance. To provide a good user experience, LCP should occur within 2.5 seconds of when the page first starts loading.
- **First Input Delay (FID)**: measures interactivity. To provide a good user experience, pages should have a FID of 100 milliseconds or less.
- **Cumulative Layout Shift (CLS)**: measures visual stability. To provide a good user experience, pages should maintain a CLS of 1 or less.

Optional Fields

This plugin supports all optional Create Only Fields documented in Google Analytics:

- **name**: string, tracker name
- **clientId**: string
- **sampleRate**: number
- **siteSpeedSampleRate**: number
- **alwaysSendReferrer**: boolean
- **allowAnchor**: boolean
- **cookieName**: string
- **cookieFlags**: string
- **cookieDomain**: string, defaults to 'auto' if not given
- **cookieExpires**: number
- **storeGac**: boolean
- **legacyCookieDomain**: string
- **legacyHistoryImport**: boolean
- **allowLinker**: boolean
- **storage**: string

This plugin also supports several optional General fields documented in Google Analytics:

- **allowAdFeatures**: boolean
- **dataSource**: string
- **queueTime**: number

- forceSSL: boolean
- transport: string

These fields can be specified in the plugin's **options** as shown in the How to use section.

TrackCustomEvent Function

To allow custom events to be tracked, the plugin exposes a function to include in your project.

To use it, import the package and call the event within your components and business logic.

```
import React from "react"
import { trackCustomEvent } from "gatsby-plugin-google-analytics"

const Component = () => (
  <div>
    <button
      onClick={e => {
        // To stop the page reloading
        e.preventDefault()
        // Lets track that custom click
        trackCustomEvent({
          // string - required - The object that was interacted with (e.g.video)
          category: "Special Button",
          // string - required - Type of interaction (e.g. 'play')
          action: "Click",
          // string - optional - Useful for categorizing events (e.g. 'Spring Campaign')
          label: "Gatsby Plugin Example Campaign",
          // number - optional - Numeric value associated with the event. (e.g. A product ID)
          value: 43,
        })
        //... Other logic here
      }}
    >
      Tap that!
    </button>
  </div>
)

export default Component
```

All Fields Options

- category: string - required
- action: string - required

- `label`: string
- `value`: integer
- `nonInteraction`: bool
- `transport`: string
- `hitCallback`: function

For more information see the Google Analytics documentation.

hitCallback A timeout is included by default incase the Analytics library fails to load. For more information see Google Analytics - Handling Timeouts

Troubleshooting

No actions are tracked

Check the tracking ID Make sure you supplied the correct Google Analytics tracking ID. It should look like this: `trackingId: "UA-111111111-1"`

Make sure plugin and script are loaded first The analytics script tag is not properly loaded into the DOM. You can fix this by moving the plugin to the top of your `gatsby-config.js` and into the head of the DOM:

```
module.exports = {
  siteMetadata: {
    /* your metadata */
  },
  plugins: [
    // Make sure this plugin is first in the array of plugins
    {
      resolve: `gatsby-plugin-google-analytics`,
      options: {
        trackingId: "UA-111111111-1",
        // this option places the tracking script into the head of the DOM
        head: true,
        // other options
      },
    },
  ],
  // other plugins
}
```